A FORMAL MODEL OF E-GOVERNMENT SUCCESS FACTORS FOR DEVELOPING COUNTRIES:

Dynamic relationships of software maintenance and information systems staff management

GUNADI

Sarjana in Math. (UGM, 1989)
MSc in Comp. and Math. Sci. (VUT,1996)

School of Management and Information Systems
Faculty of Business and Law, Victoria University

Submitted in fulfilment of the requirements of the degree of Doctor of Philosophy

2012

Abstract

This research formally models e-government success factor relationships with particular reference to developing countries. Much existing research assumes the success factors are mutually independent or in one-way linear dependency. This understanding could hinder the achievement of a successful e-government system – that is, a sustainable system. It is critical for developing countries to ensure that implemented e-government systems can be sustained.

This research employs a system thinking approach which is implemented using system decomposition and System Dynamics methods. Acknowledging the broad and complex nature of the success factors and adopting a point of view that the success factor is a system, the system decomposition method organises the success factors system into a manageable number of subsystems, each of which is further decomposed into sub-subsystems. This decomposition method enables the researcher to focus on a (set of) sub-subsystem(s) while at the same time adopt a holistic view. The System Dynamics method is applied to model the detail and dynamic relationships of the success factors within and between selected sub-subsystems. From within two important sub-subsystems, software maintenance and information systems staff management, the success factors relationships and its dynamic nature are modelled using this method. The relationships between the success factors are derived from the existing literature. A case study data collected from a successful e-government system in Indonesia and some necessary System Dynamic processes are used to validate the model – that is, to build soundness and confident usage of the model.

The developed model captures the complexity of the relationships between elements and processes of the software maintenance, the expectancy theory variables, and the competence and availability of the information systems staff. It provides an explicit perspective on the dynamic feedback relationships between these factors and e-government success. For this model, e-government success is operationalised as a high level of e-government software availability over time,

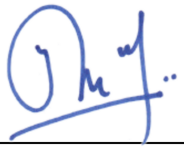
Through the model, this research results in an improvement of insight into the explicit mechanism of success factors, within software maintenance and information systems staff management, influencing e-government success and

assists decision-makers to implement decisive policies for achieving sustainable system services delivery. Simulation of the model, for example, shows that the ability of the e-government system of the Indonesian Ministry of Agriculture to achieve success is because of its ability to provide a sufficient level of rewards. These rewards dynamically influence staff effort and availability levels which in turn affect recurrent faults of software maintenance. These faults then influence the level of software availability over time which in turn affects rewards.

Opportunities for future research include linking the model with systems development, funding, change management and users' feedback, and incorporating hardware and computer networks.

Student Declaration

"I, Gunadi, declare that the PhD thesis entitled *A formal model of e-government success factors for developing countries: Dynamic relationships of software maintenance and information systems staff management* is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work."



Signature

16th July 2012

Date

Acknowledgements

Alhamdulillahi rabbil ‘alamiin.

I would like to express my sincerest appreciation and gratitude to all people and organisations for their contributions to the completion of my PhD study.

First of all, I am most grateful to Associate Professor Geoffrey A Sandy. As my Principal Supervisor, his constant support, guidance and encouragement have made my effort results in successful thesis completion. Especially, his infinite patience dealing with my “very difficult to understand expressions of way of thinking” and for showing me the way to make them clear and understandable is extremely invaluable. I also deeply appreciate for his persistent in reminding me to always take the readers’ point of view in writing a useful thesis.

I also would like express my deep gratitude to Prof. Michael McGrath, my Associate Supervisor, who has introduced me to System Dynamics method and how to do research with this method. His deep insight in System Dynamics made me possible to undertake an independent research journey.

I express my sincerest thank to the government and people of the Republic of Indonesia, especially to the DITNAGA of the Ministry of National Education, which provides constant financial support through the overseas PhD scholarship program. My appreciation also goes to the Rector of and my colleagues at Gajayana University, Malang, Indonesia for their invaluable support and assistance. I am also highly indebt to the managers and all staff of PUSDATIN, particularly Ir. Bayu Mulyana, MM., and Subandrio, S.Sos., for their assistance during data collection period.

I am also highly thankful and acknowledge the professional proofreading assistance of Ms Mary-Jo O’Rourke and Ms Sam Oliphant of The Second Office.

Finally, words cannot express my deepest appreciation to my wife, Rusmeiyati, for her unconditional understanding and support to me to pursue this degree. Her willingness to sacrifice, to experience a very difficult time and to take over my responsibility to lead the family alone for years has been the invaluable motivation of the fruitful accomplishment of this thesis. This thesis completion would not have been possible without unending support and understanding of my son, Ahsani Hafizu Shali, and my daughters Hasna Insani Kamila and Hanifa Insani Aulia. I should say sorry to them for being away and unable to accompany them for several years.

List of Publications

Gunadi, G & Sandy, GA 2009, 'A Formal Model for Electronic and Mobile Government Service Delivery Success Factors', in *Proceeding of the* 20th *Australasian Conference on Information Systems 2009*, 2-4 December 2009, Melbourne, Australia.
http://aisel.aisnet.org/acis2009/38

Gunadi, G, McGrath, GM & Sandy, GA 2010, ‘Towards a Formal Model of E-Government Success Factors: System Dynamics Modelling of Human Resource Impacts During Maintenance’, in *Proceeding of the 24 Annual Conference of the Australian and New Zealand Academy of Management,* 7-10 December 2010, Adelaide, Australia
ISBN: 1-877040-82-7

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Research purpose and research outcomes

This thesis reports on important research in the area of e-government success factor (eGSF) relationships with particular reference to developing countries. It identifies and formally models the relationships of the success factors. It tests the predictive value of the model with particular reference to the software maintenance (SM) and information systems (IS) staff management domains of e-government. The context is the operation and maintenance stage of an e-government system life cycle. Overall the model is validated and its usefulness tested through simulation of a number of scenarios.

The basic underpinning of this research is that it is not only the success factors that determine the success or otherwise of e-government, but also the relationships between those factors that play important roles in e-government success. E-government is a complex phenomenon formed by the interdependency of many factors; however, existing endeavours on e-government success factors have been dominated mainly by views that the factors are mutually independent in influencing e-government success, or that the factors are in one-way dependency relationships between independent and dependent factors. Ignoring the complex relationships between the success factors might hinder the achievement of e-government success, as dynamic feedback relationships between the success factors are shown by this study to govern e-government success.

The importance of this research can be viewed from the perspectives of the e-government system life-cycle and the limited resources owned by developing countries. While most of the current knowledge on e-government success is associated with and dominated by e-government system development, in fact an e-government system is expected to spend most of its lifetime within the operation and maintenance stage to deliver services. During this stage the emergence of unresolved errors and unsatisfied new requirements can jeopardise the sustainability of an e-government system's capability. From this point of view, achieving e-government success means realising sustainable service deliveries which ultimately assure sustainable benefits to both citizens and government. It is critical for developing countries to ensure that the benefits can be realised for as long as possible, otherwise their scarce resources for establishing and managing e-government systems will be wasted. Reports from previous research have indicated that many e-government systems implemented in developing countries cannot be sustained.

Consequently, in order to achieve this sustainable e-government services delivery the level of e-government software availability should be consistently high over time, which in turn necessitates continual and effective SM management. For this purpose, e-government success factors and their dynamic relationships in the e-government SM domain are investigated and formally modelled. Effective SM certainly requires high performance IS staff and, within this context, this research also models the relationships of success factors related to the motivation of the internal IS staff who maintain the e-government software. The expectancy theory is used as the theoretical foundation to model the dynamic feedback relationships of these motivational factors.

A system thinking approach implemented using IS decomposition and System Dynamics (SD) methods is adopted and employed by this research to formally model those e-government success factors and their relationships. The system decomposition is employed to organise the large system of e-government success factors into manageable subsystems and to provide an overview of the broadness of the success factors, and the SD method is applied to represent formally dynamic feedback relationships between the success factors. The SD method is capable of showing the level of relationships between success factors over time.

The developed dynamic relationships model is validated into an actual successful e-government system in Indonesia, a developing country which has been able to deliver sustainable services. Therefore, this research contributes significantly to both theoretical improvement and understanding. It extends the body of knowledge of e-government success by formally modelling the relationships of e-government success factors; improving insight into the significant role of the success factors and their relationships on e-government system SM and IS staff domains; and broadening the application of the expectancy theory into e-government SM through SD modelling.

This research also contributes significantly to the practice of e-government. The model enables e-government decision-makers or managers from developing countries to learn how to achieve e-government success. The resulting model facilitates their evaluation of policies leading to e-government success. Specifically, it could assist them to manage e-government SM and IS staff performing the maintenance in order to ensure highly available e-government software for services delivery over time. Accordingly, the model could be utilised to assist in overcoming one of many critical e-government problems – unsustainable e-government systems – in developing countries which, according the United Nations e-government survey 2010, mostly occupy a medium to low level on the e-government index.

Specifically, the research outcomes are as follows:

- A formal model of eGSF relationships validated with a successful e-government system in a developing country. The model, which focuses on ensuring e-government sustainability, allows an extension of current understanding of e-government success.
- A model describing the dynamic relationships between the success factors of e-government SM and IS staff management. The model extends the representation of SM elements and processes through dynamic and formal representation. It also presents an extension of the application of the dynamic model of the expectancy theory into the area of e-government SM. The extension also includes staff competence and availability success factors.
- A specific profile of a sustainable e-government system in Indonesia. This profile description presents a thorough examination of the characteristics of a successful e-government system specific to developing countries, and demonstrates how e-government SM and internal IS staff are successfully managed, leading to sustainable e-government service delivery.
- Simulation results of the validated formal model of eGSF relationships. These simulation outputs provide explicit details about the way the dynamic levels of a success factor influence the levels of other success

factors. This suggests valuable lessons for e-government decision-makers or managers about the impacts of changes in the success factors of a successful e-government system on other success factors and the level of e-government success.

- Lessons learned from scenario simulations of success factors. A scenario simulation suggests why and how a (set of) selected factor(s) and relationships eventually causes e-government failure or success. From a scenario simulation, a policy can be introduced to and experimented with the validated model without costing any resources or changing organisational settings in order to achieve e-government success and avoid failure.

1.2 Thesis structure

Chapter 1 summarises the rationale for this research, its scope, aims and expected outcome. It also describes the structure of this thesis document.

Chapter 2 presents a comprehensive review of existing e-government success modelling literature. The review reveals the nature and importance of e-government and indicates that e-government success is a complex phenomenon inheriting the complexity of e-government. However, the review shows that previous research has paid less attention to the relationships between the success factors. There is also rare research into e-government systems during their operation and maintenance stage. This review results in the identification of a research gap and the determination of the present research aims and scope.

Chapter 3 attempts to elicit the relationships between e-government success factors identified in previous research. Referring to the research gap, scope and success factor categories, in this chapter a comprehensive literature review is undertaken focusing on SM and IS staff management domains. This chapter identifies elementary-level success factors and their relationships from these two domains. The theoretical underpinning on which to build the relationships model is also identified and discussed.

Chapter 4 discusses the research approach, modelling methods and techniques, and a case study data-collection method. This chapter argues for the suitability of a systemic approach in viewing e-government success factors and their relationships, which is then implemented using system decomposition and SD modelling methods. The research design presented in this chapter to achieve the research aim broadly involves developing an initial model from the literature; collecting data from a successful e-government system in a developing country as a case study based on the initial model; and using the case study data and other techniques to validate and improve the initial model in accordance with the suggested validation process in the SD method.

Chapter 5 presents in detail the development of the decomposition and SD models of the e-government success factors system using the success factors and their relationships identified in Chapter 3 and the modelling methods described in Chapter 4.

Chapter 6 describes the realisation and results of the case study process dealing with a successful Indonesian e-government system which was selected to represent a successful system in developing countries. The chapter then demonstrates the use of case study data and the implementation of a validation process of the SD model, to ensure soundness, usefulness and confident usage of the model in supporting e-government decision-makers to realise e-government success.

Chapter 7 explores the dynamic behaviour of the model in which scenarios, referring to the success/failure factors identified in previous research, are introduced to the model and simulations are run on this adjusted model. Therefore, the chapter reveals explicitly how e-government success factors relate dynamically to each other. It also shows the use of the model, considering success factors and their relationships, and indicating how and why these scenarios could lead to e-government success or failure.

Chapter 8 concludes the thesis with the contributions and limitations of this research. These include contributions to the body of knowledge and practice in e-government. Some limitations of the research are also stated explicitly. Several possible directions for future research are suggested.

Chapter 2 Literature Review

2.1 Introduction

This chapter presents reviews of the literature pertaining to the nature and importance of e-government, eGSF models, and e-government in developing countries. The chapter also describes the identified research gap, aims and scope.

First, the presentation begins with reviewing e-government definitions and formulating a definition to be adopted by this study. To obtain a broad understanding, this is followed by reviews of e-government forms, stages of advancement, types of applications, their complexity and benefits. Second, varying notions of e-government success will be presented along with an adopted working definition. Next, eGSF models available within the existing e-government literature will be reviewed and evaluated. This section is especially intended to identify approaches and models adopted by previous research considering the relationships and organisation of the success factors. The strengths and weaknesses of previous approaches and models will be evaluated in order to identify the research gap. Fourth, the current status, issues and challenges of e-government in developing countries will be described. This section provides ideas regarding the focus of this study on developing countries. Last, the chapter presents research gap identified from previous studies, based on which the aims and scope of the current research are formulated.

2.2 Nature and importance of e-government

2.2.1 E-government definition

The first step in studying e-government is to understand the definition of e-government (Gil-Garcia & Luna-Reyes 2006). However, as a growing field of study, e-government literature has not established a relatively simple and widely-shared definition. The literature has shown that there are many different definitions (Misra 2007; Yildiz 2007) that depend on the objectives of the activities and who performs them (Yildiz 2007). The review of previous literature has indicated that most definitions of e-government underline the use or adoption of information and communication technology (ICT) (Bhatnagar 2004; Dawes 2002; Hovy 2008; Hughes 2003; Kraemer & King 2008). Most often e-government definitions confined themselves to the Internet or Web-based technology (Fang 2002; Gauld & Goldfinch 2006; Kahraman, Demirel & Demirel 2007; Kaylor, Deshazo & Eck 2001). Alhujran (2009) also concluded from reviewing existing literature that e-government is about delivering public services utilising web-based tools and applications. Furthermore, having extensively reviewed the e-government literature and examining the variations expressed in the available definitions, Hu et al. (2009) concluded with a collection of concepts that is mostly shared by previous studies in the definition formulation. They included:

1. the major initiatives of management and delivery of information and public services;
2. taken by all levels of governments (including agencies, sectors);
3. on behalf of citizens, business;
4. involving using multi-ways of internet, web site, system integration and interoperability;
5. to enhance the services (information, communication, policy making), quality and security; and
6. as a new key (main, important) strategy or approach (Hu et al. 2009**:** 979).

This short review indicates that many e-government definition formulations rank ICT or Internet technology as one of the underlying concepts necessary for an e-government system. Although it is a fact that without ICT there will be no e-government, viewing ICT – especially internet technology – as one of the focal points could prove misleading. Especially in regard to the current status of ICT in developing countries (United Nations 2010), those definitions could lead to an understanding on the part of e-government stakeholders in developing countries that e-government is equivalent to ICT acquisition and installation within government organisations in order to perform government activities. This perception certainly should be avoided. As we are reminded by Heeks and Bhatnagar (2001), it is IS that should be the focus of e-government development. An information system is an integration of various elements consisting of technology, information, organisational processes, and people who create and deliver services and values (Heeks 2005b). Further, as many studies have also shown, e-government development relying solely on ICT development often did not result in fulfilment of the expectations for e-government development (Gronlund 2005; Kraemer & King 2008). Moreover, Elliman et al. (2007**:** 370) indicated that technology could be "...the creator of the problems rather than a solution". In line with Heeks (2005b), it was emphasised by Silcock (2001) that e-government is a combination of technology, business processes and human resources with the government's customers at the centre.

In light of the aforementioned reasons, this study adopts the following e-government definition:

> *e-government is a government information system that enables better quality government operations, better services delivery, and more convenient and efficient access to information which both benefits its customers and enhances their participation*.

This definition explicitly introduces the IS notion as described by Whitten and Bentley (2007). The IS definition reflects an integration of ICT infrastructure, organisational resources, organisational objectives and values (Davis 2005; Whitten & Bentley 2007). As in general notions of IS, this definition emphasises the importance of the human and organisational aspects (Avison & Fitzgerald 2006). By embracing the notion of IS into the definition of e-government, this study also acknowledges an e-government as a socio-technical system (Beynon-Davies 2007; Heeks 2005b; Olphert & Damodaran 2007). This socio-technical point of view on e-government emphasises the importance of interdependency amongst various e-government elements in the sense that it consists of reciprocal interaction between "task, technology, people and structure" (Bostrom & Heinen 1977**:** 25). By formulating this definition, methods and tools available within the IS field will be utilised by this study.

As in Heeks (2008e), technologically the definition of e-government adopted by this study includes all types of networked-digital ICTs and public-sector activities and is not limited to the use of Internet technology. Limiting the definition of e-government by associating it only to the website that can be accessed directly by citizens might not be appropriate in view of the low levels of access to the Internet in developing countries (United Nations 2010). In other words, the limitation could cause any IS which deliver government services but are not implemented using internet technology not to be considered as e-government systems. Low levels of internet access and the digital divide in developing countries could prevent citizens from gaining access to government services if e-government is implemented only through internet technology. Even in a very developed country the digital divide is still a serious problem (Belanger & Carter 2009).

2.2.2 E-government forms, stages of development and applications

E-government can be categorised in various ways related to the main stakeholders involved and type of service delivered (Belanger & Hiller 2006; Beynon-Davies 2007; Fang 2002; Lee, Tan & Trimi 2005; Ndou 2004). For example, Belanger and Hiller (2006**:** 50) proposed to categorise e-government into six forms: "government with individuals – delivering services, government with individuals – political process, government with business as a citizen, government with business in the marketplace, government with employees, and government with government". Similar to this categorisation, Beynon-Davies (2007) classified e-government into five forms :

1. Internal e-government. This refers to the IS of the internal process of the government organisation. The systems deal with the government employees and managers who perform daily tasks and business processes within and amongst government units. The aim of this type of e-government is efficient, effective and seamlessly integrated internal operations.
2. Government to citizen (G2C). Development and implementation of this type of IS aims mainly at improving the delivery of government services to citizens. The citizens as the customers are expected to benefit from efficient processes through the elimination of many intermediaries and the high and continuous availability of government services.
3. Government to business (G2B). This form of e-government is an IS developed to accommodate the private sector's need of efficient and effective processes in dealing with government. The customers of this e-government form are business companies.
4. Government to government (G2G). This is a type of IS that enables various government units to collaborate on a particular or a set of government-activities or business. Development and implementation of a G2G is expected to be able to support the achievement of high-level or broad government objectives.
5. Citizen to citizen (C2C). Although in the past this interaction was not considered as part of e-government, Beynon-Davies (2007) argued that there is a need to involve community chains within the functions of a

government in order to improve government accountability and policy-making. Development and implementation of this type of IS can obviate this need.

This classification indicates that e-government systems are multi-faceted. In practice, it is possible that an e-government is designed solely to serve businesses such as an e-procurement system, citizens such as a birth certificate system, or both business and citizens such as a tax-filing system. However, it is not necessary that the development and implementation of an e-government system should follow that categorisation. In addition to the difference in the parties involved and services delivered, this classification indicates that the IS design and implementation of one category is different from that of other categories because of the difference in scope and coverage.

Focusing on the implementation of e-government through website technology, some studies have investigated how e-government should or may grow into its ideal functionality. These studies defined the growth of e-government into stages within a hierarchy of functionality including type of services and challenges of each stage (Belanger & Hiller 2006; Layne & Lee 2001). There are variations to the stages in previously proposed models, as shown in Table 2.1. In essence, the stage starts with the most simple and ends up with the most sophisticated system.

Table 2.1 *Proposed e-government growth models*

	Step1	Step 2	Step 3	Step 4	Step 5	Step 6
Layne and Lee (2001)		Catalogue	Transaction	Vertical integration	Horizontal integration	
Baum and Di Maio (2000)		Presence	Interaction	Transac-tion	Transform ation	
Ronaghan (2001)	Emerging presence	Enhanced presence	Interactive	Transac-tional govern-ment	Seamless	
Hiller and Belanger (2001)		Information dissemination	Two-way communica-tion	Integration	Transac-tion	Partici-pation
Wescot (2001)	E-mail and internal network	Enable inter-organisational and public access to information	Two-way communica-tion	Exchange of value	Digital democracy	Join-up govern-ment

Source*: Coursey & Norris (2008: 524)*

This growth model is expected to provide a path through which an e-government system improves its level of sophistication whereby a more sophisticated system assumes more benefits gained by both government and citizens. The benefits will be realised by citizens who can access the government services anywhere anytime using internet technology.

Similar to these proposed models, the United Nations (UN) also developed and utilised a model of e-government development stages to capture the level of online or website sophistication in delivery service by its member states (United Nations 2010). The UN's model consists of four level of sophistication (United Nations 2010**:** 95):

- **Level I –** "Emerging information services". At this level, government provides current information to the citizens regarding public policy, law and regulation and services, as well as links to the documents repository.
- **Level II –** "Enhanced information services". This level is associated with the capability of the e-government website to provide one-way or simple two-way communication in which the citizens can download documents or application forms.

- **Level III –** "Transactional services". This level indicates that the e- government website has been able to deliver two-way communication services delivery including financial transactions. A seamless back-end process is necessary to achieve this level.
- **Level IV –** "Connected service". At this level, the sophistication of the e-government development is higher than Level III in that government can establish seamless coordination and collaboration amongst government units to deliver services. Citizens and government can also exchange ideas through the website or other internet technology. The government service is more citizen-oriented. This level needs a quality IS.

However, some research has questioned the predictability of the growth model. Some empirical studies showed that many e-government systems, especially at the local level of government organisation in the United States, did not follow this model in terms of the growth of their sophistication level (Coursey & Norris 2008; Reddick 2004). Another study argued that in developing countries the growth model should be adjusted to accommodate their specific conditions – technical, social and political – compared with the developed countries (Zarei, Ghapanchi & Sattary 2008). They proposed a nine-stage model of e-government growth for developing countries: "strategy development, building infrastructure, building trust, making portal, initial interaction & stimulation, prototyping, enrichment, integration, and ICT industry development" (Zarei, Ghapanchi & Sattary 2008: 204).

Despite some critiques these models, especially the UN's simple four-stage model, is a valuable tool for e-government stakeholders to measure or benchmark their e-government systems' level of achievement. At the same time the models also provide challenges that need to be overcome and types of systems and services that need to be developed in order to elevate any e-government system to a higher level of sophistication (Layne & Lee 2001).

Although the level of sophistication can be associated with the degree of benefit, the use of the model to measure the benefits of an e-government system should be used with some caution. It must be noted that as the models refer only to service provision through an e-government website, therefore the services delivered through other methods are not considered. These models also implicitly assume that all government customers have access to the internet and know how to access e-government services through a website, while in fact that this is not always the case, even in a most developed country (Belanger & Carter 2009).

With respect to developing countries' conditions, especially the low level of internet penetration (United Nations 2010), the use of one of these models to evaluate the value of e-government to the citizens also needs to consider non-internet based e-government services. This is so because, as shown by Zhang et al. (2008), the IS deployed by a government organisation can also deliver highly valuable services and provide substantial benefits to citizens without delivering e-government services through the internet. This means it is possible that a government organisation has been able to deliver services by implementing an information system which highly benefits its citizens even though in terms of website services provision the e-government system is at level I or II. On the other hand, a government organisation could achieve level II but it is the only service available for the citizens. With low-level internet penetration, the service will not be able to reach the intended stakeholders.

In addition to the forms of an e-government system and its stage of sophistication, the extant studies indicated that e-government systems are also implemented in a variety of specific applications. One example of an e-government application is a "virtual town-hall" information system (Baldersheim 2004). This application provides a website interface to citizens and business in various aspects of government-to-citizens and government-to-business communication and transactions (Becker et al. 2004). Other examples are special-purpose e-government systems such as human resources management IS (HRMIS), general office environment systems (Singh & Goel 2007) and project management systems implemented within a central government office (Hussein et al. 2007). These types of e-government applications are not used to deliver services directly to citizens but to improve the efficiency and effectiveness of internal government operations such that the government has the capability to serve its citizens better.

Examples of e-government applications that provide services directly to citizens or businesses are public hospital service information systems (Gauld 2007), the e-stamping service (Luk 2009), tax administration (Gonzalez, Gasco &

Lopis 2007; Smith 2011), e-procurement (Smith 2011) and the geographic information system (de Kool 2010).

2.2.3 E-government system life cycle

As an IS, an e-government system can also be studied or managed according to the IS development life-cycle (ISDLC) (Heeks 2005b; Tsai, Choi & Perry 2009). By following the ISDLC, the e-government system life-cycle subsequently consists of (Avison & Fitzgerald 2006**:** 31-4):

1. "Feasibility study." This stage concerns the identification of a possible e-government project, analyses the real need, the broad requirements and feasibility.
2. "System investigation." This is a thorough fact-finding process about the current state of the system including functional requirements and problems as well as system constraints.
3. "Systems analysis." Based on the investigation stage, the system analysis tries particularly to elicit the requirements of the users who are responsible for the system operation, who will use the output, provide input and manage the processes of the system.
4. "Systems design." This stage implements the knowledge that is obtained during the system analysis. The activities comprise designing a computer-user interface, computer network, process algorithm and data repository, which are mainly technological aspects, as well as new business processes or new modes of service delivery.
5. "Implementation." This part deals with creating the software, acquiring the hardware, implementing the computer network, building facilities, testing and documenting the system. Other activities within this stage are introducing the new system, training the users in how to use the newly-implemented system, and converting old data into the new format.
6. "Review and maintenance." This is the operational stage of an e-government system where the system should spend most of its lifetime to deliver the services for which it is being designed. During this operational stage, flaws undetected at previous stages and new requirements can emerge (Grubb & Takang 2003). At this stage the most important thing is to ensure that the implemented system can continuously support the organisational objectives. Therefore, maintaining the system to correct emergent problems and to

modify or enhance the system with respect to new organisational or user requirements are the main activities of this stage (Grubb & Takang 2003). At a certain level, the system maintenance might not be able to cope with the new requirements and the advance of technology, hence a completely new system may need to be developed.

This e-government system life-cycle indicates that each stage has its own specification in terms of activities performed, resources required, organisational setting, personnel involved and other support needed (Heeks 2005b). In short, the factors necessary for ensuring e-government success are different from one stage to another stage of the cycle. By understanding this e-government life-cycle, decision-makers and those who are in charge of e-government management can establish strategies for each stage to ensure the achievement of their e-government system objectives.

This life-cycle, broadly, can be divided into two stages: "1. The initial development of software, and 2. The maintenance and use of software." (April & Abran 2008**:** 10). To achieve the e-government system objectives, proper implementation of IS development project management for the first stage and quality maintenance of the system for the second one are necessary.

2.2.4 Complexity of e-government

Variations in definitions, forms, sophistication growth and applications found in literature and practice indicate the complexity of e-government systems. This complexity has also been acknowledged by much of the previous e-government research. For example, Scholl (2007) and Shareef et al. (2011) emphasised that the e-government system is a complex phenomenon which involves not only government organisation and processes and ICT implementation but also individual, social, political, economic and marketing factors etc. Further, in order to fully understand e-government systems, interdependence and interaction among those factors should also be comprehended (Beynon-Davies 2007; Scholl 2007). Admitting this complexity, Beynon-Davies (2007) developed and proposed a meta-model of e-government that was constructed from the interdependencies of vertical and horizontal dimensions, adopting a socio-technical paradigm. The vertical dimension deals with the informatics infrastructure supporting human resources infrastructure, while the horizontal dimension models the government business system that creates and delivers values to the government's customers. In line with this perspective, especially in regard to achieving e-government system goals and objectives, Shareef et al. (2011) and Titah and Barki (2008) argued that it is crucial to consider the interaction and complex relationships between the success factors. In other words, comprehending this complexity is necessary to avoid failure, which is relatively common in e-government development and implementation (Heeks 2003).

In order to implement this concept of e-government complexity, Gil-Garcia and Pardo (2006) urged the use of a comprehensive and dynamic view of e-government systems. This view was explained by Scholl (2007**:** 74), involving six high-level variables (information use, technology use, public policy, government operations, government services, citizens' engagement), that:

> relationships are likely to exist between all variables defying the establishing of any clear-cut cause-and-effect, independent-to-dependent-variable hierarchy between them.
>
> More importantly, all high-level variables interact with each other in a feedback fashion. When, for example, certain policies are imposed on the uses of technology, not only that particular variable will be affected, however; over time all other variables in the grid will be impacted.

Acknowledging these complex feedback relationships and using this dynamic view, Martinez-Moyano and Gil-Garcia (2004) for example modelled the interaction of citizens, government agencies and interaction forms as well as the impact of time to understand the growth of e-government sophistication using a qualitative modelling method of SD. Likewise, Luna-Reyes and Gil-García (2009) investigated the interaction among institutional arrangement, organisational forms and enacted technology in order to understand the development of e-government systems websites in Mexico by implementing a quantitative modelling method of SD.

Comparing the formulation of the e-government definition adopted by this study and acknowledging the complexity of an e-government system, it seems clear that the definition adopted has embraced the complex nature of e-government.

2.2.5 Importance of e-government

The importance of an e-government system can be evaluated from the degree of benefits that can be realised through a successful development and implementation of that system (Gichoya 2005). However, as there are many different stakeholders involved with an e-government system, then there can be a number of different perceptions as to the realised benefits (Dada 2006; Sedera, Gable & Chan 2003). In other words, the benefit of an e-government system can be viewed from many different standpoints including government as organisation, government staff, government managers, e-champions, government leaders, citizens as a whole and as individual, private businesses and other non-governmental organisations. Considering this variability, it is very unlikely to have a converging perception of benefits among e-government system stakeholders. The degree of importance of an e-government system to its stakeholders much depends on what position the stakeholder holds when viewing the system.

Irrespective of the stance of the stakeholders, by referring to some attributes, there are many kinds of benefits that can be realised by successful e-government system development and implementation. For example, Gichoya (2005) listed benefits such as easier communication, increased capacity of government, improved decision-making and transparency, and improved services efficiency. Dawes (1996) in Pardo and Tayi (2007) classified the benefits as technical, organisational and political.

The importance or benefits of an e-government system can also be assessed by what is being delivered by the system. For this, Bhatnagar (2004 Ch. 3) provided an encompassing list of good governance goals and the role of a successful e-government system, as depicted in Table 2.2.

***Table 2.2** Potential benefits of the e-government and how benefits can be realised.*

Good governance goals	How e-government can help
Increasing Transparency	• Dissemination of government rules and procedures; citizens' charters; government performance data to wider audience • Disclosure of public assets, government budget; procurement information • Making decisions of civil servants available to public
Reducing administrative corruption	• Putting procedures online so that transaction can be easily monitored • Reducing the gatekeeper role of civil servants through automated procedures that limit discretionary powers • Eliminating the need for intermediaries
Improving service delivery	• Less time in completing transactions • Reduction of costs associated with travel for citizens to interact with government • Improving government's ability to deliver service to larger segment of population
Improving civil service performance	• Increased ability of managers to monitor task completion rates of civil servants • Improved efficiency of civil servants by automating tedious work • Increased speed and efficiency of inter- and intra- agency workflow and data exchange • Eliminating redundancy of staff
Empowerment	• Providing communities with limited or no access to government with a new channel to receive government services and information • Reducing the brokerage power of intermediaries
Improving government finances	• Reducing cost of transactions for government processes • Increasing revenue by improving audit functions to better track defaulters and plug leakages by reducing corruption • Providing better control of expenditure

***Source**: Bhatnagar (2004**:** 37-8)*

2.3 E-government success

2.3.1 Definitions and measurements

Existing research has shown that there is no single encompassing construct to define the success of an e-government system (Liangwen 2009); it varies and certainly depends on the context and purpose of the studies. This variation also applies in practice depending on the range of stakeholders involved and their subjectivities (Dada 2006; Heeks 2008b; 2008d**:** 10). In this respect, for instance, Heeks (2008c) and Sedera, Gable and Chan (2004) suggested consulting the associated stakeholders in regard to what level of their goals should be met in developing an e-government system. In addition to the dependence on the stakeholders and their subjective views, a review of previous studies indicates that e-government success has been defined by associating it with e-government project development, adoption by citizens (external users), adoption by internal users, internet- or web-based system or non-internet-based system.

For example, in studying success factors that influence the development of an e-government project, Rosacker and Olson (2008) used the successful implementation of an e-government project to define e-government success. In their definition, an e-government project's success is measured by project completion time, budget allocated and used, and the ability of the system to deliver its intended functions and purposes.

Certainly, much research has also studied e-government success from the demand side, or external users' acceptance and adoption, for example Kumar et al. (2007), Lin, Fofanah and Liang (2011), Song (2010) and Titah and Barki (2008). In essence, they defined the success of an e-government system by evaluating whether users intended to use or actually used the system to obtain government services. From this context, Kumar et al. (2007) measured success by examining the recurring and preferred use of an e-government system by external users to obtain information, as well as to interact and transact with government. Many other studies implemented DeLone and McLean's (2003) IS success model to evaluate the adoption of e-government system by focusing on the quality of the system, the information and the services provided that lead to persistent use of an e-government system, for example Gonzalez, Adenso-Diaz and Gemoets (2010), Liangwen (2009), Sang, Lee and Lee (2009) and Teo, Srivastava and Jiang (2008).

There were also research studies that examined and defined e-government success from the supply or government side. Using this viewpoint to judge e-government success, the studies stipulated some criteria or indicators from the government side such as perception of government employees, degree of functionality of the e-government website, or the abilitiy of the e-government system to operate over a long time. For example, in studying the success of back-end systems of e-government in Malaysia, Hussein et al. (2007) used internal users to evaluate e-government success in terms of the perception of the government employees in performing their duties using the system. In their study they applied DeLone and McLean's (2003) model to define and measure e-government system success. Gil-Garcia (2005) adopted website functionality of e-government systems to define the success of local- or state-level e-governments in the United States. On this supply side study, Gil-Garcia used the "overall state e-government ranking (score), digital state e-commerce score, number of e-commerce systems and number of online services" (Gil-Garcia 2006: 4) as the indicators in measuring e-government success. Within the context of a developing country, India, Krishna and Walsham (2005) and Best and Kumar (2008) defined an e-government system as successful if the system is sustainable and based on how various stakeholders view e-government. Based on this sustainability criterion, they classified an e-government system as successful if it has been able to persistently operate and expand its functionality over several years.

This review of the definition of e-government success has underlined the existence of diverse notions and formulations of such success. The context, stance, stakeholder position and development or growth stage of an e-government system should be considered in evaluating, understanding and defining its success.

2.3.2 A working definition and measurements

The review of the literature has revealed the existence of the differences and lack of consistency in defining e-government system success. An intention to adopt an existing definition suited to the present study could be hindered by this diversity, because the available definitions do not fit with the purpose and limitation of this study. Accordingly, the study will develop and adopt a working definition of e-government success. Acknowledging the dependency of a definition of e-government success on the context and purpose, the definition formulation will adopt a widely validated IS success model, be based on a particular side of e-government, and be within the context of developing countries.

In the IS field, DeLone and McLean's (2003) IS success model is an exemplary one. It received empirical support, fully as well as partially, from various researchers in individual and organisational contexts (Petter, DeLone & McLean 2008). This model consists of six variables: information quality, system quality, service quality, intention to use/use, user satisfaction and net benefit in a dependency relationship. The first three variables separately influence both the intention to use/use and the user satisfaction variables. Then, the intention to use/use and the user satisfaction variables influence each other and also both influence the net benefit. The latter in turn carries influence back to both the intention to use / use and the user satisfaction. In this model, DeLone and McLean emphasised that the net benefit is the most important variable in measuring the success of an information system. However, they explained that the net benefit variable alone cannot be understood without measuring the other variables in the model, with measurement of the net benefits being dependent on the context and objectives of the system development.

This IS success model has also been utilised and validated within the government information system context – see for example Hu et al. (2005), Hussein et al. (2007), Hussein, Karim and Selamat (2007), Sedera and Gable (2004) and Wang and Liao (2008). Referring to the description of this model, these validations can be interpreted as that the net benefits can be realised, whether by individual or organisation, only if the e-government system is of high quality, can provide quality information and supports the provision of quality services (Chutimaskul, Funilkul & Chongsuphajaisiddhi 2008; Wang & Liao 2008).

For this study, the definition of e-government system success is formulated based upon the perspective of the importance of the sustainability factor of an e-government system (Ali & Bailur 2007; Horiuchi 2006; Hunt 2001) for developing countries, the antecedent variables of the widely adopted and validated DeLone and McLean's IS success model, as well as the UN's e-government sophistication growth model. By considering these backgrounds, the study defines e-government success as:

the realisation of a sustainable quality e-government system that delivers quality information and services.

The definition can be explained as follows.

- Basically, this definition emphasises that the most important variable of e-government system success is the net benefits, as underlined by DeLone and McLean's model. However, based on the variables dependency of this model, the net benefits can only come to evidence through realisation of a sustainable quality e-government system that delivers quality information and services.
- The realisation should be sustainable in the sense that the quality e-government system should be able to operate continuously for a long period of time and be able to grow in its scope and coverage of services (Krishna & Walsham 2005). In terms of the period, logically it should be as long as possible. In their study, Krishna and Walsham (2005) used four years as a measure of e-government sustainability. Following Krishna and Walsham's study, this research uses a period of at least four years as a measure of sustainability. In addition to the system's lifetime, its sustainability is also measured by an increasing number of features or services available in the implemented application system, the number of application systems and number of users.
- The level of quality of an e-government system is measured by the level of the growth in sophistication achieved by the system based on the UN's model. Referring to this online index measure and e-government system development in developing countries (United Nations 2010), a quality e-government system should at least achieve the second level. As stated by Shareef et al. (2011**:** 18), considering level III and beyond that "most of the countries are still struggling to attain this e-Gov service level". In addition, an e-government system should also be supported by an integrated computer network, although not necessarily an internet-based system. This description is based on the fact that most government websites in developing countries are still within the second level, and many non-internet-based but computer network-based e-government systems in developing countries can provide significant benefits to their customers (citizens, businesses and government

employees or government managers) such as those studied by Hussein et al. (2007), Krishna and Walsham (2005) and Zhang et al. (2008).

The definition and description of e-government success provided is based on the reasoning that the success level should be realistic and achievable from a developing country's position. This means that an e-government system which is judged as successful can be used as a referent for other e-government system decision-makers as a learning example. The lesson drawn from a successful e-government system case can be used to develop strategies for achieving success within similar conditions. Also, with respect to the 'position' or stance from which the success is viewed, the adopted definition considers the success from the supply or government side. This point of view focuses on harnessing strategic efforts, managing available resources and overcoming barriers to materialise a quality e-government system development, implementation and utilisation (Kunstelj & Vintar 2004). This is in contrast to the demand-side point of view where the success is measured by the level of adoption by citizens in term of recurrent and preferred use of e-government websites (Kumar et al. 2007; Titah & Barki 2008).

2.4 E-government success models

2.4.1 Success factor

The success of an e-government system depends on e-government success factors. A factor is "one of the elements contributing to a particular result or situation" (Dictionary.com LLC 2011). An eGSF is a thing (object or activity or condition) that contributes to or determines the success of an e-government (Fairchild, Ribbers & Nooteboom 2004; Gichoya 2005). Gichoya (2005) distinguished success factors into drivers and enablers. The former refers to those that reinforce success while the latter refers to active elements that help overcome potential barriers. Gichoya also defined failure factors – the things the occurrence of which constrains proper development and implementation of an e-government system. The failure factors can be either barriers or inhibitors. The occurrence of barriers hinders e-government system implementation, while inhibitors prevent advancement and restrict successful implementation and sustainability. Additionally, Fairchild et al. (2004) differentiated success factors into two kinds: those that are beyond the control of an organisation but with which the organisation operates and interacts, and those that characterise organisational operations and are under the organisation's control. The first are external factors whereas the latter are internal ones.

From the description of the eGSFs, it can be concluded that the success factors and the failure factors are just opposite sides of the same coin. Therefore, in a study of e-government success it is also necessary to examine e-government failure factors. This stance has also been taken by many previous e-government studies. For instance, Coursey and Norris (2008) studied barriers that prevent an e-government system from achieving success; Heeks (2008b) studied e-government success versus failure factors in developing countries; and Kumar and Best (2006) reported the results of their research on the causes of an e-government failure. This research showed the factors that cause an e-government failure and proposed methods to avoid the failure, hence to achieve a success.

Considering the aforementioned reasons, this study will take into account and name both e-government success factors and failure factors as success factors. For example, a lack of internal ICT-skilled staff is a failure factor and a sufficient number of internal ICT-skilled staff is a success factor. In this study, both are success factors.

2.4.2 Success factor models

By acknowledging the complexity of e-government systems, existing eGSF literature will be reviewed to explore how the eGSFs were modelled in influencing an e-government success. There has been much research done to investigate eGSFs and to model the relationships between eGSFs and e-government success from various perspectives.

From the demand side, much previous e-government research tried to investigate what attributes e-government systems should have and/or other success factors that influence the intention of citizens or business to use e-government system based services. For example, Carter and Bélanger (2005) empirically modelled eGSFs derived from the technology acceptance model (TAM), the diffusion on innovation (DOI) and web-trust model to advise e-government decision-makers on the factors that affect citizens' adoption of e-government services. More recently, Shareef et al. (2011), by acknowledging the interdependence of eGSFs, also tried to empirically model technological, social, cultural, organisational, economic, political and marketing aspects of customer success factors in order to predict the Canadian adoption of static and interactive e-government services. In their research they were extending currently available e-government user adoption models by adopting the first two levels of e-government growth as dependent variables. As distinct from those studies interested in citizens' behaviour intention, Lee, Kim and Ahn (2011), by focusing on business users of local e-government in Korea, modelled the influence of available e-government offline services quality and trust in internet technology success factors on the willingness of business users to adopt online services. Further, by adopting and extending TAM, Lin, Fofanah and Liang (2011) empirically implemented the model to reveal the relationships between the success factors of TAM, IS quality, information quality and the intention of the citizens of Gambia, a developing country, to use e-government services.

On the other hand, many researchers have also studied and modelled eGSFs from the supply side of e-government systems in order to understand the relationship between eGSFs and the successful development of e-government. For example, Gil-Garcia (2005, 2006) empirically studied and modelled possible relationships between a number of eGSFs: web management practices, general organisational factors, institutional arrangements, overall size of the economy, voting preferences and demographic factors, and their influence on e-government success within the states of the US. In this case, the success level was relatively objectively measured based on the states' website functionality, which was based not on users' perception but on the number of facilities available on the website and the overall e-government score. In contrast, Hussein, Karim and Selamat (2007) investigated Malaysia's e-government systems success based on the perception of internal users. They implemented an IS success model as the attribute of e-government success, and empirically modelled the relationships between IS facilities, IS competence, IS integration, user support and decentralised IS structure and the success of e-government. Further, from the supply side of e-government systems but different from Gil-Garcia's (2005, 2006) and Hussein, Karim and Selamat's (2007) perspectives, Rosacker and Olson (2008) modelled the influence of a number of critical success factors on an e-government development project which has a specific set of objectives, allocated budget and time period.

In addition, there has been some research which did not specifically position itself as demand or supply side, or which may address both sides in investigating and modelling the eGSF, such as Lal and Haleem (2009) and Rashid and Rahman (2010). In their research, they included the eGSFs associated with the internal organisation of e-government systems as well as those that specify citizens' attributes in developing countries.

The aforementioned descriptions view previous research on eGSFs models from the position of those who judge the success of e-government. Models of eGSFs relationships proposed and/or formulated in previous e-government success research can also be evaluated from the way the models were formulated, tested and validated, which is in line with the interest of this study.

In a study to identify local e-government success factors, Grabow, Druke and Siegfried (2004) listed 50 eGSFs and categorised them into 10 eGSF categories. These eGSFs have been utilised to study several local e-government system successes from several different countries (Druke 2005). By using a list, this study did not implement any modelling tools or methods to understand the eGSF relationships.

Much research has studied e-government success and modelled the success factors using a qualitative case study approach. This is in the sense that the relationships are presented using narrative descriptions. For example, Krishna and Walsham (2005) presented how eGSF influences the success of local e-government in India. In this longitudinal case study they described how the eGSFs identified influence the sustainability of an e-government system. In a different case study, Gauld (2007) analysed and identified the failure factors of a health information system in New Zealand. Similar to Krishna and Walsham, Gauld described qualitatively the way in which the eGSFs caused the e-government system to fail and the relationships between the eGSFs. Another example studied the eGSFs of the e-stamping service in Hong Kong. From the perspective of the leadership, stakeholders and service quality in public administration success factors, Luk (2009) studied the success of this e-government system development and implementation. In this study, Luk identified and explained how the eGSFs – leadership; law amendment;

collaboration etc. – relate to each other in influencing e-government success qualitatively.

A qualitative model of eGSFs relationships can provide rich insight into the way the success factors influence e-government success. However, the value of relationships describing the relative importance of the factors in determining directly or indirectly the e-government success is missing from the models. In addition, what the identified eGSFs are and how they relate to each other might depend on the subjective understanding of the readers.

Previous research has also indicated that quantitative approaches have been widely implemented to study the relative degree of importance of relationships or influence of the eGSFs on e-government success. These approaches were based mainly on statistical methods. For example, both Almahamid et al. (2010) and Hussein et al. (2007) implemented statistical correlation analysis. In their studies, Almahamid et al. (2010) applied non-parametric statistical correlation analysis to empirically investigate the intention of Jordanians to use e-goverment services involving the citizens' perceptions on usefulness, ease of use and information quality success factors, while Hussein et al. (2007) calculated statistical correlation values of a number of pairs of an eGSF and an e-government success dimension in studying several of Malaysia's e-government systems. In spite of quantitative information about the association of the eGSFs and e-goverment success, they did not evaluate the correlation between the eGSFs. In addition, correlation does not reflect the degree of influence of the eGSF on the e-government success because this method does not specify the cause and effect variables (Mendenhall, Beaver & Beaver 1995).

If the aim of a study is to investigate the influence of eGSF on an e-government success, a multiple regression method is more appropriate than the correlation analysis. A regression model consists of one dependent variable and several independent variables (Hair et al. 2006). There was much previous research that investigated what success factors influence e-government success by implementing a multiple regression method (Best & Kumar 2008; Ho & Ni 2004; Hussein, Karim & Selamat 2007; Lee, Kim & Ahn 2011; Mohamed, Hussin & Hussein 2006; Schwester 2009; Shareef et al. 2010; Shareef et al. 2009). However, a multiple regression method assumes that the eGSFs as independent variables do not correlate with each other (Mendenhall, Beaver & Beaver 1995). Therefore, by using this method, that research treated the eGSFs as though they are independent of each other. Accordingly, the eGSFs relationships will not be able to be revealed.

A growing number of studies have also implemented more flexible linear models, allowing researchers to investigate the relationships among eGSFs in influencing e-government success, such as the Structural Equation Modelling (SEM) and Partial Least Square (PLS) methods. Examples of previous research that studied e-government success and implemented one of these modelling methods are those by Ozkan and Kanat (2011), Shareef et al. (2011), Lin, Fofanah and Liang (2011), Prybutok, Zhang and Ryan (2008), Gupta, Dasgupta and Gupta (2008), Wangpipatwong, Chutimaskul and Papasratorn (2008), Hung, Chang and Yu (2006) and Gil-Garcia (2006). Using one or other of these methods, these research studies also investigated the influence of one or more eGSFs over other eGSFs, in addition to studying the influence of the eGSFs on e-government success. To formulate the model, most of these researchers adopted and extended well-established models in the IS field such as TAM, Unified Theory of Acceptance and Use of Technology (UTAUT), DOI and the trustworthiness model for e-government success; therefore the research mostly focuses on the perceptions of users about e-government system attributes: attributes that, if satisfied, could make citizens or businesses have the intention to use, use, or continuously use an e-government system service. In addition to focusing on the user's perception, further review of this literature shows that they only modelled the influence of eGSFs in a linear and one-way direction on e-government success.

In addition to the aforementioned modelling methods, other methods have been adopted by other studies to model the eGSFs. Lal and Haleem (2009) modelled the eGSFs structural relationships of rural e-government systems in India using the Interpretive Structural Modelling method. The method involves identifying relevant success factors, establishing a contextual relationship between factors, developing and manipulating a structural self-interaction matrix of factors and drawing a directed graph. From this modelling process, a structural diagram describing relationships of the eGSFs can be revealed. The relationships indicated in the resulting diagram show a chain of dependence from one eGSF to another. An insight can be excerpted from this model regarding what is the most influential factor and what is the most dependent on other factors. However, reciprocal relationships and the quantitative degree of relationships cannot be observed from the model. Distinct from this method, McMillan (2009) used matrices of success factors in order to organise a large number of success factor relationships. On the basis of six domains of eGSFs: business re-engineering, education, acceptance, security, cost and access, McMillan empirically identified lower-level success factors for each domain and then investigated the possible relationships between each factor from one domain and each factor from all other domains. Thereby, various relationships between success factors can be disclosed. In contrast with Lal and Haleem's, this model does not specify the (chain of) direction of relationships, and the relationships were stated qualitatively. Another eGSFs modelling perspective was used by AlShihi (2006). AlShihi studied the critical barriers that prevent the willingness of Omani citizens to use the national level e-government system service deliveries. Several strategic-level factors were identified and eventually mapped into a chain of cause-effect relationships of the eGSFs. This map qualitatively models the relationships of eGSFs that explains the way various critical factors affect the adoption and diffusion of the e-government system in Oman. As aimed for by the research, the identified qualitative cause-effect relationships of the eGSFs were expected to be able to assist the e-government decision-makers at a national level to develop strategies for improving the willingness of Oman's citizens to adopt e-government system services.

In sum, eGSFs and how they influence e-government success have been attracting the attention of many researchers. There was also a wide diversity in the perspectives from which previous research treated the eGSFs, how they considered possible relationships among eGSFs in influencing e-government success within their models. Despite its contribution to improving the body of knowledge regarding eGSFs and assisting e-government decision-makers in understanding what, and how, success factors could lead to e-government success, much of the previous research, to some extent, did not embrace the complexity notion of the eGSFs. Especially, many existing models provide only a snapshot view and lack attention to the impact of time on e-government success.

The aforementioned literature review indicates that some of the research listed the eGSFs, described the relationships qualitatively, assumed there is no relationship among eGSFs, modelled the relationships without cause-effect direction, modelled one-way and linear influence of eGSFs, or qualitatively modelled the cause-effect relationships focusing on the e-government users. In addition, previous research was also dominated by the adoption and diffusion from the demand side, and the development and implementation stage of e-government life-cycle from the supply side. Many eGSFs models were built by acknowledging the complex relationships between the success factors; however, the models did not accommodate the feedback relationships between eGSFs and feedback influence of dependent factors on the independent ones. The reviews also indicate that there was a lack of modelling research dedicated to modelling the success factors when success means a sustainable e-government system.

2.4.3 High-level success factor models

In addition to modelling the influence of eGSFs on e-government success, literature in previous research also tried to organise the identified success factors into a smaller number of classifications. By doing this, a more strategic level of eGSFs that encapsulate complex and detailed or operational-level success factors might be able to help e-government decision-makers to comprehend the complex nature of e-government in a more enlightening and holistic way. Existing eGSF literature indicates that the way in which eGSFs are classified varies. There was no agreement in conceptualising the success factors into a smaller number of “high-level” success factors.

With respect to their identified success factors of local level e-government systems in Germany, Grabow, Druke and Siegfried (2004) categorised them into 10 high-level categories. These 10 categories, which were used as a study framework for a number of local e-government systems in several countries, are guiding principles and strategy; organisation, project and change management; applications; benefits and costs; the right technology and organisation of the use of technology; competence, motivation, and qualifications; creation of acceptance, marketing; co-operation and partnerships; sustainable securing of resources; and legality. The use of eGSFs categories as a study framework was also indicated in Gil-Garcia and Pardo’s (2005) research. They examined the gap between researchers and practitioners at the strategic level with regard to five success factor categories: information and data; information technology; organisational and managerial factors; legal and regulatory factors; and institutional and environmental factors. On the basis of these categories and using the e-government systems in Canada and the United States as the background, Gil-Garcia and Pardo investigated the eGSFs from existing research and tried to seek the correspondence between research output and government practitioners' practices at the strategic level.

Some other research indicated a different way of classifying the eGSFs and formulating their proposed model. For example, by acknowledging the fact that the issues of adoption and acceptance of e-government systems are naturally complex, Titah and Barki (2008) inferred and synthesised the success factors found in existing literature into five broad topics. These are the influence of managerial practices, the influence of organisational and individual characteristics, the influence of governmental sub-cultures, the influence of IT characteristics and the assessment of e-government impacts. In addition, they concluded the importance of conceptualising "the interdependencies and non-linear relationships that exist between these factors as well" they had found missing in past studies. Another concept, which was developed based on the existing literature examining e-government success factors, was also proposed by Altameem, Zairi and Alshawi (2006). Their model proposal consists of three categories of critical success factors: governing factors, technical factors and organisational factors for e-government adoption, which they believed are able to assist decision-makers to achieve a successful e-government implementation. Another high level factor model, adopting Bostrom and Heinen's socio-technical model, was proposed by Alalwan and Thomas (2011), which consists of management of e-government, e-government technology, e-government process and e-government stakeholders. This proposed model was intended to be an evaluative framework for e-government systems. This model acknowledges complex interaction of the factors constructing an e-government system.

Different from these proposed models, Weerakkody, El-Haddadeh and Al-Shafi (2011) moved further in studying the category of eGSFs. In addition to conceptualising and categorising the eGSFs derived from existing e-government literature, they also empirically evaluated their model against e-government implementation and diffusion at the national level in Qatar. On reviewing the existing e-government literature, they divided the eGSFs into four broad themes: organisational, technological, political, and social; then each of these themes is further broken down into narrower themes. These eGSF categories were used as a guide to study the challenges surrounding Qatar's e-government implementation and diffusion. In another empirical study, Rashid and Rahman (2010) categorised eGSFs found from existing e-government literature into four high-level factors: institutional, resource, access and legal as critical determinants of e-government success. As in other studies, they broke down each of these factors into more detailed critical determinants forming a hierarchical success factors model. They used this model as a framework to empirically investigate the critical success factors in association with Bangladesh's e-government implementation. They based their conclusion on the difference between the respondents' perception of the level of importance and commitment regarding the attributes of each high-level success factor. Also based on the commonality found in e-government literature review, McMillan (2009) concluded with six eGSFs categories which further form a hierarchical eGSFs model.

From a different stance, several other "high-level" success factors were also suggested by Heeks (2008a) after observing many failures of e-government system projects and initiatives. Heeks suggested that the e-government decision-makers should evaluate the gap between what they want and the reality they currently have, regarding seven strategic factors: information, technology, processes, objectives and values, staffing and skills, management systems and structures, and other resources (time and money) – ITPOSMO. This model is a valuable method for predicting possible e-government failure, and therefore suggests to e-government decision-makers what decisions and actions should be taken. Based on previous reports, Heeks concluded that the larger the gap, the higher the probability of e-government development failing.

In sum, the literature review on categorising eGSFs into strategic or high-level success factors shows that reaching a consensus on how to categorise the eGSFs as they depend on context and adopted perspectives is most unlikely. Nonetheless, to some extent the categorisation models promote a holistic view of the complexity of e-government systems and success factors. The reviews show, depending on the viewpoint of the decision-makers, that the strategic-level model provides them with an understanding of the range and variety of the high-level success factors, assists them to properly prepare the e-government system development by evaluating some critical aspects, and supports them in comprehending the complexity and dependency of the strategic factors.

2.5 E-government in developing countries

2.5.1 Current state

Despite significant awareness of the wide gap between developed and developing countries in ICT development levels and a number of noteworthy success stories, "much of the developing world was still lagging far behind the developed world five years ago" (Davison et al. 2005: xv). There are numerous reasons to devote attention to the plight of e-government in developing countries, as has been done by many global institutions such as the United Nations (UN) and the International Telecommunication Union (ITU); it is, at least, necessary to help increase the possibility of sustainability of the e-government initiatives that have been implemented and to ensure the spending of scarce resources for this "inevitable luxury" could benefit the developing countries' citizens (Davison et al. 2005: 1).

In general, referring to the UN's e-government index 2010, the level of the current state of e-government in developing countries is medium to low (United Nations 2010). The index reflects the advancement of e-government in the member states of the UN which in turn is also a sign of the degree of benefit delivered to the citizens. The benefits are obtained by "better access to information, more efficient government management and improved interactions with governments, primarily as a result of increasing use by the public sector of information and communication technology" (United Nations 2010: 59).

The index ranges from 0 to 1 in which 1 is the highest and is measured by weight-averaging three separate indices: online service index, telecommunication infrastructure index and human capital index. Compared to the average of the world government index in 2010 which is 0.4406, the average index of the region occupied by most of the developing countries is around or below that figure, see Table 2.3, (United Nations 2010). The survey report also indicates that no developing country is in the first 20 highest on the index (United Nations 2010**:** 60).

Table 2.3 *E-government index of developing country regions*

Region	Index		Region	Index
Africa	0.2733		Southern Asia	0.3248
Caribbean	0.4454		South-Eastern Asia	0.4250
Central America	0.4295		Western Asia	0.4732
South America	0.4869		Oceania	0.4193
Central Asia	0.4239			

Source*: United Nations (2010)*

Further review of this UN 2010 report regarding each component of the index gives a more detailed view of the current state of e-government in developing countries. Despite 98% of the UN's member states having government websites and four developing countries within the top 10 of the online service index, the report also states that most developing countries have only a limited amount of online transaction services. The average online index – which measures the provision of basic information through online service, multimedia usage and two-way communication with citizens, the use of internet to deliver government services and solicitation of important public issues, and connected public services and public consultation for public decision-making – is 0.2823 for developing countries and 0.1121 for the least developed countries. The telecommunication infrastructure index, the second component, is measured by estimated internet users, main fixed telephone lines, mobile subscribers, personal computers and total fixed broadband, all per 100 inhabitants. The index for developing countries is 0.2046 and for the least developed countries it is 0.0445. The human capital index for developing countries is 0.8406 and for the least developed countries it is 0.5743, the index of which is measured by percentage of adult literacy and percentage of combined gross enrolment for primary, secondary and tertiary schools.

In general, this report provides a broad view about the achievement of the member states of the UN in developing e-government from the government side as the supplier of the services. In terms of the first measure, the online service, the index represents the extent to which government services are delivered through the internet or government websites. Especially the index measures the services associated with the ministries of education, finance, health, labour and social services. The measure refers to the UN's "four stages of online service development" (United Nations 2010: 95) which range from the simplest and least sophisticated to the most advanced service delivery, as presented in section 2.2.2.

A report delivered by Waseda University named "The 2011 Waseda University World e-Government Ranking" also shows that the advance of e-government in developing countries still lags far behind that of developed countries, which occupy the first to 22nd ranks (Anonymous 2011). By using seven indicators, each of which further uses three to ten dimensions, the report indicates that only some developing countries are grouped as having a middle score while most are categorised as having a lower score. Some of the developing countries in the lower-scoring group have been in that position for several years.

For 2008, by referring to the UN's 2008 online index measures and based on selected developing countries, Baqir and Iyer (2010) classified the African and most of the Asian countries' e-government development as *enhanced*, while the South American countries were at early *interactive* level. Concerning Africa, Rorissa & Demissie (2009) reported that in 2008 94.66% of African governments' websites, at most, are only able to provide downloadable information services. In addition to this still-low achievement of the governments in e-government development, the UN's report (United Nations 2008) provided figures indicating a low level of services utilisation of government websites by the citizens in developing countries. The figures are mostly between 34% and 66% and there are still many between 10% and 33%.

It can be summarised that, despite a small number of developing countries achieving a relatively high level of e-government development, most developing countries are still struggling to achieve even a medium level. In this respect, there is a need to facilitate and ensure that the efforts and resources spent by developing countries to develop e-government systems can be fruitful. By understanding the general status of e-government in developing countries, eGSFs relevant to their conditions might be identified, understood and modelled appropriately. At the same time, it is necessary to be aware that there are huge numbers of issues and challenges facing those countries.

2.5.2 Issues and challenges

In developing and implementing e-government, developing countries must confront a significant degree and range of issues and challenges that hamper a meaningful achievement of e-government development, for example "cost of technology, lack of infrastructure, limited human capital and weak private sector" (United Nations 2010: 4). Among those developing countries that have devoted their effort or have been assisted by other countries to develop e-government systems, they have had to experience various degrees of failure, despite the fact that examples of e-government success can also be found (Krishna & Walsham 2005). The failure rate was estimated up to 80% (Heeks 2003). This can mean total failure in that the development has never been implemented, or the e-government is implemented but then immediately abandoned, or it can indicate partial failure whereby the main goals cannot be realised (Heeks 2005b). For example, an e-government project in Vietnam called Project 112 started in 2001 and finally ceased in 2007 without achieving its intended goals (Obi & Hai 2010), or an e-government project to build a financial management system for the central government in India was finally closed in 2009 after seven years' effort due to significant partial failure (De' & Sarkar 2010). Specifically, many developing countries' e-government systems face sustainability issues (Islam & Alawadhi 2008; Kimaro & Nhampossa 2005; Krishna & Walsham 2005; Kumar & Best 2006; Nengomasha, McHombu & Ngulube 2010; Rashid & Rahman 2010) in the sense that the systems have been successfully developed, implemented and, at the beginning of their operational life, able to deliver services; then, however, the level of services deteriorates over time and finally stops.

The challenges for developing countries to develop an e-government system with fruitful achievements span a wide range of areas. Dada (2006), by reviewing e-government literature and implementing an ITPOSMO framework, showed the existence of significant gaps between the existing reality and the intended system among the e-government systems of developing countries. These included the ICT infrastructures, human resources, funding, IS development management, government readiness, politics, law and regulation, private sectors and so on.

The low level of telecommunication infrastructure as shown by the United Nations E-Government Survey 2010 (United Nations 2010) and limited internet penetration (International Telecommunication Union 2009) are significant challenges when developing countries seek to take advantage of and reap benefits from e-government. The immediate and clear impact of this condition is a low level of utilisation of online services in developing countries (United Nations 2008), which might also indicate that the services provided fail to reach the majority of citizens within that country. The available infrastructure is normally concentrated in urban areas, which causes a digital divide problem (Şandor 2006): that is, information “haves and have nots”. This means that only a small number of citizens can access as well as have skills enabling their effective use of internet technology; in turn, few citizens can get benefit from an e-government system (Şandor 2006) if the system is delivered through websites. Many of the developing countries experience this problem including Jordan (Alhujran 2009), Kazakhstan (Bhuiyan 2010) and Zambia (Joseph 2010).

Other significant challenges are that the education and the competence level of human resources (including government organisation leaders or managers, government employees and citizens) necessary for successful e-government system development and implementation cannot be satisfied. For example, knowledge about the potential benefits of an e-government system and a commitment to satisfying the requirements for successful e-government development and implementation on the part of many government decision-makers were not present in an e-government system development in Namibia (Nengomasha, McHombu & Ngulube 2010). Government employees with specialised competence in IS who are responsible for IS management are not fully available or cannot meet the level required to develop and implement an e-government system, such as was the case with e-government system initiatives in Vietnam (Obi & Hai 2010), Cambodia (Sang, Lee & Lee 2009), or Namibia (Nengomasha, McHombu & Ngulube 2010). Other challenges are a low level of ICT knowledge or a high resistance to using the e-government system amongst government employees who deliver government services to citizens (Obi & Hai 2010), and a low level of general ICT literacy among citizens (Abu-Shanab, Al-Rub & Md Nor 2010; Islam & Alawadhi 2008; Sang, Lee & Lee 2009). In addition, many developing countries also experience a high turnover rate of ICT-competent employees such as in Cambodia (Sang, Lee & Lee 2009), or a high dependency on overseas consultants such as in Namibia (Nengomasha, McHombu & Ngulube 2010). These human resource factors have been shown to have significant impact on the sustainability of implemented e-government systems (Best & Kumar 2008; Islam & Alawadhi 2008; Krishna & Walsham 2005; Nengomasha, McHombu & Ngulube 2010; Rashid & Rahman 2010).

The limited availability of funding not only reduces a government's ability to develop a successful e-government system but also raises sustainability problems (Best, Thakur & Kolko 2009; Kumar & Best 2006). More than that, the e-government initiative also causes competing priority issues in terms of whether e-government is really necessary for the country compared with basic need fulfilment (Sang, Lee & Lee 2009).

Some other countries that have been putting efforts to develop an e-government system have faced tough challenges in terms of managing IS development projects (De' & Sarkar 2010; Obi & Hai 2010), such as inappropriate requirement analysis or approach, or over-ambitiousness in scope and coverage. Further challenges may also come from legal and regulatory requirements for establishing e-government system service delivery (Basu 2004).

This review indicates a wide range of issues and challenges currently faced by developing countries that arguably prevent them from achieving a successful e-government development. In order to be relevant to their context, any study in eGSF should take these issues and challenges into account. Among many issues and challenges revealed from the existing literature, the e-government decision-makers or managers need to address themselves seriously to e-government systems sustainability and human resources.

2.6 Research gap and scope

2.6.1 Research gap

The review of the literature led to conclusions about the existence of research gap within existing research on the eGSFs. The main identified gap is:

- Lack of attention to the existence and importance of the relationships between the e-government system success factors, especially feedback-relationships.

The review of the e-government success model literature has shown that there was a limited amount of previous research of e-government success which address the importance of the relationships between the success factors in influencing the e-government system success. Many of the models developed by previous studies are unable to address the complexity of the e-government success factors in the sense that it is not only the individual and sets of factors that promote the successful design, development and delivery of e-government systems but also the relationships among the success factors. A similar conclusion was also observed by Titah and Barki (2008) when they reviewed the e-government adoption and acceptance factor literature. They recommended the conceptual development of the interactions and complex relationships of these factors from the multidimensional and multilevel point of view. The research that described the relationships of the success factors (AlShihi 2006**:** 214) did not present at an operational and detailed level how the factors relate to e-government success. The literature review indicates that a formal model which presents the relationships between the success factors in influencing e-government system success in systemic and dynamic fashion is very rare. This is especially more noticeable in the context of e-government system sustainability in developing countries.

Specifically, there is a paucity of research on the success factor relationships when the e-government success is associated with the review and maintenance stage of an e-government system life cycle. The existing literature indicated that much of the previous research on e-government success factors focuses on the development of e-government systems either from the supply or demand side. Previous research mostly investigated and modelled factors, attributes and measures that have led to e-government development success but ignored the fact that once the system is in operational phase various problems, application errors and new organisational or user requirements can come to the surface, threatening the development's success – a threat that can even discontinue service delivery in the worst case. The new requirements, originating from unavoidable organisational, legal or environmental change, need to be addressed and adopted over time. Lack of ability to address these issues could jeopardise a successful e-government system development that might have been achieved at the outset. Accordingly, a success factor relationships model that addresses e-government system success at the review and maintenance phase of the life cycle needs to be developed, especially a model that is relevant, appropriate to and useful for the conditions in developing countries.

In addition, much e-government success factors research does not pay much attention to the holistic view of high-level success factors (supply side). The literature review has indicated that the e-government system success factors are myriad and cover a very broad range. The existing literature has provided models of high-level success factors or categorised the success factors into constructs. However, research attention to a model that provides a thorough and holistic view of the success factors, especially from the supply side, is lacking. This research argues that the lack of a holistic view could mislead e-government systems stakeholders in developing and implementing the system to focus only on particular angles and ignore or be unaware of the existence of other factors that could also significantly affect success. Therefore, this suggests a need to develop a high-level success factors model capable of providing a holistic view from the supply side. In particular, attention to the external success factors is lacking.

2.6.2 Research aim and scope

Based on the identified research gaps, this study broadly aims at developing a formal relationship model of the eGSFs. It attempts to reveal how the success factors relate to each other in influencing the success of e-government over time. While the study tries to fill a gap, it is also a response to Scholl (2007: 75) who emphasised that "central research questions in EG [e-government]: ... , have to address their complex interrelationships and the processes between them", and underlines the importance of considering the impact of time or the dynamic nature of the relationships.

In addition to the identified research gaps, by referring to the e-government system life cycle this study will focus on the last phase of the life cycle: review and maintenance. Within this phase it is of paramount importance to ensure that the implemented e-government system can be maintained successfully to a particular level in such a way that it achieves the successful delivering government services over a long period as intended. Accordingly, by confining the scope to this phase this research assumes that the e-government systems, to a certain extent, have been successfully developed.

A formal model of the feedback relationships of more detailed success factors within this scope that lead to the success of e-government will be developed. However, to provide the context of the relationship model within the broad range of success factors, a high-level success factor model will first be developed; therefore, the model also acknowledges a holistic view of the success factors.

2.7 Summary

This chapter has presented a review of the literature concerning the nature and importance of e-government systems. The study defines e-government systems by embracing the IS concept underlining the importance of the socio-technical view. The description of e-government system forms, the stage model of e-government system advancement and types of applications acknowledges the fact that an e-government system is a complex phenomenon. This complexity was emphasised by previous research which indicated the existence of complex feedback relationships amongst various e-government system components. Though it is a complex system, an e-government success promises many significant benefits to both government and citizens.

The success of an e-government system has been defined by considering the importance of the sustainability of the quality system, information and services, highlighting the role of the review and maintenance phase in ensuring long-lasting success.

Various eGSF models, categorised according to level of abstraction of the success factors, have been reviewed. These models indicated that the success factors are numerous across a broad of aspects. The review has identified that much of the research considered e-government success factors influencing e-government success in a one-way and linear direction, distinguishing dependent and independent factors. Much of the previous research did not adopt feedback relationships of the success factors and the dynamic nature of the relationships within their models.

Most developing countries face more difficult issues and challenges than do developed ones because they have to invest in all three of the UN e-government development dimensions: "online services, telecommunication infrastructure, and education" (Popović, Atlagć & Kovacević 2001: 60). E-government system sustainability is one of the crucial problems facing developing countries. Among many factors, human resources have been shown to be of paramount importance in ensuring e-government system sustainability.

The next chapter, Chapter 3, will further review the eGSFs literature, revealing its wide dimensions and complexity. This section will present the wide range of factors or constructs that are deemed important in order to achieve e-government success and indicates relationships among them. Further, by drawing a particular eGSF from this broad dimension and associating it with the research aims and scope, Chapter 3 will also review and present the concept, elements and process of software maintenance. The interdependence within and among these elements and processes will also be identified using the literature from the software engineering field as its main source. Finally, Chapter 3 will address and examine the importance of human resources, and other factors associated with it, for e-government success, especially the internal employees with specific skills and knowledge. A well-known theory that was widely implemented to investigate human resource performance and other factors associated with internal personnel will be reviewed, and how these factors relate to each other leading to e-government success will also be sought and revealed.

Chapter 3 E-government Success Factors Relationships

3.1 Introduction

This chapter explores and presents the eGSF relationships indicated in previous research. The review of various existing literature focuses on identifying the relationships between the identified eGSFs. This review will identify these relationships from a variety of previous (separate) studies and put the identified relationships together to reveal how success factors relate to each other in influencing e-government success. This is in response to the identified gap found in the previous chapter that there has been a lack of attention in previous research with regard to the importance of the relationships between the eGSFs. The literature review of the eGSFs relationships is organised as follows.

This chapter first explores the existing eGSFs literature to reveal the broad nature of eGSF dimensions and the existence of complex relationships among the eGSFs. Along with reviewing previous literature, the beginning of this chapter also attempts to organise the dimensions of the identified eGSFs and their relationships according to whether those eGSFs are high level or low level. The description of the eGSFs' dimensions provides context to a specific dimension study. Based on this presentation, two specific subsets of the eGSFs' broad dimensions are chosen as the focus of this eGSFs relationships study along with the justification for this selection. These subsets are software maintenance and information systems staff domains.

Next, software maintenance concepts, elements and processes are presented and reviewed. Factors and their relationships within the SM domain that determine software availability over time are revealed from the existing literature. As described in the e-government success definition adopted by this study, success is defined through e-government system sustainability and one of the determinant components is software availability over time, or software sustainability.

Finally, the success factors and relationships from the IS staff domain that influence SM performance are presented and reviewed. However, as any study related to human resources will cover very broad aspects, this study focuses its literature review mainly on factors related to motivation, competence and availability.

3.2 E-government success factors

3.2.1 Introduction

This section presents the identification and organisation of the eGSFs, which were identified through the reviews of the literature on e-government success. In this identification, the selection of the eGSF literature was not confined to the working definition of e-government success, nor to the issue of sustainability. The selection was also not limited to the literature that is associated with e-government in developing countries. This broad selection is aimed at obtaining a wide insight from various resources; therefore the review and identification did not focus on a particular domain.

The review has been able to identify a large number of eGSFs which can be viewed as originating from a number of different domains. Considering this view, the identified eGSFs are then categorised in such a way that a category represents a particular high-level eGSF. Using the existing e-government literature, the organisation of the eGSFs is undertaken in a top-down manner. Accordingly, the presentation begins with describing high-level success factors and is followed by the description of the success factors within each of these categories.

In addition to the identification and organisation of the eGSFs, this section also presents possible relationships among them, as indicated by previous e-government literature. This is considered important for giving the preliminary broad insight that, if the identified relationships are put together, complex relationships between the eGSFs emerge and are observable. However, as the identification covers a broad range of eGSFs, the possible relationships are not pursued in detail.

3.2.2 High-level success factors

The literature review in Chapter 2 has shown that a number of high-level success factors have been used to conceptualise the eGSFs; but it suggested the need to organise the eGSFs in a relatively simple way that is nonetheless capable of providing a holistic view from the supply side. In response to this need and upon reviewing the e-government literature related to e-government success and failure, adoption and diffusion, this study argues that there are four high-level categories of eGSFs.

The first category is derived from AlShihi's (2006) description regarding issues and outcomes of the e-government system life-cycle. AlShihi argued that there exist adoption success factors that play critical roles and provide fundamental support from the inception of an e-government system through the rest of its life-cycle phases. The existence of these success factors is crucial and cannot cease at any time if e-government success is to be ensured. This category encapsulates all success factors that drive the initiation and govern the development and operation of an e-government system, leading to its success.

The second and third categories are inferred from Beynon-Davies's (2007) proposed meta-model of e-government. The meta-model consists of two dimensions: vertical and horizontal. The vertical dimension encapsulates "human activity infrastructure and informatics infrastructure" and the horizontal dimension sums up two inter-related aspects: "business model and value-creating system" that are associated with human activity infrastructure (Beynon-Davies 2007**:** 8). These two proposed dimensions of e-government are used to encapsulate the eGSFs that correspond to the vertical and horizontal dimensions, hence the second and third high-level eGSF categories. The second category associates all success factors necessary to ensure the achievement of a quality IS. They consist of success factors in relation to internal ICT infrastructure management to support activities in delivering services and values to the consumers of an e-government system. The third category summarises all success factors required to attain quality e-government service delivery. These success factors are concerned with creating and delivering value and services to the government's customers.

The fourth category summarises all success factors associated with the external environment of an e-government system. Much of the previous literature (Heeks 2005b) has indicated that these success factors, which deal with external entities or activities, play an important role in the success of an e-government system. The existence of these entities or activities is not part of government organisations.

These four categories of success factors can be named: Driving and Governing Factors, Information Systems Management, Service Delivery Management and External Environment Factors. These high-level success factors can be broken down into their corresponding component success factors.

3.2.3 Driving and governing factors

A number of success factors identified by previous studies can be categorised as Driving and Governing Factors.

Most research indicated that leadership is one of the most important success factors (Furuholt & Wahid 2008; Krishna & Walsham 2005; Lal & Haleem 2009; Luk 2009; Rashid & Rahman 2010; Weerakkody, El-Haddadeh & Al-Shafi 2011) for any type of e-government system at any level. This is so because the leadership makes “go or no-go” decisions and determines the direction of an e-government system (Weerakkody, El-Haddadeh & Al-Shafi 2011). Examples of attributes that indicate the importance of this success factor are strong political leadership (Furuholt & Wahid 2008; Lal & Haleem 2009) and transformational leadership (Shi 2002). In addition to its existence, the leadership success factor has been shown to have a significant influence, directly and indirectly, on many other factors (Lal & Haleem 2009; Prybutok, Zhang & Ryan 2008). Krishna and Walsham (2005) showed that leadership influences e-government system sustainability. It also influences organisational culture and structure (Furuholt & Wahid 2008). Willingness of the leaders at various levels to reduce their authority (Hesson 2007) will affect the willingness of the staff to reduce their authority also in delivering services to citizens (Bhatnagar 2002; Hesson 2007).

In addition to leadership, politics has also been considered as a significant factor for e-government system success (Furuholt & Wahid 2008; Gauld 2007; Gil-Garcia 2006; Gil-Garcia & Pardo 2005; Jansen 2011; Weerakkody, El-Haddadeh & Al-Shafi 2011). For example, political pressures and interference with the e-government system development process have been identified as one of the factors that caused an e-government system failure in New Zealand (Gauld 2007); political parliamentary support is required for an e-government system initiative (Furuholt & Wahid 2008). As one of important success factors, politics is closely related to the leadership success factor (Krishna & Walsham 2005).

Politics also affects law and regulation enactment (Weerakkody, El-Haddadeh & Al-Shafi 2011).

Much e-government research has indicated that law and regulation is an essential success factor for e-government success. It is required by the government, for example, to adopt new technologies (Anonymous 2005; Bhatnagar 2004), to establish a new e-government system unit which is a means of institutionalising an e-government system (Furuholt & Wahid 2008) and to create a new mode of government service delivery (Luk 2009). It is also needed to stipulate standards regarding ICT, data and information, and procedures (Gil-Garcia & Pardo 2005), or to ensure compliance with privacy and security requirements (Basu 2004). From this example, it can be inferred that law and regulation can influence the design of an e-government system. In addition, for government organisations, law could be used or even required to justify funding which, along with institutionalisation, can ensure the sustainability of e-government.

The role of funding for e-government system success is obvious. On a broad level, there is a significance difference in the achievement level of the UN's e-government development index between high-income and low-income developing countries (United Nations 2010). Examples of funding success factor attributes are: justification for return on investment of fund spending (Coursey & Norris 2008; Schwester 2009), finance availability and sustainability (Kumar & Best 2006; Weerakkody, El-Haddadeh & Al-Shafi 2011) and innovative funding schemes, especially within the private sectors (Heeks & Davies 2001; Hosman & Fife 2008). In affecting the success of e-government systems, funding is closely related to the strategic planning success factor as well as to many other success factors.

E-government system strategic planning is another very important success factor (Anonymous 2005; Bhatnagar 2004; Prybutok, Zhang & Ryan 2008; United Nations 2010). One attribute of this factor is a national ICT strategy (United Nations 2010), which might depend on law and regulation and political support. The strategic planning should address the scale of the project (Bhatnagar 2004; Gauld 2007), which certainly affects funding. It also should include the arrangement of stakeholders' and agencies' relationships (Bhatnagar 2002; Luk

2009), a clear-cut plan (Bhatnagar 2004), and explicit development milestones and measurable deliverables (Gil-Garcia & Pardo 2005; Prybutok, Zhang & Ryan 2008), as well as consideration of the sustainability issue (Furuholt & Wahid 2008; Kumar & Best 2006; Wood-Harper, Ibrahim & Ithnin 2004). This strategic planning success factor may well be influenced by the e-government system leadership (Prybutok, Zhang & Ryan 2008) and leader's vision.

Vision is an e-government system success factor normally attributed to the e-government system leader. This factor's attributes include a clear vision (Furuholt & Wahid 2008; Krishna & Walsham 2005), a realistic vision (Gauld 2007) and a shareable vision (Krishna & Walsham 2005) on the part of the leader.

In sum, this Driving and Governing factor can be broken down into lower-level e-government success factors: leadership; politics; law and regulation; funding; strategic planning; and vision. The significant role of the attributes of these success factors in achieving e-government success has been shown by many previous studies. While these success factors directly or indirectly influence e-government success, they also affect or are affected by other success factors, thus indicating the existence of relationships between the success factors.

3.2.4 Information systems management factor

Referring to McNurlin and Sprague Jr (2006, Ch. 1), IS management associates with information technology (IT) infrastructure; IS development and delivery; IS staff; and IS organisation.

The IT infrastructure that includes software, hardware and computer networks is a necessary condition in order for government processes and services delivery to become an e-government system. The attributes of this factor that can lead to e-government system success comprise the degree of technology complexity (Schwester 2009), appropriateness of the chosen technology to the goals and objectives (Bhatnagar 2004), and availability of the software and hardware to support the e-government system, as well as reliability of the computer network technology (Best & Kumar 2008; Coursey & Norris 2008; Hussein, Karim & Selamat 2007). These attributes arguably relate to other success factors. Appropriate technology selection depends on the e-government system goals and objectives embodied in the strategic planning. Further, the software and hardware availability will depend on the funding, and the computer network reliability will influence service delivery (AlShihi 2006; Best & Kumar 2008).

There has been much research on the development and maintenance of e-government application systems to identify the attributes that lead to e-government success. That research has found that the system should be designed such that the users can easily use it (Gil-Garcia & Pardo 2005) and improve their attitude and skills towards the application system itself (Heeks 2005b); it should also involve the system users during the application system development (Heeks & Davies 2001). The system is also expected to produce quality information (Hussein, Karim & Selamat 2007). To achieve success in an e-government system project development, which is expected to result in an application system as intended, Rosacker and Olson (2008) found empirically that project mission, top management support, project schedules, etc. are important success factors. Inevitably, any implemented e-government systems need continuous maintenance (Cheng, Huang & Peng 2010; Moon 2002), which can involve huge amounts of resource over years (Pokharel & Park 2009; Tolbert, Mossberger & McNeal 2008).

Another very important success factor is staff with specialised competence in ICT. For instance, Coursey and Norris (2008), Reddick (2004), Gil-Garcia (2006) and Schwester (2009) considered that the availability of an appropriate number of full-time staff significantly influences the success of local e-government systems in the US. Further, the availability of appropriate skills for ICT problem resolution among internal ICT staff is a necessary success factor (Heeks & Bhatnagar 2001; Hussein, Karim & Selamat 2007; Lal & Haleem 2009; Rashid & Rahman 2010). Also, training to maintain ICT staff competence is necessary (Furuholt & Wahid 2008; Gil-Garcia 2006). On the other hand, turnover of this type of staff (Sang, Lee & Lee 2009) and a dependence on expatriate ICT-skilled employees (Heeks 2005a; Nengomasha, McHombu & Ngulube 2010) could jeopardise the sustainability of the system.

Furuholt and Wahid (2008) found that the establishment of an e-government system unit specifically to organise or manage the IT infrastructure, IS project development and IT staff plays a significant role in realising e-government success. The managerial capability of IT management is considered to be one of the success factors by Kamal (2006). The organisational factor has been empirically found to be significant in influencing e-government success, and this factor is affected by an institutional arrangement which in turn is influenced by demographic, political association and economic conditions (Gil-Garcia 2005). Another example of the success factor within this category and its relationship with other success factors was empirically shown by Hussein et al. (2007), who found that resource allocation has positive relationships with system quality, information quality, perceived usefulness of the system and user satisfaction.

In sum, various low-level success factors within Information Systems Management can be organised into IT infrastructure, application system development and maintenance, internal ICT-skilled staff and IS organisation. The reviews of literature have shown that success factors within this high-level factor relate to each other and eventually determine e-government success. These success factors are also influenced by those from other high-level success factors.

3.2.5 Service delivery management factor

Normally, an e-government system is utilised by government organisations or units to deliver services to citizens. Various success factors found in the extant research within this category can be associated with the organisational success factor. As is widely acknowledged in the management IS field, in order to realise e-government success, most likely an implementation of an e-government system requires organisational changes. For example, change management is required (Grabow, Druke & Siegfried 2004; Lal & Haleem 2009) to ensure that the organisational change due to the implementation of an e-government system can take place with less cost. The implementation of an e-government system will affect and be affected by the organisational culture (Furuholt & Wahid 2008), the organisational structure (Furuholt & Wahid 2008; Heeks & Davies 2001) and the organisational diversity (Gil-Garcia & Pardo 2005). Proper implementation also needs collaboration and coordination among the government's departments (Coursey & Norris 2008; Ndou 2004), which in turn necessitates departmental ownership of the e-government system (Bhatnagar 2002). Change management will be influenced by the leadership factor, and collaboration and coordination depend on a shared vision (Luk 2009). Organisational diversity affects the application and computer network system design, which in turn impact the system quality (AlShihi 2006) as well as funding.

In addition to the organisational success factor, much of the previous research also found that the human resources operating the e-government system to deliver services is a critical success factor. In relation to this success factor, among many attributes, staff's loyalty or resistance (Anonymous 2005; Bhatnagar 2002; Coursey & Norris 2008; Furuholt & Wahid 2008; Gil-Garcia & Pardo 2005; Krishna & Walsham 2005; Schwester 2009), employees' fear of being fired and stress about doing new jobs (Heeks & Davies 2001) and staff's willingness to reduce authority when dealing with citizens (Bhatnagar 2002; Hesson 2007; Ndou 2004) are considered some of the most important ones, mainly in developing countries. Extant literature has indicated how to overcome this problem. For example, the staff should be involved from the very beginning of the e-government development process (Furuholt & Wahid 2008; Gil-Garcia & Pardo 2005; Papantoniou et al. 2001; Wood-Harper, Ibrahim & Ithnin 2004) and provided with incentives (Anonymous 2005). In addition, how staff values formal information resulting from the e-government systems (Heeks & Davies 2001) and the degree of uniformity of users (Gil-Garcia & Pardo 2005) also affect e-government success. Further, staff literacy or skills in ICT (AlShihi 2006; Furuholt & Wahid 2008; Heeks 2005a; Krishna & Walsham 2005; Ndou 2004), especially at managerial levels or among key staff (Furuholt & Wahid 2008; Hussein et al. 2007; Ndou 2004), are other success factors related to human resources. Therefore, training for staff is necessary not only for e-government applications (Bhatnagar 2002) but also for service quality (Furuholt & Wahid 2008).

In brief, from reviewing the success factors within this high-level factor category, this study concludes that there are two lower-level success factors: organisation and service delivery human resources. Past studies have also indicated various relationships between the success factors within these two categories, as well as with other success factors in other categories.

3.2.6 External environment factor

The most important external entity relative to an e-government system is the citizens (and business) that will ultimately determine whether or not an e-government system achieves success. Among many aspects that relate to the citizens are awareness about the existence of e-government system services, and understanding the benefits of an e-government system. For this purpose, marketing activities are important success factors. They are needed so that citizens are aware not only of the existence of the e-government system but also of new ways to deal with the government (AlShihi 2006; Bhatnagar 2004; Ndou 2004; Wood-Harper, Ibrahim & Ithnin 2004). In addition, educating the public about the benefits of using the e-government system (Druke 2005; Heeks & Davies 2001) is an important success factor. Further, Tan and Pan (2003) found that involving citizens during the development process is an important success factor. In this case, the leader of the e-government system initiative considered that “customer” feedback is an invaluable factor for e-government success. Citizens who are aware and understand may adopt and use the system to obtain government services and then be satisfied. However, recurrent use of the e-government system depends on many factors, such as the degree of usefulness of the system, which in turn depends on system, information and service quality (DeLone & McLean 2003).

Some research showed that cooperation with private sectors is a significant factor in achieving e-government success. A variety of forms and aspects of cooperation can be designed and implemented which include funding, expertise and training provision by private companies (Grabow, Druke & Siegfried 2004). Also, private companies may develop and maintain the infrastructure and system, receiving a reasonable fee from government—citizen transactions (Gil-Garcia 2005; Hosman & Fife 2008; Krishna & Walsham 2005). Furthermore, private companies or other organisations could help the government by providing consultancy regarding the development of e-government (Heeks 2005a). Cooperation between government and private companies is normally implemented as a contract legally binding to both parties. Therefore it requires the availability of a legal framework which, in turn, needs political support. In addition, the government also needs skilled staff with understanding of both legal and technical aspects.

Mainly in developing countries, the regular availability of electricity and the availability and reliability of the communication network infrastructure become important success factors for e-government (Heeks 2005a; Heeks & Davies 2001). Moreover, the ability of the government to cope with ever-changing ICT is another factor in e-government success (Heeks & Davies 2001; Schwester 2009). A lack of general ICT infrastructure greatly contributes to a low level of e-government achievement in developing countries (United Nations 2010). ICT advancement adversely affects the competence level of ICT staff and the IT infrastructure. This means regular training for ICT staff is necessary, which in turn depends on funding. Regular upgrading is also necessary for the infrastructure, which also means a funding expense.

Factors such as the "me too" attitude of government leaders could play an important role, negatively or positively, in achieving e-government success (Heeks & Davies 2001). The demands of citizens or private companies for e-government could encourage the government leader to attain success in e-government. On the other hand, citizens could be unwilling to change the way they access government services, hence preventing e-government from successful development and implementation (Coursey & Norris 2008).

General economic conditions as well as demographic aspects also determine e-government success (Gil-Garcia 2006). Leenes (2004) explained that, for the local government level with few inhabitants, the development of a complex e-government system is economically an unbearable burden. In other words, a local government level with a larger population is more likely to be successful in terms of the delivered services (Schwester 2009). Greater population can mean more economic potential capability, which also means more tax collected, therefore more funding availability. Greater population can also mean a larger work volume for government staff; therefore it is more urgent to streamline the processes and procedures.

To summarise, the high-level external environment factor can be further divided into several low-level success factors: marketing and education to the "customer"; general electricity and ICT infrastructure; cooperation arrangements with private companies; socio-economic conditions; and other success factors. The review also indicates co-dependencies of the success factors within this high-level category and with other factors in other categories.

3.2.7 Software maintenance and information systems staff factors

The previous sections have presented and suggested an organisation of a wide range of eGSFs and their relationships. Considering this wide range of success factors and the limited resources and time available, detailed investigations of the relationships between all of the eGSFs will be unfeasible. The usefulness of this kind of study is also questionable. Therefore, assisted by the organisation of the eGSFs provided in a previous section, this study will focus on two specific low-level success factors. This subsection describes these factors and presents the reason for their selection.

First, this study will focus on the success factors within the Information Systems Management high-level factor. As was argued in the development of the e-government working definition, the e-government decision-makers or managers must be aware that e-government is an IS which integrates a variety of different interdependent elements, in addition to paying attention to a variety of other specific success factors. So, this study argues that the management of these interdependent elements of the IS is necessary to guarantee e-government success.

Second, as suggested by the identified research gap and the critical role of internal ICT-competent staff, especially for developing countries, this study further narrows down its scope into two low-level success factors: the software maintenance and IS staff management. These are two subsets of the Information Systems Management high-level success factors within which the elementary level of eGSFs along with their relationships will be investigated. The reason for choosing these two will be elaborated as follows.

Software is a necessary condition for any e-government system. Heeks (2005b) considered software as an integral part of his ITPOSMO design-reality gap framework. In order to ensure e-government success, Heeks urged e-government decision-makers or managers to evaluate the gap between the design and reality of the e-government technology – software, hardware and network – among seven high-level success factors or ITPOSMO. If the gap is wide, policy and actions should be taken before the e-government development is continued. Further, Heeks emphasised that it is necessary for an e-government system to have software which possesses characteristics enabling the achievement of e-government success, in addition to other important success factors. As a necessary condition for an e-government, the role of various different types of e-government software was also highlighted by Ebrahim and

Irani (2005) in their proposed e-government architecture. In this proposal, in which a number of types of e-government software connect the e-government infrastructure and many different e-government user interfaces, they showed the indispensible role of e-government software in aligning government service delivery processes and transactions. They also showed that optimum implementation of e-government software enables efficient and effective communication, data and information sharing and processing, and collaboration across the government's units as well as allowing efficient information retrieval and transactions for the e-government users.

As was indicated in the review of previous studies and referring to the e-government life-cycle presented in Chapter 2, the software life-cycle consists of two different stages: development and maintenance. However, as reviewed in the previous section and identified as a research gap, most eGSFs studies focus on the development stage of e-government but lack attention to the success factors in association with or as part of the review and maintenance stage; see Tsai, Choi and Perry (2009) and Patel and Zu (2009) for further examples. More specifically, there was a lack of attention to e-government systems maintenance, that is, when the e-government application systems software needs some changes or correction (Heeks 2005b).

On the other hand, it was found and concluded by Melin and Axelsson (2009), after studying two e-government case studies, that systems maintenance is an important success factor and attention to maintenance should be paid from the early stage of an e-government project. The importance of this stage was also emphasised by Heeks (2005b: 260): "maintenance is not only a management challenge, but an activity that begins to absorb more IT staff and budget than new systems development." Further, "[c]ompared to software development, software maintenance has more impact on the well-being of an organization" (van Vliet 2008: 462).

Few e-government studies are known to have addressed e-government SM. Among these few are Gibson and McGaley's (2008) study on an e-voting system. The study was undertaken as they acknowledged the fact that this system is prone to human error and creeping user requirements, and should always accommodate the introduction of new laws, election types or technology. This system therefore requires SM, which should result in a trusted system. In their study, in order that the e-voting system is trusted, they proposed that as a result of implemented maintenance the system should be certified. Gibson and McGaley's study focused on creating and implementing maintenance standards such that the resulting e-voting software can be trusted. From a different perspective, inability to maintain the software can result in the abandonment of an e-government system.

In addition to e-government SM, staff with specialised skills in IS along with its various attributes has been well acknowledged to be one of the e-government success determinant factors by many of the previous studies reviewed in the previous subsection. For further example, Heeks' (2005b) ITPOSMO places the staff and their skills in an e-government system as one of the key eGSFs that must be evaluated to ensure that the gap between the system design and its reality does not jeopardise e-government success. However, the relationships of eGSFs within this domain and with other eGSFs within the SM domain have not been explored.

By focusing on these two comparatively low-level eGSFs, a relatively elementary level of the eGSFs will be explored and reviewed, and the relationships of the eGSFs within and between these two domains will also be examined and reviewed.

3.2.8 Summary

This literature review has shown the wide range of eGSFs, which can be organised into four broad or high-level success factors: Driving and Governing factor; Information Systems Management, Service Delivery Management, and External Environment factor. The review also suggests that each of these high-level success factors can be further organised into low-level categories. At this low level, various possible elementary success factors from different perspectives have been identified. As also underlined by Druke (2005) in regard to the set of success factors for local government identified by Grabow, Druke and Siegfried (2004), various eGSFs are not equally critical, although they are important in achieving e-government success. Priority should be given to the eGSFs which are critical in a particular situation or level of e-government, while at the same time being aware of many possible success factors that might affect that critical eGSF.

In addition to organising the eGSFs, this review has indicated the existence of relationships among the eGSFs and underlined the complexity of the relationships between them, which in previous studies have not been much explored. The relationships indicated by this review link various success factors between different high-level or low-level success factors, as well as within the same high-level or low-level success factors. Further, Druke (2005) stressed that it is the relationships between the eGSFs that can make them play their roles effectively.

These eGSFs identified from previous studies have indicated that they relate to each other, they form a large set of eGSFs as a whole, and the eGSFs have different leverage levels depending on the context, stage or level of the e-government system. Therefore, based on these attributes the eGSFs characterise themselves as a system and the high-level eGSFs can be considered as subsystems (Myers & Kaposi 2004). Considering this broad coverage of the eGSF system and the restrictions of resources and time, studying the whole system will be an impossible undertaking. Guided by the aims and scope of the study, this research will be focused on two relatively low-level success factors or sub-subsystems:

- software maintenance; and
- information systems staff.

These two low-level success factors are subsets of the IS management high-level success factor. The identified success factors along with their relationships become a subsystem of the broader eGSFs system which, from this subsystem perspective, is expected to be able to explain e-government success.

3.3 Software maintenance

3.3.1 Introduction

This section presents the literature review of SM, which aims at understanding SM itself; reasons for maintenance necessities; differences between software development and maintenance; process, classification and phases of maintenance; and impact of maintenance. Having reviewed these aspects of SM the elements, processes and relationships within and between these two can be mapped to understand how SM can ensure e-government success through the dynamic of e-government software availability.

The review concerns SM in general and is not specifically associated with e-government systems. However, as an e-government system is a subsystem of a wider IS, the review also applies to any e-government system.

3.3.2 Role of effective software maintenance for e-government success

As mentioned in the previous chapter, as well as referring to the working definition of e-government success one of the focal points of that success in developing countries is sustainability. Previous research into e-government sustainability in a developing country has shown that one important factor for this sustainability is the sustainability of e-government IT (Kumar & Best 2006), which consists of software, hardware and network.

Within the IS management area, the critical importance of focusing management attention on the software and its sustainability has been well-acknowledged by much of the literature. For example, in terms of the costs to organisations of software for the whole system life-cycle, van Vliet (2008**:** 3) stated that:

> So the *cost* of software is of crucial importance. This concerns not only the cost of developing the software, but also the cost of keeping the software operational once it has been delivered to the customer. In the course of time, hardware costs have decreased dramatically. Hardware costs now typically comprise less than 20% of total expenditure ... The remaining 80% comprises all non-hardware costs: the cost of programmers, analysts, management, user training, secretarial help, etc.

Using the description of a sustainable quality e-government system in section 2.3.2, e-government software sustainability means the software is available to perform its required purpose – that is, delivering government services for at least a specified period of time after the software is developed and implemented. In other words, this definition is about software availability over time. Software availability is defined by Pressman (2005**:** 763) as the probability of the software to perform its intended function at a particular time. Based on this definition, logically, in order to be available the required software must exist in the sense that it has been developed and is usable; the software's features to perform the intended functions or to be used to deliver a (set of) particular service(s) must also exist; and the existing features must be able to be used for the proper delivery of its intended purposes as required. Practically, software availability over time means that the e-government system "can adapt and change with demand" (Elliman, Irani & Jackson 2007**:** 370) over time, in addition to successfully overcoming system errors that may occur while delivering service. In sum, e-government software can be unavailable because of software errors, or the software features to deliver services not being available.

This description of software availability, which extends from the usual notions of software availability that are limited to technical up-time or down-time due solely to software error (Xiong, Xie & Ng 2011), has also been adopted and implemented in evaluating IS service availability as studied and reported by Gaj and Germani (2008). They argued that systems availability is increasing in its level of importance and pointed out that a measurement of availability of an ICT-based system by using the system's up-time is no longer sufficient and therefore should be extended to a wider concept of service availability. To measure service availability within a particular time interval, they split services into subservices, and each subservice in turn was divided into subservice components. They formulated service availability as a function of the down-time of subservices and subservice components. They also identified that maintenance of hardware, software microcode and application software plays an important role in minimising the down-time. By referring to the e-government system life-cycle, achieving sustainable software is especially associated with, but is not limited to, the last phase of the life-cycle when SM activities take place. The successful maintenance activities are necessary in order to match with the required level of software availability over a particular period of time. The fulfilment of this requirement ensures longer lasting e-government success. Without maintenance, e-government software will become obsolete in the face of the ever-improving requirements and will be abandoned soon after implementation, hence failing to operate and deliver services. This leads to e-government failure.

3.3.3 Understanding software maintenance

Definition. There is much literature providing definitions of SM: for example, Edwards (1984), Swanson and Beath (1989), the IEEE Computer Society (1998), and Pigoski (1997). Pigoski (1997**:** 9) provided some lists of this definition which he considered "after-the-fact" maintenance oriented, that is, the maintenance that takes place once the software is ready to be used to deliver organisational services. There is no basic difference between these definitions. This study adopts the SM definition described by the IEEE Computer Society (1998**:** 4), which defines SM as "modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment." This definition indicates that once a software development project is completed and the systems start to operate, changes to the software are inevitable due to problems within the software itself or emerging new requirements from the system users or the organisation. The ability to satisfy the required modification is necessary to ensure that the system is able to operate and deliver services as was expressed by the system design created prior to software development and according to users' requests for improvements and to organisational needs. Over time as the business process changes, the software should be able to accommodate these changes (Lehman 1997 in Lehman et al. 1997).

The difference between software maintenance and development. In many cases, SM means new software development, but much smaller in the scale of activities or introduced software size. To clarify the difference between maintenance works and new software development, Abran and Nguyenkim (1993) characterised maintenance as:

- requiring mainly only one or two resources
- occurring randomly or not planned
- firstly reviewed to determine its priority
- managed under the annual budget
- putting the requests in a queue
- not using project management techniques to manage workload
- being more "user-service-oriented" and "application-responsibility-oriented"
- capable of increasing funding to raise service level but not to support a specific request and
- needing less than 60 person-days to deliver a solution for a maintenance work request.

By comparing them to existing software and using quantitative criteria, Boehm (1981) considered software engineering activities as maintenance if they redesign or redevelop less than 50% of software products, or design and develop software interfaces amounting to less than 20%, or modify software codes, documentation or database structures. Furthermore, IS maintenance: a) requires a high degree of functional integration of the system, because maintenance staff should deal with a number of different aspects of the system; b) encompasses "hardware, system software, application software, database system, security procedures, users and other elements"; and c) can be triggered by external users, other elements of the IS and by the elements of the system itself (Edwards 1984).

In essence, SM is unplanned changes of small-scale software developments based on and implemented to the existing and operating software.

The significance of software maintenance. In addition to its significant role in ensuring longer-lasting systems, the significance of the maintenance stage within the IS life-cycle can be seen from the budget allocated for this purpose. Previous studies in IS maintenance in general have reported that a large portion of IS budgets was allocated to maintain the system (Kaplan 2002 in Hoffer, George & Valacich 2005; Klint, van der Storm & Vinju 2010). For example, the estimation of maintenance costs over time ranged from 35% in the 1970s to 90% in the 1990s in Bhatt et al.'s paper (2006). Considering overall IS budget allocation, system maintenance accounted for 60%–80% in Huff's research (1990). Out of a total IS department budget, 57% was for department operations and 33% for system and programming and, out of this 33% budget, 70% was for maintenance in McNurlin and Sprague Jr's book (2006). Although these figures are for the SM of IS in general, they also inform the degree of significance of e-government SM.

3.3.4 Maintenance requests

Causes of software maintenance. Any SM activities start from a maintenance request (MR). This study assumes that any need for maintenance, either to correct faults, to improve functionality or to adapt to a new environment, will always materialise as a maintenance request (MR). An MR is a request lodged by IS users, either internal or external with respect to the organisation, or the software maintainers themselves, with those who are responsible for the SM to undertake the requested SM.

There are several reasons for software needing maintenance:

- software development cannot produce error-free software
- ever-changing business-rule requirements and environment
- incomplete and new user requirements
- IT innovations and advancement
- SM activities themselves (Grubb & Takang 2003**:** 25)
- an incremental software development approach (Boehm & Lane 2007; Tan et al. 2009).

It is well understood that software defects or bugs can escape from the testing stage of software development (Abdel-Hamid & Madnick 1991; Pressman 2005). This volatility, caused by undetected errors during the development stage, makes an information system subject to failure from time to time. Quite often, new business problems or new business requirements emerge during the post-implementation stage, which require system adjustments or enhancements. Also, there are many occasions when users are unable to express their requirements clearly and completely during development, or users need a more user-friendly interface to make their job easier. Maintenance becomes particularly important for a web-based IS (Hoffer, George & Valacich 2005). As a website should be available continuously over time, persistent maintenance and improvements are necessary. Preventing the website from being down and having broken links, validating web-pages before publishing, re-registration to search engines, recreating user interfaces and creating future editions as well as updating information accessible from the website are maintenance activities related to an e-government's website. The evolutionary nature of hardware and software technology also affects current software used by an organisation. The IT environment where current software is operating might cause a necessary adjustment to the software in order to keep it running smoothly. The ever-changing nature of business requirements and IT place environmental pressure on e-government software, so that even a maintenance activity conducted on the current software may trigger other maintenance needs, especially the quick-fix ones. In addition, the incremental approach in IS development, which is currently the most adopted approach, necessitates continual maintenance over time.

This review indicates a number of sources for MRs, which can be categorised as:

- software development
- environmental pressure and
- the maintenance itself.

This source of SM influences software availability in delivering the required e-government services. Inability to respond and cope with MRs causes the degradation of software availability, hence the usefulness of the e-government. To a certain extent, this degradation can cause the overall e-government to be unavailable to deliver the majority of its intended services, hence rendering it unsustainable. The ability of the organisation to respond and cope with MRs, among other things, greatly depends on SM or IS staff which, together with its associated factors and relationships, will be explored in section 3.4.

Software complexity and MRs. The complexity of software depends on the design and implementation of software project development, as well as on completed maintenance accumulated over time. Much research on software engineering has shown that the degree of software complexity affects the number of software faults, which in turn influence the level of MRs.

There are various IS attributes associated with software complexity, such as computing distribution, computing environment and component variety (Schneberger 1995). Especially in terms of the software, its complexity can be determined from its dynamic complexity, coordinative complexity (Banker, Davis & Slaughter 1998), source line of code (SLOC) or function points (Ahn et al. 2003). Because of its simplicity and intuitive meaning, the number of SLOCs has been investigated by many studies to predict the number of faults, such as by Andersson and Runeson (2007), Jorgensen (1995), Lucia, Pompella and Stefanucci (2005), Ostrand, Weyuker and Bell (2005) and Zhang (2009). Although the evidence varies across the system software investigated as to the existence of the relationships between the number of the SLOCs and the number of faults, Andersson and Runeson (2007) indicated that within some software applications the SLOCs can be used to predict faults. Other studies investigating other software have supported this claim. Zhang (2009), using the Weibull distribution model, showed that there is a relationship between the SLOCs of the software module and the number of defects. Zhang's study demonstrated that "larger modules tend to have more defects": the first 20% largest modules contain 60.62% of post-release faults and 70%–95% of the total number of faults can be estimated by the defect density of the top 10% largest modules. Ostrand, Weyuker and Bell (2005: 346, 9), by employing a negative binomial regression model, showed that "the number of faults is proportional to the number of lines of code" and claimed that, even using their highly simplified model with only SLOC data as the predictor, "the model generally predicts a large number of faults for large files." This simplified model formulation is

$$\text{number of faults} = \exp(-1.636 + 1.047 * \log(\text{SLOC}))$$

while the complete formulation of this negative binomial regression model comprises the number of faults as the dependent variable and the log (kilo SLOC), square roots of prior faults, age of files, three variables on file-status, a number of variables on type of program, and variables on released number of the program as the independent ones.

There is much literature on closed-developed software indicating that the time-series pattern of reported faults or maintenance requests is decreasing and leads to zero at the end of the software lifetime, and there is a jump in the MR number soon after a release of software or software upgrade; see for example Burch and Kung (1997), Lee and Jefferson (2005) and Ng, Gable and Chan (2002). In contrast, for open-developed software, reviews of some literature analysing time-series data of the reported faults (Antoniol et al. 2008; Banker et al. 1993; Lucia, Pompella & Stefanucci 2005; Ng, Gable & Chan 2002; Raja, Hale & Hale 2009) indicated that there is no specific prevalent pattern depicted by the time-series plot of maintenance request frequency data, such as continuous decline or incline, either linear or exponential or seasonal, that applies for all types of software systems. Rather, the patterns seem inconsistent across the open-developed application software.

In essence, the more complex the software, the higher the level of MRs; on the other hand, the farther the existing software from its release time, the lower the MR level. This is especially common in closed-developed software.

Unfulfilled and recurrent maintenance requests. It is possible that some MRs cannot be fulfilled because of a change in priority, unsolved faults, and infeasible requests, cancellations or other causes. All of these possibilities, consequently, can be additional sources of MRs because the users are likely to repeat the request. The data from an empirical case study showed that this accounts for 12% of total MRs (Jorgensen 1995). This fact indicates that the level of software availability cannot be perfect, assuming that this unfulfilled request is less critical.

It is also possible that, as in any software development project, software faults go undetected and escape from the testing phase of the maintenance. So, in addition to unfulfilled maintenance requests caused by operational maintenance management problems, any implemented SM (namely the maintenance itself) is very likely to cause other maintenance-related problems, or a ripple effect (Black 2001; Grubb & Takang 2003; Wang, Schach & Heller 2001) because of the interdependence of the software components. The probability of an occurrence of this problem is even higher when a quick-fix or ad-hoc approach is implemented (Lanubile & Visaggio 1995). The maintenance results of this quick-fix approach can be used to represent the quality of the performed maintenance. However, the prediction of the occurrence of the ripple effect is difficult to quantify because of the components' interdependence and because the occurrence is not always immediate (Black 2006). In one case, the effect may come to evidence in a very short time as the input data needed for the maintained software components is supplied and, therefore, the input triggers an unmatched link among the interdependent software components. In other cases, the emergence of the ripple effect may take a very long time until the unmatched link is triggered by an activated link or by inappropriate input.

Many previous studies, such as those by Black (2001, 2006) and Turver and Munro (1994), have tried to understand the causes of the ripple effect and sought models that are able to predict its occurrence and quantity. These efforts are meant to improve SM management (Lanubile & Visaggio 1995; Wang, Schach & Heller 2001). However, to date the effort to develop a prediction model of the ripple effect has not been conclusive, although many researchers have agreed that quick-fixes and shallow analysis of the software have a very high probability of causing other problems (Lanubile & Visaggio 1995).

From this review, it can be concluded that although the level cannot be well predicted, unfulfilled and recurrent MRs also affect the level of total MRs. These types of request are caused by the maintenance management and process itself.

Additionally, in terms of the cause of the ripple effect, Wang et al (2001) argued that the competence level of the maintenance staff plays a significant role. The learning and training factors especially can reduce the ripple effect level. As the competence level increases due to learning and training, the quality of maintenance is also expected to improve.

3.3.5 Classification of software maintenance

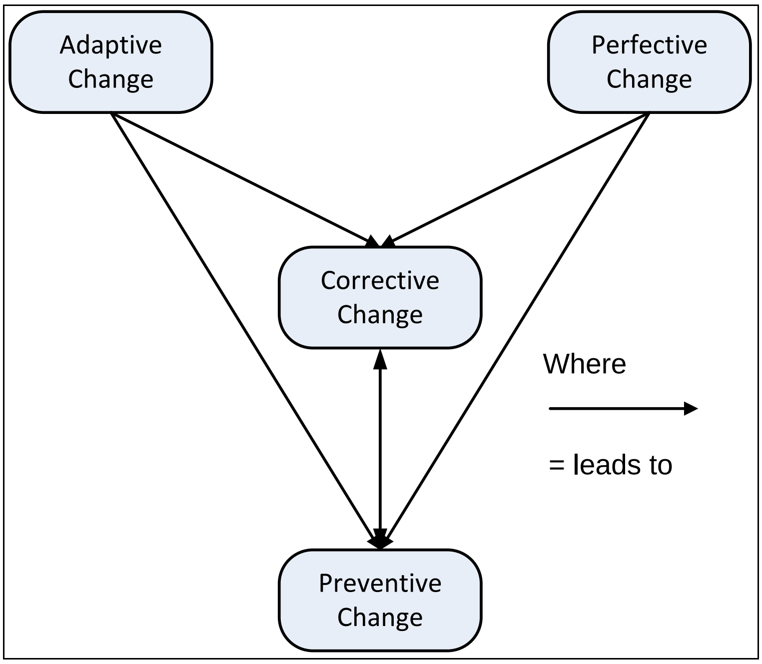
By assuming that all sources of maintenance materialise into MRs, any incoming MRs will be categorised and prioritised, either informally or formally and systematically, for further process (April et al. 2005). The maintenance activities to correspond to the requests can be classified into corrective, adaptive, perfective maintenance (Edwards 1984; Hoffer, George & Valacich 2005; Huff 1990; Perry 1981; Swanson & Beath 1989) and preventive maintenance (Hoffer, George & Valacich 2005; International Organization for Standardization 2006). To simplify the classification and for further terminology usage, this study adopts two classifications according to the International Organization for Standardization (2006): corrective maintenance (CM) and enhancive maintenance (EM), as also suggested by Pigoski (1997**:** 17) that "[n]o matter which definitions you decide to use, it is imperative to discriminate between corrections and enhancements." The first maintenance category consists of corrective and preventive maintenance, while the second comprises adaptive and perfective maintenance. The International Organization for Standardization (2006**:** 25) further defined each category as follows:

- Corrective maintenance refers to modifications necessitated by actual errors in a software product. If the software product does not meet its requirements, corrective maintenance is performed.
- Preventive maintenance refers to the modifications necessitated by detecting potential errors in a software product.
- Adaptive modifications are those modifications necessary to accommodate a changing environment. Adaptive modifications include modifications to implement new system interface requirements, new system requirements, or new hardware requirements.
- Perfective modifications improve the software product's performance or maintainability. A perfective modification might entail providing new functionality improvements for users or reverse engineering to create maintenance documentation that did not exist previously or to modify existing documentation.

These definitions can be further elaborated (Grubb & Takang 2003; Pigoski 1997) as follows. CM is required if defects or faults are found or reported during systems operation. The faults may be discovered by users or by staff of the IS unit. The occurrences of the problem that need CM could take place at any time after system installation is completed. The level of urgency is usually high and actions need to be taken immediately to prevent disruption of business activities due to improper system behaviour or system crash. Preventive maintenance is a necessary change to lessen the possibility of system failure in the future. Adaptive maintenance occurs due to new business requirements as a result of operating the current system for some time and of the changing business environment. Perfective maintenance is conducted to enhance system performance, improve ease of use of the interface or add some features.

The description of SM categories indicates that this classification is necessary because the characteristics of SM problems are different. Response time towards the requests, urgency levels, activities and the maintenance process described above show that they are different. By understanding the characteristics of a maintenance task that have to be implemented, an MR that inherently needs high priority is allowed to be completed according to its urgency or importance level for service delivery (Grubb & Takang 2003). This different nature of MRs necessitates the allocation of different levels of resources.

Despite the necessity to differentiate MRs because of their differing natures, these categorisations do not mean they are mutually exclusive: in practice they are linked together. This intertwining was argued by Grubb and Takang (2003) and examples of possible links are described as follows. Introduced adaptive maintenance might unintentionally contain bugs, which certainly need CM. Likewise, perfective maintenance undertaken to introduce an improved sorting algorithm might necessitate restructuring the program code, which is preventive maintenance. These links are presented in Figure 3.1.



Source: *Grubb & Takang (2003**:** 40)*

Figure 3.1 *Possible dependencies of different categories of maintenance*

In addition, some end-user-support activities also need to be performed as part of the maintenance effort (Abran & Nguyenkim 1991; Burch & Kung 1997; Whitten & Bentley 2007). End-user-support maintenance is defined as activities that support end-users to enable optimisation of the application and provide training to end-users. Although end-user-support maintenance does not directly touch the information systems’ software, it is an important activity to ensure optimum utilisation of the system’s potential capability in delivering organisational services. In some cases, a significant percentage of IS staff effort is dedicated to this activity (Abran & Nguyenkim 1991; Ng, Gable & Chan 2002).

Table 3.1 *Percentage of each type of maintenance reported in previous research*

Source	System	% corrective	% enhancive	% user support
Lee and Jefferson (2005)	Web-based Java application	32	68	
Burch and Kung (1997)	SQL system of a private organisation	48.93	26.3	24.77
Antoniol et al (2008)	Three open-source software	45	-	-
Kemerer and Slaughter (1997)	Commercial merchandise	17	83	-
Ng et al.(2002)	ERP-system of a government agency	31	69	-
Jorgensen (1995)	COBOL-based system of a large company	43	57	-
Abran and Nguyenkim (1993)	Application in financial organisation	20.26	47.31	32.41
Abran and Nguyenkim (1991)		26	39	35

Empirical research on the percentages of each maintenance category indicated that they were significantly different. In general, as shown by Table 3.1, the percentage of corrective MRs is much lower than the percentage of enhancive MRs, which is CM, accounting for less than 50% of all types of the MRs. The percentage of CM provided by Burch and Kung (1997) was exceptionally high. For this case, they explained that a high percentage of the CM within their study was due to the classification of maintenance mostly determined by the vendor, and the vendor will receive payment from the organisation. It is also interesting that user-support service accounts for a significant amount of the percentage, and occupies a considerable amount of the time spent on maintenance activities (Abran & Nguyenkim 1993).

3.3.6 Software maintenance phases and activities

The SM life cycle consists of phases similar to the information system development life-cycle (ISDLC) (Grubb & Takang 2003; Hoffer, George & Valacich 2005; Perry 1981**:** 57-63; Whitten & Bentley 2007). The IEEE Computer Society (1993; 1998**:** 5) defined phases of the SM process as: "a) problem/modification identification and classification; b) analysis; c) design; d) implementation; e) regression/system testing; f) acceptance testing; and g) delivery". Referring to this life-cycle, for any SM activity there should be a flow of SM products. It starts from a maintenance request from users and ends up with completed and delivered maintenance. There are some variations of SM activities and phases description shown in previous research although the variations were insignificant: for example, in April and Abran (2008), Kitchenham et al. (1999), Lucia et al. (2005), Niessink and van Vliet (1998), Singh and Goel (2007) and the International Organization for Standardization (2006).

April and Abran (2008) defined the phases according to the key maintenance process: analysis, modification development and delivery, as depicted in Figure 3.2. The analysis phase is the first and includes activities such as identification of the real problem underlying the requested maintenance including communication with the users, understanding the existing software system, analysing the new requirements and possible impacted software components, and proposing possible solutions. The modification development phase takes place after the analysis phase and includes activities such as designing, developing and introducing the software components, implementing newly introduced software components, modifying the software within a restricted setting, testing the modification within a restricted setting or copy-versions of the existing system. The delivery is the last phase. It includes the implementation of the modification to the existing system, the communication to the requester to ensure that the maintenance satisfies their requirements, and training for the requester about the implemented modification as necessary. Once the maintenance problem resolution delivery satisfies the MR, then the maintenance is considered complete. The maintenance activities are mostly not carried in linear manner but in an iterative way, in the sense that it is always possible to return to a previous phase at any time if necessary; see Figure 3.2.

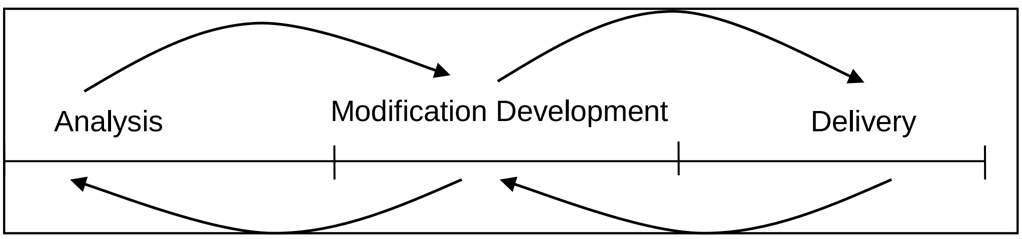
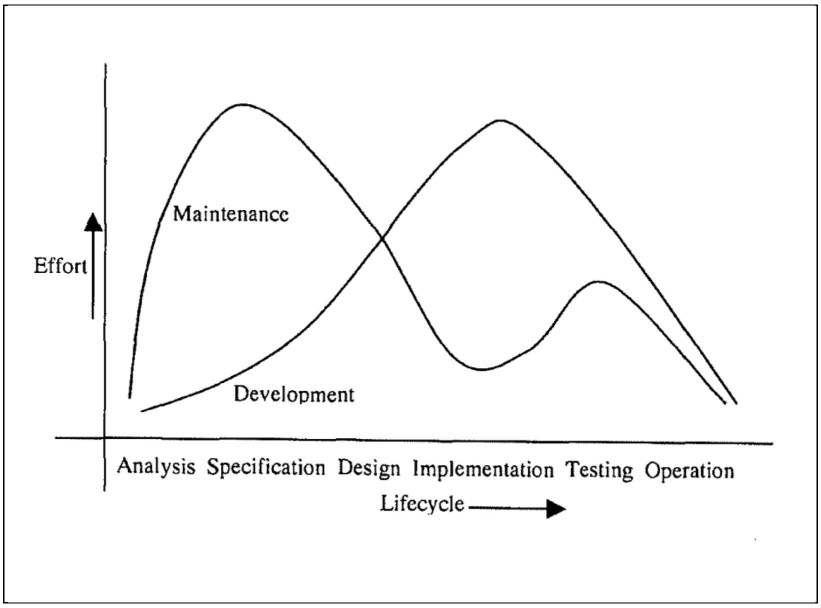


Figure 3.2 *Iterative phases of a maintenance process*

The diagram presented in Figure 3.2 also shows the relationships between phases. The degree of success of the analysis determines the level of modification implementation, which in turn influences the success of maintenance completion and delivery. This simple chain of influences can also operate in the reverse direction. For example, an incomplete analysis process can cause difficulties in implementing modifications, which necessitates a deeper and more revealing analysis; or a carefully undertaken testing with users on an implemented modification can reveal that one or more users' requirements are not satisfied, which triggers further modification or even analysis.

Compared with software development – specifically with respect to the time spent for each phase of the life-cycle – while most development efforts are spent during design, implementation and testing, more than half of the maintenance time and emphasis are devoted to analysis and specification, see Figure 3.3 (Grubb & Takang 2003**:** 74-5). In particular this is the case for enhancement. The analysis is conducted mainly to understand the program and documentation associated with the required maintenance, as well as any possible adverse impacts of that maintenance. In addition to performing SM, an analysis should also be conducted to ensure that foreseen ripple effects can be avoided as a result of adding, deleting or modifying the software (Grubb & Takang 2003**:** 271).

Contrasting the effort and time spent for CM and EM, most CM only takes a very short amount of working time or effort (Jorgensen 1995), implements less consistent methodology and documentation (Taylor & Wood-Harper 1996) and is usually assigned to one staff member (Jorgensen & Sjoberg 2002), while EM requires a longer activity time and there are considerable time differences required to finish each phase (Grubb & Takang 2003). In Jorgensen's (1995) study, time taken for CM is about one-third of that for EM.



Source: *Grubb & Takang (2003: 75)*

Figure 3.3 *The comparison of effort spent by staff over the life cycle between software development and maintenance*

3.3.7 Impact of delivered software maintenance

Any completed and delivered maintenance means the maintained software has experienced changes due addition, deletion and/or modification of the program code (Jorgensen 1995; Jorgensen & Sjoberg 2002). Normally, maintenance causes an increase in complexity of the software, as explained by the law of increasing complexity of Lehman's Law of Software Evolution that, as the system evolves, due to implemented changes over time its complexity increases unless work is done to maintain or reduce it (Lehman et al. 1997: 21). Future maintenance will much depend on these persistent changes; if the implemented changes are undertaken without considering the overall system structure, then upcoming maintenance will be increasingly difficult (Grubb & Takang 2003), and therefore this can reduce the quality of future maintenance delivery. Also, a carelessly implemented change can be a threat to data integrity (Chapin 2009). These eventually negatively influence future software availability.

In addition, successfully completed and delivered maintenance, especially enhancement, might have a significant impact on the organisation. A significantly enhanced system might require the organisation to train users, for example, in how to operate and comply with the enhanced system (Khoo, Robey & Rao 2011; Yokota 2011). Also, a re-alignment between the IS and business process might be necessary, as the enhancement changes the systems (Rashid, Wang & Tan 2010).

3.3.8 Software maintenance management

One of the importance issues for SM management is staff maintenance productivity (Banker, Datar & Kemerer 1991; Calow 1991; Ketler & Turban 1992). While the ripple effect can be associated with the quality dimension, maintenance productivity can be used to indicate the quantity dimension of the SM. Many previous studies have defined and measured productivity in terms of the total labour hours spent by a maintainer to complete an assigned maintenance task (Banker, Datar & Kemerer 1991; Banker, Davis & Slaughter 1998; Narayanan, Balasubramanian & Swaminathan 2009); or the number of completed tasks in a period of time (Benestad, Anda & Arisholm 2010). Pigoski (1997: 256) suggested "the number of problem reports and enhancement requests completed and fielded" as one measure of SM productivity. In other words, the sooner the maintainer completes the maintenance task, the more productive is the maintainer. They considered that the maintainers' or maintenance project team's time spent for undertaking maintenance tasks is a scarce resource that managers should concentrate on.

There have been many studies investigating the factors influencing maintenance productivity. Ketler and Turban (1992), based on reviewing previous SM literature, identified three major factors that potentially affect overall SM performance: technological, organisational and behavioural. They found the technological factor that highly complicates SM to be the software itself. As also shown in previous sections, the MRs caused by this factor are highly unpredictable and less controllable by managers. Ketler and Turban concluded that languages, structure and design of code, size of system, documentation and age of system are software factors determining maintenance success.

Further, Ketler and Turban's identified organisational factors affecting the maintenance are: planning; setting estimates, schedules and priorities; developing programming standards; installing control procedures for changes in program; and the division of labour and organisational structure. In organising the maintenance process, the management has an option either to separate the organisational units of development and maintenance, or to combine them within one single unit that performs both development and maintenance (Yeh & Jeng 2002). In addition, for continuous improvement of the SM process it is necessary for the management to adopt a SM maturity model (April et al. 2005).

Ketler and Turban also concluded that the behavioural and personnel issue is the second significant problem for maintenance success, especially motivational-related factors of the programmers. They identified that motivation, in addition to job characteristics, is dependent on the programmers themselves. Much research has also shown that maintenance personnel motivational factors have been problematic for maintenance success (Grubb & Takang 2003; Pigoski 1997). For example, SM has been associated with being a second-class job compared with software development; and personnel assigned to maintenance tasks are usually within their early careers in IT (Tan & Gable 1998). These become de-motivational factors for them. Additionally, as a subset of a greater IT professional cohort, software maintainers also have to deal with other various motivational, skills and communication factors in order to achieve a high level of maintenance productivity.

A detailed review of how SM staff factors, especially those related to motivation, influence SM performance will be provided in section 3.4.

3.3.9 Summary of software maintenance processes, elements and relationships

Software maintenance is essential in assuring the required level of software availability over time. This maintenance is unavoidable due to the adopted approach and quality of system development, environmental pressure and maintenance itself. A high level of maintenance performance enables an e-government system to cope with the growing and varying demand for service delivery.

The literature review in this section has also indicated a great variety of elements and processes of SM. If they are put together and mapped, feedback relationships and the impact of time can be revealed within and between elements and processes. This will help e-government decision-makers or managers to better manage their e-government software in order to achieve e-government success.

An SM process starts with an MR, which occurs randomly with various degrees of severity. The requested maintenance is carried out based on the type of problem and using the appropriate level of resources. A completed maintenance task is delivered and implemented to the source software. Eventually, the quality and quantity of the maintenance delivery add to the software complexity, which in turn can trigger other MRs. A successful completion in one maintenance phase is influenced by the completion of previous phases. The software development quality and environmental pressures determine MRs, which are then transformed into maintenance tasks. The completion of these tasks depends on the complexity of the software and the type of the tasks, personnel and organisational factors. The complexity of the maintained software eventually affects future maintenance processes.

The quality and quantity of the delivered maintenance can be used as a measure of the SM performance level or software availability over time. One of the key elements affecting this level is the performance of the IS staff (Grubb & Takang 2003; Pigoski 1997).

3.4 Information systems staff

3.4.1 Introduction

This section describes the literature review of IS staff and their role in software development and maintenance. The review is not limited to the literature in e-government but also include a wider context of IS and human resources management.

Despite the existence of complex factors affecting human resource performance, the performance of internal IS staff in maintaining e-government system software will be reviewed through the lenses of their motivation, competence and management. The expectancy theory will be used as the foundation to reveal complex relationships between factors within these areas that lead to the achievement of a particular level of e-government software sustainability through the SM process.

From this review, relationships between various factors affecting the performance of internal IS in e-government SM are expected and can be mapped out.

3.4.2 The role of internal information systems staff for e-government success

Broadly, the e-government human resource factor and its role can be divided into three categories: e-government customers, internal users and IS staff. The first consists mostly of citizens and businesses as the ultimate users (Shareef et al. 2009) who ultimately determine the success of e-government through the benefits they realise from it. They are part of external factor of an e-government system. The second category comprises internal staff, managers and decision-makers of government organisations who operate and utilise the e-government system to deliver government services (Sabherwal, Jeyaraj & Chowa 2006). They can be considered as part of the service delivery management factor. In the last category are internal staff, managers and decision-makers who are employed based on their specific competence in IS. These internal human resources, which are part of the IS management factor, are responsible for designing, developing and maintaining the e-government system (Kamal 2006). They are the ones on whom a sustainable quality e-government system can depend.

Of these three categories of human resource, the eGSF literature review presented in section 3.2.4 has indicated and underlined that particularly internal IS staff play a critical role in e-government success. For example, internal IS staff are required to support the continuous availability of state-level e-government websites in the US (Gil-Garcia 2006); also a sufficient number of skilful and professional IT internal personnel determines the level of e-government adoption (Detlor, Hupfer & Ruhi 2010; Schwester 2009) as well as the ultimate success of an e-government project (Herrera & Gil-Garcia 2011).

In the broader context of the IS management field, the pivotal role of internal IS staff in improving software process and performance has been significantly underlined by much of the existing IS literature, for example, by Boehm (1981), Acuna, Juristo and Moreno (2006) and Madachy (2008). Although the impacts of properly managed internal IS staff on an organisation are less tangible, they are more valuable to the organisation; therefore, employing and managing highly motivated and competent IS staff remains one of the top priorities (Nelson & Todd 2004). For example, as software processes depend heavily on people with regard to the constraints associated with internal IS human resources, managing their variability in capacity and capability is necessary for software development and maintenance (Junchao et al. 2008).

3.4.3 The expectancy theory of human resource motivation

Motivation has been the central focus of researchers and practitioners for a long time in studying human resource performance. There have been many models and theories – such as the stimulus-response theory, equity theory, job characteristics theory, need theory and motivation-hygiene theory – developed to investigate and explain the factors that can motivate human resources to exert their effort in a certain direction in order to achieve a particular degree of performance (Couger & Zawacki 1980; Robbins et al. 2009). These models and theories identified various factors which are considered to affect human resource performance, such as rewards and the allocation of them, significance of tasks, job feedback, need for achievements and self-esteem, responsibility, recognition, advancement and growth, salary, and social interactions. In addition, the various models and theories are also used to understand how these factors determine the degree of motivation (Isaac, Zerbe & Pitt 2001; Rasch & Tosi 1992). Within the IS domain, there have been extensive empirical studies that implemented these motivation-related theories to investigate and explain what and how factors influence IS staff performance (Blanton, Wingreen & Schambach 2000; Couger & Zawacki 1980; Rasch & Tosi 1992). Further, in e-government system development and implementation, a high level of motivation of e-champions plays a determinant role in achieving e-government success (Bhatnagar 2004; Krishna & Walsham 2005).

Among these motivation related theories, the expectancy theory of Vroom (1995), which was originally published in 1964, has been widely implemented and empirically studied by much human resource research; for example, Lee and Jimenez (2010), Liu and Mills (2007) and Memary and Kuan Yew (2009). More specifically, this theory has been investigated and implemented in IT areas including software engineering; for example, by Hall et al. (2009), Rasch and Tosi (1992) and McGrath and More (1998). This expectancy theory strongly indicates the interdependence of its constituting variables.

The classic expectancy theory was built on the assumption that a decision made by a person is subjective and rationally motivated (Vroom 1995). The degree of force that motivates people to act in a particular direction was proposed by Vroom (1995: 21) as:

> a monotonically increasing function of the algebraic sum of the products of the valences of all outcomes and the strength of his expectancies that the act will be followed by the attainment of these outcomes.

On the relationship of the valence of outcomes and their expected consequences, Vroom (1995: 20) proposed:

> The valence of an outcome to a person is a monotonically increasing function of the algebraic sum of the products of the valences of all other outcomes and his conceptions of its instrumentality for attainment of these other outcomes.

On this theory, Isaac, Zerbe and Pitt (2001: 215) identified three components that motivate people to perform a particular act:

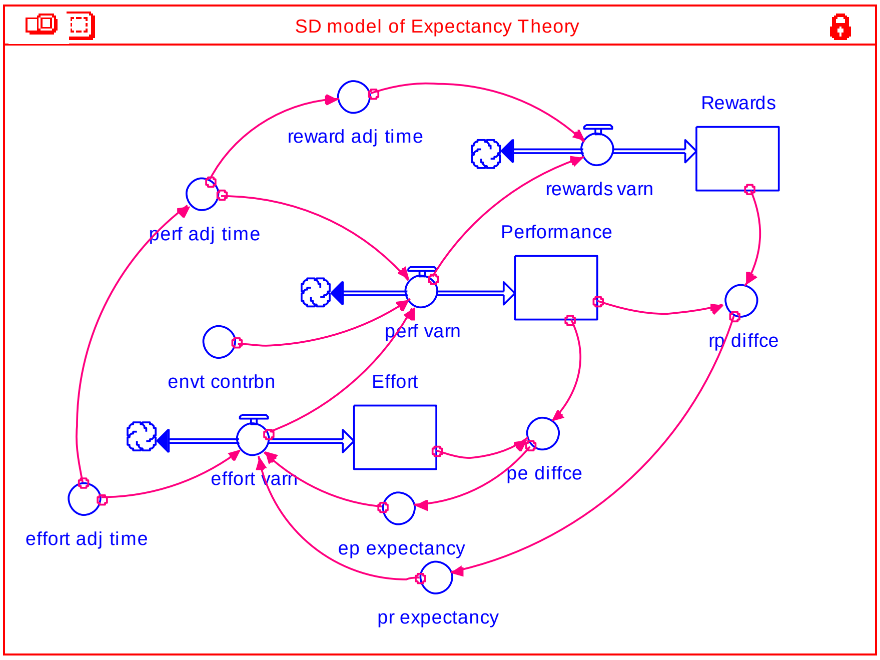
1. The personal expenditure of effort will result in an acceptable level of performance.
2. The performance level achieved will result in a specific outcome for the person.
3. The outcome attained is personally valued.

Further, they described these three points, respectively, as "expectancy" in the sense that effort will result in performance, "instrumentality" which means that a specific outcome depends on a specific level of performance, and "valence" that represents the associated value attached by the person to the received outcomes. A person's level of motivation then can be formulated as:

$M = E \times I \times V$
where E: expectancy; I: Instrumentality; and V: valence (Isaac, Zerbe & Pitt 2001: 215).

In a simple representation, this expectancy, instrumentality and valence link individual effort, performance, organisational rewards and individual goals (Robbins et al. 2009). This link underlines the dynamic feedback relationships of the elements of the expectancy theory as expressed in Vroom's proposition (Howard 1989).

McGrath and More (1998), using the SD modelling method, identified and showed explicitly and visually the existence of feedback relationships between the variables that compose the expectancy theory. Based on the SD premise, they also argued the importance of the impact of time within these relationships in the sense that effort, performance and reward factors may fluctuate over time and the influence of one factor on another is not necessarily instantaneous. The model depicted in Figure 3.4 explains how the variables of the theory and the time factor influence the performance of IS staff in feedback-fashion relationships.



***Source**: McGrath & More (1998)*

***Figure 3.4** The system dynamics model of the expectancy theory*

In this SD model the effort, performance and rewards are represented as stock. The level of each stock is a value within an interval set to 1 to 100. The levels depend on bi-flows of “effort varn”, “perf varn” and “rewards varn”. The model also states that the flow of effort depends on the difference of the effort and the performance levels, as well as the performance and the reward levels. For the performance level, in addition to the impact of the effort, the model also accommodates the contribution of environmental factors to the performance, namely “envt contrbn”. Also, the presentation models the effect of performance towards the rewards. The important things to note from this SD model is that the levels of effort, performance and rewards are time-dependent and the feedback loops are explicitly presented. These time dependencies are modelled as “effort adj time”, “perf adj time” and “reward adj time’.

In essence, according to the expectancy theory, in order to direct the staff to exert their best effort and achieve best performance it is necessary to understand the relationships between effort, performance and reward. Based on the SD method, feedback relationships and the importance of the impact of time must also be taken into account. McGrath and Moore's (1998) SD representation of the expectancy theory shown in Figure 3.4 will be explored and extended to the factors affecting internal IS staff performing e-government SM and the review of previous literature will use this SD representation as a reference.

3.4.4 Relationships between motivation and performance

In SM research, the effort of IS staff was defined as the amount of working time expended to resolve a maintenance task, and therefore IS staff productivity, is indicated by the number of maintenance tasks that can be completed within a specified period of time; see section 3.3.8. In other words, within a specific period of time of exerting their effort, if the staff can complete more maintenance tasks, then they have a higher productivity level than those who can complete only a few. In this section, staff productivity is a function of effort; the role of staff competence towards staff productivity is presented in section 3.4.5. The overall staff productivity in association with overall maintenance tasks indicates SM performance.

As shown by Figure 3.4, there is a feedback relationship between staff effort and SM performance. However, Figure 3.4 does not distinguish staff performance from organisational performance and does not indicate how to measure these performance factors. Staff performance can be measured by staff maintenance productivity and quality, whereas SM organisational performance is indicated by the total performance of the overall maintenance staff, as the organisation is concerned with how to overcome all MRs over time with high-quality delivered maintenance so as to be able to deliver organisational services.

On receiving SM tasks, individual staff will exert their effort. Over time, as the staff carry out SM and observe their productivity, they will judge and compare between the exerted effort and the resulting productivity which, based on their understanding that a certain level of effort is required to achieve a required level of performance, results in either changing or maintaining the current level of effort (Abdel-Hamid & Madnick 1991). Thus, the productivity level can change up and down along with the effort oscillating level. In turn, this will affect overall SM performance in the same way.

Through the instrumentality of the expectancy theory, any SM performance level will result in a certain number of organisational rewards; and staff understand that to obtain the rewards a particular level of SM performance is necessary. In a general sense, the number of rewards provided to the staff could be from zero up to a certain stipulated maximum level. Organisational rewards can be provided in a variety of forms which are not necessarily static and immediate. Normally, organisational rewards are provided to the staff in such a way that the number of rewards reflects the level of SM performance. Mahaney and Lederer (2006) found that intrinsic and extrinsic types of rewards significantly influence the developers of successful IS projects. In their research the intrinsic rewards consist of pride, a sense of contribution to organisation and public praise, while the extrinsic ones comprise favourable annual performance appraisals, project completion celebration, job security, technical training, flexible work schedules, job promotion, financial bonuses, etc. Using the employees of research and development units as the target population, Chen, Ford and Farris (1999) found that the intrinsic rewards contributed the most to the organisation, followed by group rewards and fixed salary increases. Well-designed organisational rewards can include intrinsic, individual-based monetary, symbolic, professional recognition, promotion, group-based, organisational-based and items that failed on an appreciable level (Chen, Ford & Farris 1999). As one way to motivate IS staff in SM (Grubb & Takang 2003, Ch. 10), the rewards must be associated objectively with the IS staff performance (Cappelli 2001). Grubb and Takang (2003) further emphasised that these rewards must be thoughtfully designed such that the maintainers feel valued and rewarded, as SM is generally perceived as lower in status than software development.

For any level of SM performance the staff will expect certain rewards, and for any number of rewards the staff judge objectively and/or personally the value of those rewards. This is, as Vroom stated, that IS staff exert their effort to a certain degree because they expect to be rewarded and the that reward is of comparative value with their performance. Examples of rewards expected by staff include: receiving opportunities to build a professional career, introducing their own ideas, working with other competent staff, attaining personal development, being challenged to change application systems, obtaining annual rewards etc. (Laumer 2009). Staff might expect to receive rewards because of their achieved performance based on their perceived significant skills and tasks (Hirschfeld, Schmitt & Bedeian 2002). Even by being part of an IS unit and achieving a particular degree of performance, IS staff have their own inherent expectations. The staff perception of the received rewards based on their expectations and judgement can be different from the actual value of the organisational rewards. The difference between the actual organisational rewards and their perceived value will determine the degree of staff effort. This further emphasises the important role of organisational rewards and the significance of understanding IS staff expectation and perception of the rewards.

In sum, as shown in Figure 3.4 the degree of IS staff motivation, which is reflected by staff effort over time, is affected by the difference between SM performance level based on normal effort and the required SM performance level, and by the discrepancy between the perceived value of the organisational rewards and their actual value. The impact of this difference on effort could also take time.

In addition to the attained performance level and the organisational rewards, many SM studies have indicated that staff productivity is positively influenced by software complexity (Abran & Nguyenkim 1993; Ahn et al. 2003; Banker et al. 1993; Banker, Davis & Slaughter 1998; Jorgensen 1995; Schneberger 1995). Software complexity can be a result of software development as well as of software's dynamic nature. It must be noted that, as described in section 3.3.7, this dynamic complexity is due to the implemented maintenance over time. For example, maintained complex software of a clinical IS increased about 167% in software size during five years of system operation (Blum 1995).

3.4.5 Relationship between competence and productivity

Previous literature on eGSFs has shown the importance of ICT-related competence of e-government staff in achieving the success of e-government, for example, Ojo et al. (2007) and those presented in section 3.2.4. Competence is associated with "the set of skills and knowledge that an individual needs in order effectively to perform a specific job" (Baker et al. 1997: 266). In addition to motivational factors, much research has shown that the competence level of IS staff significantly determined the software productivity (Madachy 2008). This was also shown by Mishra and Misra (2007) for software engineering products. Especially in SM, the importance of maintainer competence levels in influencing SM performance was also underlined by Collofello and Orn (1989) and Kajko-Mattsson, Forssander and Olsson (2001).

To define the maintenance competence level of IS staff, this study adopts Abdel-Hamid and Madnick's (1991) productivity variation formulation for a software development project, which was derived from a psychology study. In this formulation, the actual productivity equals to the potential productivity minus losses. For this study, the potential productivity of the staff is adopted to represent the staff competence level. It is a function of staff formal education levels and experience relevant to SM, by assuming that the higher the education level and the more experience, the more SM tasks can be completed successfully. Accordingly, if the staff are fully motivated and the environment perfectly supports the staff effort or there is no productivity loss, then the actual productivity equals the staff competence level.

With respect to the experience level, much research in SM has indicated that experienced staff were more productive than inexperienced ones in the sense that as experience increased over time, the productivity level also increased (Banker, Datar & Kemerer 1987; Banker et al. 2002; Calow 1991; Narayanan, Balasubramanian & Swaminathan 2009; Pigoski 1997; Sneed 1997). However, with regard to the impact of education level and experience on SM productivity, when productivity was defined as the sum of SLOC inserted, updated and deleted divided by effort, Jorgensen's (1995) finding did not support the claim that education and experience factors improve productivity. As Jorgensen acknowledged that this measure is weak, therefore the researcher suggested using these research results cautiously, as he found that more experienced maintainers were assigned to more difficult maintenance tasks. It can be noted that the experience level is dynamic, because experience will rise as knowledge or skill acquired from seeing and doing things accumulates over time. Therefore over time, as the number of completed maintenance tasks increases, the staff experience will also increase.

Additionally, appropriate training attended, in terms of quality and quantity, during maintenance tenure can also increase the level of staff competence. Berman et al. (2006: 219) defined training as "the effort to increase the knowledge, skills, and abilitiies (KSAs) of employees and managers so that they can better do their jobs". There are several reasons for the need of training:

- the staff should be able to cope with the persistent degradation of their own competence level due to the ever-advancing nature of IT (Joseph, Kuan-Koh & Hao-Foo 2010; Lee & Mirchandani 2009)
- the staff is new to the organisation or to the normally utilised tools within the organisation (Jeet et al. 2010)
- it is necessary to improve the quantity and quality of completed maintenance (Taylor, Moynihan & Laws 1998) or
- there is a transfer of developed software from the developer to the maintainer (Pigoski 1997).

Taylor, Moynihan and Laws (1998) found that training for SM can be formal, such as software testing training commercially provided by external companies, and informal, such as knowledge transfers from more experienced internal IS staff through supervision, or presentation and discussion. Through improving competence levels, appropriate training is an important factor that eventually influences SM performance (Collofello & Orn 1989; Kajko-Mattsson, Forssander & Olsson 2001; Mishra & Misra 2007; Pigoski 1997).

In contrast with the experience and the training factors which can raise the competence level, IT advancement causes the staff competence level to decrease so that over time it will be eroded (Hafeez & Abdelmeguid 2003). Lee and Mirchandani (2009) identified five areas of IT that can cause this decrease and that need anticipation from IS professionals generally in order to maintain their competence level: enhancement of web applications; wireless (mobile) technology; security; open source programming; and data warehousing to support analysis. For this reason, Lee and Mirchandani urged the need for continuous upgrades of the competence level. Although this threat to the competence level is real, there seem to be no specific figures from the literature as a general indication of the rate of competence obsolescence as a result of IT advancement.

For an organisation, it is quite obvious that attrition of experienced staff affects the overall competence level. As inexperienced and experienced staff have different levels of performance (Banker, Datar & Kemerer 1987), the fractions of inexperienced and experienced staff within an e-government unit directly determine the overal competence level. An increase in the number of inexperienced staff due to new recruitment and experienced staff turnover causes the overal level to decrease. In most cases, new staff recruited for SM are inexperienced, therefore they have to be trained (Pigoski 1997).

In sum, this review indicates the existence of the feedback relationships between IS staff productivity level, competence level and other factors. This review suggests that competence level influences IS staff productivity level, which in turn affects staff experience level and the latter factor influences staff competence level. In addition, the competence level is negatively affected by IT advancement. The decrease in competence level, proportion of experienced and less experienced staff and quality of completed maintenance trigger the need for training and the implemented training influences the competence level. The impacts of relationships might change over time.

3.4.6 The influence of environmental factors on productivity

Staff productivity is not influenced only by motivation and system complexity, but also by the environment surrounding the staff workplace (McGrath & More 1998). Among the elements of the environment are other maintainers, software developers and system users. To perform their maintenance jobs, staff need to communicate with other maintainers and system users (Jorgensen 1995). Maintenance team communication and collaboration, communication between the IS staff and maintenance managers, and the working environment (Grubb & Takang 2003), team size and communication with users (Narayanan, Balasubramanian & Swaminathan 2009), deadline pressure (Banker, Datar & Kemerer 1987) and system users (Calow 1991) are other environmental factors affecting the performance level of staff.

In addition, in some cases inexperienced maintainers have to communicate with their supervisors (Jorgensen & Sjoberg 2002). In other cases, maintainers should speak to software developers during software transfer, as well as during software operation, in order to collect information about the software being maintained (Pigoski 1997; Seaman 2002), although in some cases the developers are less valuable sources of information compared with others because developers' knowledge of the maintained software is no longer current (Seaman 2002). Internally, within their workplace the SM staff have to communicate with other IS staff in order get help through discussion, to obtain understanding of the software being maintained from previous maintainers, or to discuss the link of the software module being maintained with other parts of the software which are maintained by other staff (Elahi, Yu & Annosi 2009; Seaman 2002).

The SM staff quite often also have to communicate with the system users who lodge MRs for either correction or enhancement (Bendifallah 1987; Pigoski 1997). The purpose of the communication can be to improve specification of the users' requirements; to consult on a new design for EM; or to test and give training to the users for the delivered maintenance. The degree of communication smoothness between the staff and system users determines staff productivity (Calow 1991).

3.4.7 Information systems staff management

Generally, IS staff management has to deal with staff availability through recruitment of new staff, allocation and assignment of the staff to a particular task, staff development and staff turnover (Nelson & Todd 2004). Overall, SM performance to some extent is obviously determined by the number of staff available to undertake SM tasks. It is management's role to make sure that there is a sufficient level of staff availability. However, staff availability is dynamic. Viewing the level of availability might be easier if taken from the perspective of staff unavailability. Staff can be unavailable because they have not been recruited; are assigned to other organisational jobs; have been moved to other units or promoted to a managerial level; pursued other jobs in other organisations; or are absent from work for various reasons.

An organisation may recruit new IS staff for maintenance with or without degree backgrounds. For example, a government unit in Indonesia recruited IS staff with high-school and higher education backgrounds (PUSDATIN 2006). Newly recruited staff generally can be assumed to have a lower productivity level than current staff, irrespective of their educational level, because at least they have not been familiar with the current operating software being maintained, or with the internal environment (Abdel-Hamid & Madnick 1991; Madachy 2008 Ch. 4). Through self-learning from experience and training during their SM tenure, IS staff can be categorised as experienced staff. The rate of IS staff being recruited depends on the maintenance needs (Hafeez & Abdelmeguid 2003).

During their tenure as software maintainers, staff may be transferred to other organisational units, promoted to managerial positions, or even exit to pursue careers in other organisations or retire. In addition to this, the management may also have to allocate some experienced staff to provide on-the-job training (Abdel-Hamid & Madnick 1991; Banker, Datar & Kemerer 1987; Jorgensen & Sjoberg 2002). This training will reduce the availability of experienced staff numbers to perform maintenance tasks assigned to them.

IS staff turnover has been a chronic problem in IS organisation practices. Much empirical IS research has devoted attention to this issue (Igbaria & Greenhaus 1992; Kim 2005; Moore 2000) and how to retain competent staff (Igbaria & Shayo 2004; Kim 2005). Previous research has found many factors that significantly relate to IS staff turnover or turnover intention, such as low salary and limited opportunity for promotion (Igbaria & Greenhaus 1992), salary and additional benefits (Palmer et al. 1998), and structured rewards, tasks variation, career structure and recognition (Grubb & Takang 2003), as well as the need for change in career direction (Joseph, Kuan-Koh & Hao-Foo 2010). Whatever its cause, it is certain that this turnover negatively and signifcantly influences overall performance and e-government success (Sang, Lee & Lee 2009). Staff turnover affects SM performance in two ways: it reduces the proportion of highly competent staff if those who exit are indeed of a high level of competence; and reduces the total numbers of staff availabe.

Absenteeism has also attracted much research attention (Junor, O'Brien & O'Donnell 2009; Marks et al. 1986; Morgan & Herman 1976), and it commonly exists in various sectors. For example, among call-centre employees in the Netherlands in 1999, the rate of absence was 7.9% (Schalk & Van Rijckevorsel 2007), while among public servants in several developing countries, Chaudhury et al. (2006) found that the absence level was 19% for teachers and 35% for health workers. Absenteeism affects staff availability to perform their jobs, which in turn influences staff and organisation productivity (Junor, O'Brien & O'Donnell 2009).

Rewards or perceived rewards seem to correlate with absenteeism. By acknowledging the existence of differences in the characteristics of IS jobs and the profession, the results of the research from other fields are worth noting as there was relatively little knowledge about absenteeism from previous IS research. For example, Griep et al. (2010) have empirically shown, using a nurse sample in Brazil, that effort—reward imbalance is a good predictor of absenteeism. A significant absence level becomes more prevalent among skilled government employees who can earn more payment from doing outside jobs or moonlighting while still keeping their main employment and, of course, its wage (Dabalen & Wane 2008). Even clerical public-sector workers, who perceived that their jobs required higher skill or task significance but received limited rewards, tended to have a higher level of absence (Hirschfeld, Schmitt & Bedeian 2002).

3.4.8 Summary of information systems staff success factors relationships

Much literature in IS management has shown theoretically and empirically the critical role of IS staff performance for the success of an information system.

In accordance with the expectancy theory, IS staff motivation, which is reflected as staff effort, has dynamic feedback relationships with staff performance and organisational rewards. Staff performance is represented by staff maintenance productivity and delivered maintenance quality.

Putting these expectancy theory factors together with other performance-related factors, further dynamic feedback relationships leading to staff performance can be mapped out. The other factors include staff competence level, IT advancement, training, experience level and communication with the SM environment. In addition, overall SM performance is obviously influenced by total staff available for SM, which in turn is affected by total staff, absence levels, staff turnover and promotion, recruitment policy and staff allocation.

3.5 Summary

This chapter has presented the review of literature in association with eGSF relationships. It has presented the wide dimension of eGSFs, in topics and constructs, and the complex relationships among them. This presentation did not specifically identify what set of factors is believed to lead to e-government success; rather, it indicated how this broad range of eGSFs can be organised, while at the same time emphasising the fact that the eGSFs relate to each other. In other words, it is not only the eGSFs but also the relationships among them that should be considered by decision-makers and managers in achieving e-government success. Given this broad dimension of eGSFs and their relationships, a further review to reveal how the elementary level of success factors relate to each other while influencing e-government success would be an unfeasible undertaking.

Given the lack of attention to the operation and maintenance stage of the e-government life-cycle and its important role in achieving e-government success, this study focuses its investigation on the relationships between eGSFs in e-government SM and internal IS staff management subsystems.

The review of SM literature has revealed the elements, processes and relationships within and between these two. Any SM activities are initiated by MRs. An MR can be triggered by software development quality, environmental factors and SM itself. The MRs occur randomly and their frequency is decreasing over time. The MRs are then classified as either CM or EM tasks, each of which requires a different amount of time to complete. A CM is required to correct a software fault, while an EM is needed to enhance the software in accordance with the systems users' or organisation's needs. In general, there are three phases of the SM process. For any completed and delivered maintenance, it may contain faults which can trigger a new MR, thus showing the existence of feedback relationships within SM.

Overall SM performance can be measured by the fraction of MRs and completed and delivered maintenance, and the delivered maintenance that contains faults. A high level of SM performance ensures the dynamic software availability level to be high. This is in the sense that unavailability of the software, due to software faults and unavailable software features, can adversely affect system sustainability, leading to e-government failure.

The review has also shown that one of the key factors determining SM performance is the performance of internal IS staff. This performance in SM depends on staff motivation in exerting their effort which, according to expectancy theory, is determined by staff perception of their performance and the received rewards. The effort, performance and rewards of the expectancy theory relate to each other, forming a closed loop. The level of staff effort influences their performance, which in turn affects SM performance. The difference between the required and realised levels of SM performance carries influence back to the effort. SM performance will result in both provision of organisational rewards and staff expectation of rewards, the difference between which affects the staff effort level.

Other significant success factors influencing IS staff performance along with its relationships have also been reviewed. The review has also shown the level of competence of IS staff significantly influences IS staff performance in SM. Average competence level influences IS staff productivity, which in turn affects their experience level and this experience level will influence the average competence level. The average competence level is affected by the proportions of experienced and less experienced staff and is negatively affected by IT advancement. This, along with the quality of delivered maintenance, causes a need for staff training and the result of staff training influences the average competence level.

In addition to these success factors, software complexity and communication with other IS staff and system users have been shown also to influence IS staff maintenance performance. IS staff management-related factors, including recruitment policy, staff turnover and staff absence level, have been indicated by previous literature to influence the level of staff availability, which in turn affects SM performance. Staff turnover and absenteeism were indicated as being affected by organisational rewards.

However, these relationships are scattered and were studied separately. The literature review of the two subsystems in this chapter has tried to put these relationships altogether and shown the existence of feedback relationships between the elementary success factors. The level of the success factors as a result of the relationships may change over time.

This summary of success factors and their relationships obtained from existing literature can be tabulated as follows.

Table 3.2 *Tabulated summary of success factors and relationships*

Success factors	Relationships	Success factors
SM performance	⟶	Software availability level
SM performance	⟶	SM requests
Environmental pressure for enhancements	⟶	SM requests
Software quality resulting from a software development	⟶	SM requests
SM requests	⟶	IS staff effort
IS staff performance	⟶	SM performance
IS staff effort	⟶	IS staff performance
SM performance	⟶	IS staff effort
SM performance	⟶	Rewards for staff
Rewards for staff	⟶	IS staff effort
IS staff competence	⟶	IS staff performance
IS staff competence	⟶	IS staff productivity
IS staff productivity	⟶	IS staff experience level
IS staff experience level	⟶	IS staff competence
Ever-advancing IT	⟶	IS staff competence
IS staff competence	⟶	Training and learning
Training and learning	⟶	IS staff competence
Software complexity	⟶	IS staff performance
Communication level	⟶	IS staff performance
IS staff availability	⟶	SM performance
Rewards for staff	⟶	IS staff availability

The eGSFs and the existence of feedback relationships among these factors, as well as the impact of time, lend themselves to the use of the systems approach, especially the SD method. Therefore Chapter 4, which discusses the research methodology, will describe the system approach as an adopted point of view towards eGSFs relationships. Chapter 4 also presents and discusses the modelling methods implemented for eGSFs relationships based on this systemic view: the SD method. Chapter 5 will present the initial model of eGSFs relationships based on the relationships identified in this chapter (Chapter 3) and the modelling method presented in Chapter 4.

Chapter 4 Methodology

4.1 Introduction

In the previous chapter, a broad range of eGSFs and their relationships were revealed that are viewable as a system. Further detailed investigations on two relatively low-level success factors subsets – software maintenance and information system staff – indicate the existence of complex feedback relationships among the success factors leading to e-government success.

This chapter presents the research methodology approach, methods and techniques adopted and implemented in this study in order to fulfil the aims of this research, that is, to develop a formal model of eGSF relationships for developing countries. Considering the nature of the eGSFs and their relationships revealed in Chapter 3, a system thinking approach is adopted. The system decomposition and SD methods are chosen to implement this approach along with their modelling techniques. A case study method will be used to collect the appropriate data.

The chapter begins with describing the research approach adopted in investigating and developing the model of eGSFs and their relationships. This includes the rationale for selecting the approach and a description of the modelling methods. This is followed by a description of the research design, which will be further explored in a step-by-step fashion. Next follows the detailed description of the modelling methods: the use of the methods to develop the model and how the model will be described. Finally, the testing procedure to ensure the validity of the developed model is presented. In this section, a case study procedure to collect data from an e-government system is also described.

4.2 Research approach

4.2.1 Systems thinking

A system is defined as a complex whole formed by interrelated elements for a particular common purpose (Myers & Kaposi 2004; von Bertalanffy 1950; Wand & Weber 1990). Furthermore, this literature explains that a system is not a sum of its elements; rather, the interrelationships among its elements form the wholeness of the system, in which the behaviours or properties of the elements together with their relationships define the system. In addition, an element or a set of elements of a system can be a lever to other elements in the sense that the role of a lever element is much higher than that of the other elements, such that any small changes in the element cause significant changes to the whole system. Any subset elements of a system are systems in themselves, while at the same time a system is a subset of a wider context system. A system is always represented as a model, which is a purposefully simplified representation that contains only the essential and relevant characteristics of the system and states explicitly the boundary between the system and its environment (Myers & Kaposi 2004). The system model can be visual and diagrammatic (Maani & Cavana 2007) which, as a representation of a referent, is never and does not need to be complete (Pidd 2004).

By referring to the description of the system, system thinking means viewing things as a whole which is formed by the inter-relationships of its elements with respect to a particular common purpose. In Senge's (2006: 68) words, system thinking is "a framework for seeing interrelationships rather than things, for seeing pattern of change rather than static 'snapshots'". By associating specifically with time, Maani and Cavana (2007: 7) define system thinking as "a scientific field of knowledge for understanding change and complexity through the study of dynamic cause and effect over time". Further, Maani and Cavana described the system thinking paradigm as consisting of forest thinking, dynamic thinking, operational thinking and closed-loop thinking. The first paradigm means seeing things as a whole and the way the elements relate each other; the second one considers that things always change over time; the third is associated with comprehending the way things "physically" operate and are related each other; and the last paradigm acknowledges the non-linearity of cause and effect and often the fact that the effects can influence the cause.

The eGSF literature review has shown that eGSFs are myriad, which indicates that the success factors interact with each other and suggests that in a particular situation there is a small set of success factors that have more influence in achieving e-government success. The review has also shown that relationships are not static but fluctuate over time. These characterise the complex and dynamic nature of eGSFs and their relationships.

Thus, considering both the definition of a system and the eGSFs literature review, it is quite appropriate to regard eGSFs and their relationships as a system. This consideration can be justified by the fact that eGSFs consist of elements, relationships and a purpose. The elements are the success factors, the relationships are the relationships between the success factors as identified from the literature review and the common purpose is to achieve e-government success. By adopting this system thinking for eGSFs and their relationships, this research argues that in order to achieve e-government success, the e-government decision-makers, developers or managers should consider not only the success factors but also, more importantly, the relationships among the success factors over time.

Considering the characteristics of eGSFs, the adoption of system thinking to investigate eGSFs has been recommended and claimed appropriate by Titah and Barki (2008). In a more general context, Scholl (2007) and Beynon-Davies (2007) suggested the necessity of system thinking in order to obtain full understanding of an e-government.

However, this system thinking needs implementation methods. Literature has indicated that a variety of different approaches have been developed to implement systems thinking (Jackson 2003) and have been utilised within different areas. Considering the variations in the implementation methods of system thinking, this research implements two main modelling methods as its approaches, which is different from the approach adopted by Titah and Barki (2008):

- IS process decomposition (Whitten & Bentley 2007); and
- System Dynamics modelling (Forrester 1961; Maani & Cavana 2007).

The first method is a system thinking approach commonly implemented during the analysis and design phases of an information system development. This method considers the process of an information system as a whole, which consists of interdependent elements and processes. The second method, which is also one branch of system thinking (Hjorth & Bagheri 2006), considers the importance of the time and causal feedback interaction of various elements of the system in order to achieve organisational success (Forrester 1961**:** 13; Sterman 2000).

4.2.2 System decomposition

Within IS process modelling, system decomposition, which is implemented to analyse and design an IS, is intended to organise the whole system into a small and manageable number of subsystems (Whitten & Bentley 2007). In dealing with a complex system, decomposition has been one of the most important tools and has been successfully practised to tackle a large-scale and complex domain of interests (Courtois 1985; Paulson & Wand 1992: 174). How many and what subsystems will be the result of a system decomposition is usually determined by considering a particular criterion or set of criteria such that the system has more cohesive elements within each subsystem but more loosely coupled among different subsystems (Paulson & Wand 1992).

The complex nature of eGSFs and their relationships, where the factors span a wide range of domains, indicates that it is necessary to decompose the eGSFs system first before tackling any problem. By implementing this approach, it is expected that the eGSFs system can be organised into a small and manageable number of loosely coupled eGSFs subsystems. The resulting presentation will serve as the holistic view of the eGSFs system and its subsystems. This holistic view will assist the modeller to focus on the particular scope of a problem and, on the other hand, to put the selected areas of relationships modelling into a broader context of success factors emphasising the fact that the selected areas are sub-subsystems of the eGSFs system. This might retain awareness that many other success factors can also influence e-government success; and this indicates what the other success factors are. Additionally, lack of focus due to modelling of all eGSFs and their relationships can results in a very complicated model, which undermines its usefulness.

4.2.3 System dynamics

Once the decomposition representation of the eGSFs system has been established and a holistic view of the eGSFs has been acquired, then it is necessary to model the relationships of the eGSFs and use the model to learn how to achieve success by understanding its complex relationships. The SD modelling method is implemented to formally model the relationships of the eGSFs in influencing e-government success for selected eGSF subsystems.

Literature has not defined the SD method in a single formulation. Coyle (1996: 10) defined SD as:

> System dynamics deals with the time-dependent behaviour of managed systems with the aim of describing the system and understanding, through qualitative and quantitative models, how information feedback governs its behaviour, and designing robust information feedback structures and control policies through simulation and optimization.

The method was originally developed by Forrester (1961) and popularised more recently by Senge (2006). Forrester (1961) pointed out that quite often, in a real world phenomenon, a change of the level of a variable is affected by other variables which are not in a closed distance. A chain of causal relationships of variables and time delays masks the influence. Furthermore, he argued that any system is governed by an information-feedback mechanism which forms the basic foundation of the SD method. Forrester (1961: 14) described the existence of an information-feedback system as "the environment leads to a decision that results in action which affects the environment and thereby influences future decision". In SD, the relationship between the elements of the system can be much more important than the elements themselves (Forrester 1961: 14).

The techniques and tools of this method were chosen as the means to specify and implement the detailed model of the eGSFs system. The method is a particularly appropriate modelling approach where time and feedback loops are important, and where considerable complexity, ambiguity and uncertainty exist (Vennix 1996). Also, it is especially capable of modelling the complex non-linear relationships that exist among the success factors (Maani & Cavana 2007; Sterman 2000).

For these reasons, the SD method is chosen as appropriate for modelling the complex relationships between eGSFs of selected subsystems.

Coyle's definition of the SD indicates that there are two types of model representations for this method:

- a qualitative model (Coyle 2000; Maani & Cavana 2009); and
- a quantitative model implemented through a computer simulation (Maani & Cavana 2007; Morecroft 2007).

To support these model developments, various types of data are required: “numerical, written and mental data” (Forrester 1980 in Sterman 2000: 853).

The qualitative representation of the SD is the SD model that does not include any quantitative data in the model representation and does not involve a computer simulation (Wolstenholme 1999). This qualitative representation is then visualised in a causal loop diagram (CLD), which is a means of creating a high-level conceptualisation of feedback loop relationships within the system structure (Wolstenholme 1999). The representation depicts how one variable causally relates to another variable, which eventually creates a feedback loop. Tracing causal loops within a causal loop diagram can provide an explanation of how and why a change in an important variable, which is the subject of interest, takes place over time, and can provide an understanding of how the effect variable in turn influences the cause variable. Therefore, the CLD can generate insight into the dynamic structure of the system and provide the dynamic hypothesis of the investigated problem (Madachy 2008; McLucas 2005; Sterman 2000).

Although in some SD modelling processes modellers jump directly to the simulation model, such as that undertaken by Georgantzas (2003) which is a purely quantitative one, this study developed a qualitative representation of SD prior to the simulation modelling, as was carried out by Akkermans and Oorschot (2005). Consequently, the simulation model will be developed once the CLD is set up. It must also be noted that, as Coyle (2000) argued, for a particular purpose SD modelling may involve only qualitative model development.

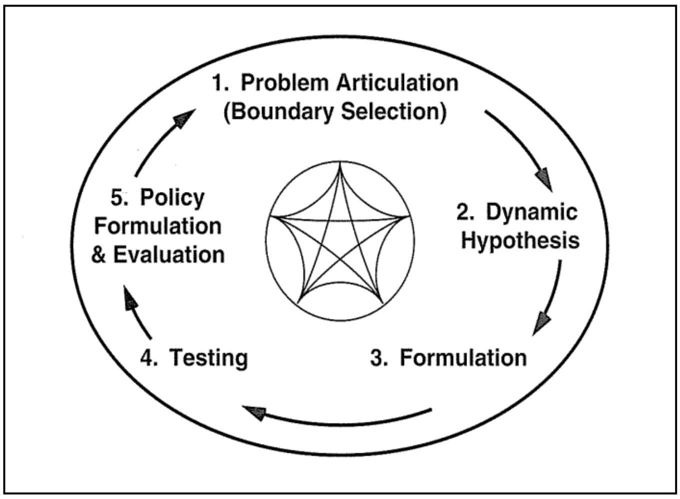
The quantitative model of SD uses stock and flow symbols as the main elements of the model, as well as other diagrammatic representations to capture the interconnected elements of the systems. It contains mathematical equations, initial and parameter values, and linear or non-linear graphical representations, which are embedded within the model. The implementation of this quantitative modelling method needs quantitative data, quantification of qualitative factors and mathematical equations along with its parameters and initial conditions to run the simulation. This method further formalises the eGSFs system explicitly to solve vague and ambiguous concepts and resolve unnoticed contradictions. Thus, a quantitative model of the SD is a computer simulation model which formally and explicitly uncovers the dynamic structure of a system. This computer simulation model plays a vital role in SD modelling. Even during the initial development as a discipline, SD means a quantitative approach for complex problems implemented as computer simulation models (Forrester 1961).

With regard to the CLD, it can involve a very large number of elements such that it becomes very difficult for modellers and model users to comprehend. The ability of people to make correct conclusions over a closed chain of causal relationships degrades sharply as the number of elements within the system increases (Sterman 2000). As the CLD complexity increases, the dynamic consequences of the system become unclear. Therefore, the SD models are eventually implemented using a computer simulation modelling method. For this, special-purpose software is used to implement the quantitative modelling of the SD.

In addition to the system explication and helping people to infer correctly, the simulation model serves a number of purposes to system modellers (Morecroft 2007: 85). Maani and Cavana (2007: 59-60) identified the purposes of the computer simulation model as follows:

- More information can be contained in a computer simulation model than in a conceptual model [CLD]
- Causal relationships and assumptions can be formulated clearly and unambiguously
- Once the model has been validated, it can be used reliably to simulate alternative model experiments without manual errors
- Assumptions can be altered easily for different experiments
- Experiments can be performed readily with different structures and policies
- Sensitive parameters or assumptions can be located quickly through repeated simulations
- Uncertainties and errors can be incorporated into the model explicitly
- Graphical and tabular output can be communicated easily to management or model users
- The model provides a laboratory tool for learning about the behaviour of the real world.

The SD model development consists of a number of steps. Luna-Reyes and Andersen (2003) identified the existence of variation in steps or activities in terms of the number of the steps and the name of each step. Although in terms of the overall understanding of the SD modelling process they are the same, the number varies from three to seven. Sterman (2000) described the SD modelling process as shown in Figure 4.1. This figure indicates that in addition to sequential steps of the model development, the processes are iterative in nature, as also underlined by Coyle (1996). In addition to the main cycle that links one step to the next one, the web-like diagram drawn inside the circle in Figure 4.1 represents that it is possible from any step for a modeller go back to any previous steps.



***Source**: Sterman (2000**:** 87)*

***Figure 4.1** General SD modelling process*

Sterman (2000**:** 86) further explained the general steps in Figure 4.1 as follows:

1. In the first step, the problem being addressed by the model is articulated, key variables are identified, the time horizon is determined, and reference modes or historical and possible future behaviour of key concepts are defined.
2. Dynamic hypothesis is formulated. In this step, current theories used to explain the problem are defined focusing on the endogenous variables and their feedback relationships. A causal loop diagram can be used to develop and represent this hypothesis.
3. A simulation model is formulated. In this step, the structure of the model is specified and parameters, relationships, and initial conditions are estimated.
4. The simulation model formulated in step 3 is then tested. This is the model validation step. There are various testing procedures that should be followed in order to validate the model – that is, to ensure a confidence in the model usage and usefulness. It is imperative that the test is performed from the very early stages of the model construction.

5. At the fifth step, once a sufficient level of confidence about the model has been established, the model can be used to design and analyse policies. What-if analyses or scenarios can be specified and their effects can be evaluated by using the model.

4.3 Research design

4.3.1 Key research questions

Based on the research gaps identified in Chapter 2, the main and broad research question can be stated as:

- What are the relationships between e-government success factors in influencing e-government success in developing countries?

Referring to the review of eGSFs and their relationships presented in Chapter 3, responding to this main research question will necessitate a complicated model of relationships that does not assist model users to understand how the eGSFs relate to each other and influence e-government success. Consequently, the detailed relationships pertaining to all identified success factors will not be modelled. However, as has been argued in the previous section that acknowledged eGSFs are a system, the myriad eGSFs need to be organised into a manageable number of subsystems to provide a holistic view of the success factors, while at the same time facilitating the decision-makers or managers to focus on a particular subsystem(s).

- What is a holistic view of the eGSFs system? What are the subsystems of this holistic representation?

Following the scope of the research, the review of the eGSFs in Chapter 3 and the context of the eGSFs system, the specific and detailed research questions are presented as:

- How do dynamic feedback relationships between SM and IS staff motivation-related success factors influence e-government success in developing countries?
- What are the success factors that have high leverage on e-government success? How do these success factors influence e-government success in term of the chain of causal relationships between eGSFs?

4.3.2 Research outcomes

The research will result in a system dynamic model that formally explicates the dynamic feedback relationships of the eGSFs within the scope of the research. It is a computer simulation model that serves as a virtual world where managers/decision-makers of an e-government system in developing countries can experiment with their decisions. The experiment informs the consequences (what will be affected at what level and when) for any decisions they make. The model also facilitates learning by understanding the trace effect of changing a factor and the impact of this change.

The outcomes of the research are that the model users (decision-makers, managers and developers) understand and become aware of the complexity of eGSFs (their wide dimension and relationships) and that in many cases the relationships between eGSFs are more important than the eGSFs themselves. This understanding enables them to make fruitful decisions, take actions to ensure e-government's long-lasting success, and have sound justification for any decision made.

4.3.3 Overall design

Considering the key research question and the adopted research approach, the overall research design to formally model eGSFs and their relationships is described in Figure 4.2. The design consists of two main stages, the first being the eGSFs system decomposition and the second the eGSFs SD modelling.

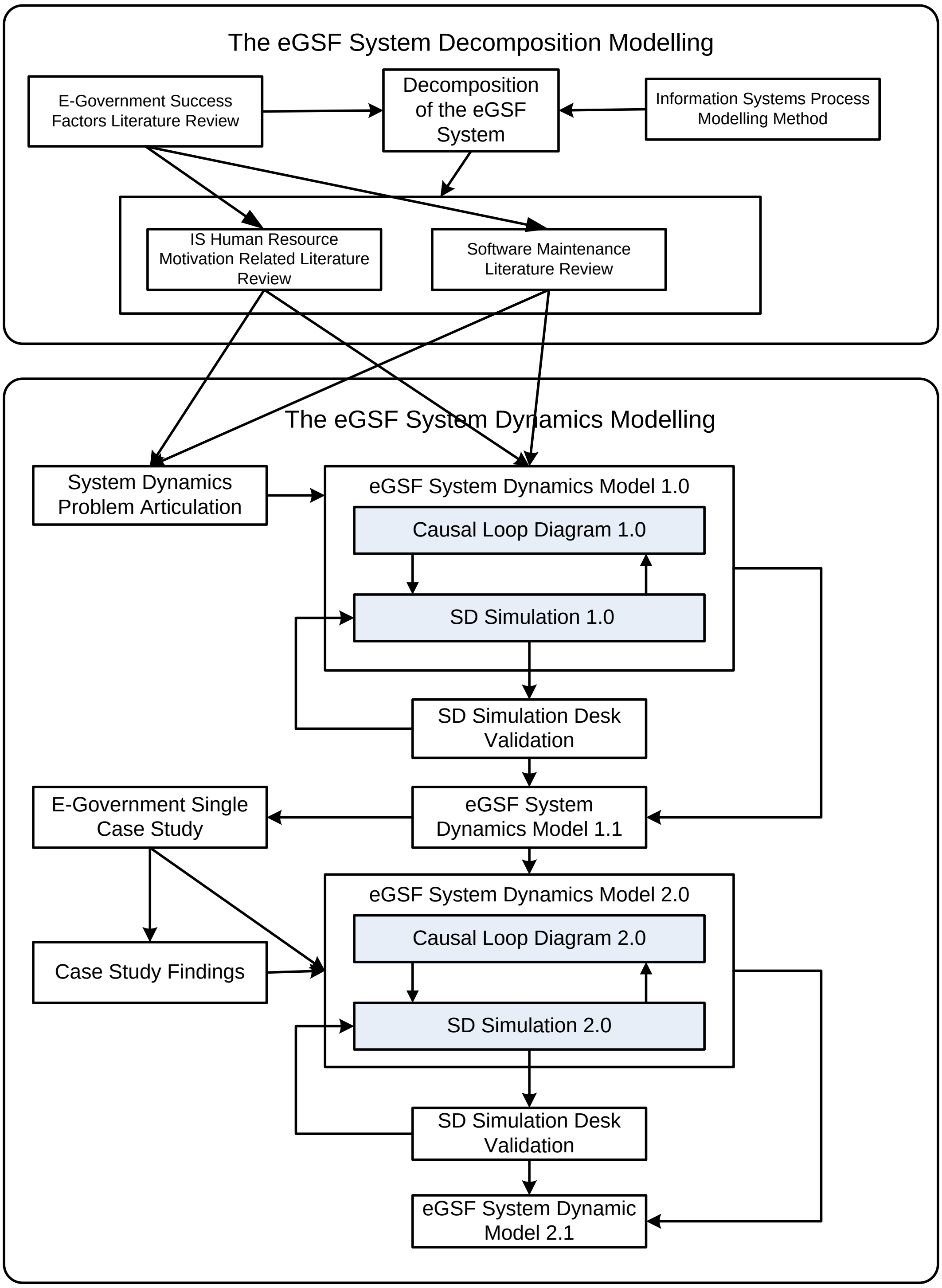


Figure 4.2 *Overall research design.*

The first stage is meant to view the eGSFs as a whole system or to see the "forest" (Maani & Cavana 2007**:** 9) which consists of success factor subsystems. This stage results in a presentation of the eGSFs system which provides a holistic view of the success factors and gives the context of any detailed model of the eGSFs system. At this stage, the way the eGSFs system and subsystems will be presented relies solely on the comprehensive review of the relevant eGSF literature. This includes the way the eGSFs system is decomposed into lower-level subsystems and sub-subsystems. The decomposition process is elaborated further in sub-section 4.4.2. It must be noted that, as the purposes of the decomposition modelling in this study are to obtain a broad overview of the eGSFs system, to organise the eGSFs system in a manageable number of subsystems and sub-subsystems and to serve the context of a more detailed success factor dynamic relationships modelling, the validation of the system decomposition will not be pursued.

The second stage is intended to investigate eGSFs and their relationships in greater detail, or to see the “trees” (Maani & Cavana 2007**:** 9). Based on the research questions and referring to the subsystems or sub-subsystems of eGSFs system decomposition, this is the stage where a formal model of the complex feedback relationships of the eGSFs, resulting in an eGSFs SD model, will be developed. This stage specifically responds to the specific and detailed research question: to formally model the success factors and relationships of e-government system SM and IS staff of the e-government systems that lead to e-government success in developing countries.

The model development of this stage starts with articulating SD problems. In addition to the research question, this narrative problem formulation uses the factors and relationships identified from the e-government system SM and the IS staff motivation related literature review. This formulation, which also reflects the boundaries of the system under interest, is used as the basic “information” for eGSFs SD model version 1.0.

The next step is to construct eGSFs System Dynamics model version 1.0, which is represented as a CLD and an SD simulation model. The first diagram is the qualitative representation and the second is the quantitative representation of the eGSFs SD model. The CLD also serves as the dynamic hypothesis, referring to Figure 4.1. The literature review of the SM and IS staff motivation-related success factors and their relationships will be used as the source for the SD model development. The CLD will be constructed first and then, based on this diagram along with the information drawn from the literature, the initial SD simulation (version

1.0) will be developed. The model will be developed in a stepwise fashion. For the rest of this chapter, “the model” means the SD simulation model or quantitative representation of the eGSFs SD model.

At this stage a validation process takes place along the way as the model is being developed. As more elements and relationships are added from the relevant literature, the validation process will be performed on a computer by evaluating and testing the model against some criteria of SD method. Equations and structural relationships that are gradually embedded within the model will also be manually and visually evaluated. Therefore, this validation process can be called a desk validation. It results in a refined model, which is in turn used as an input for the CLD refinement. Overall, the output of this process is eGSFs SD model version 1.1. The elements of SD validation process will be presented in section 4.5.

The model version 1.1 needs to be validated by a real e-government system. A case study will be conducted for this purpose, to ensure that the model is confirmed with the real-world system in addition to the facts available in previous research. Data from the case study will be collected using instruments developed from the eGSFs SD model version 1.1. The case study description will be given in subsection 4.5.3.

A case which satisfies a set of criteria stipulated by this research will be selected. The case is an e-government system in a developing country and should at least satisfy the working definition of the e-government success criterion. The findings will be presented and used to confirm or disconfirm the elements and relationships of eGSFs SD model version 1.1. The findings will also be used as an input to refine the structure, elements and relationships of the model, as well as to set its parameter values. Both the CLD and the model are refined and therefore their structure and parameters are validated by an actual e-government system. This results in an eGSFs SD model version 2.0.

It is highly probable that the SD simulation model version 2.0 is different from the model version 1.1 in terms of its structure, elements, relationships etc. Therefore, the model version 2.0 requires further desk validation using a set of criteria specified by the SD method. A result of this validation process is possibly further refinement of the CLD and the simulation model version 2.1 which can be used confidently to assist e-government decision-makers or managers to achieve e-government success from the perspective of SM and IS staff management.

The eGSF SD model version 2.1 is the validated and final formal model of this study. Therefore, the model can be used to give insight into the complex dynamic feedback relationships of the eGSFs of a successful e-government system in a developing country. The model can also be utilised to assist the e-government decision-makers or managers in developing countries to evaluate and analyse success factors and their relationships, reflecting policies that lead to e-government success or failure. These evaluations and analyses, or design, can be conducted by introducing the policies into the validated model by changing the structure, adjusting the level of influential factors or altering the time factor, and by running simulations of this adjusted model.

4.4 Model development and representation

4.4.1 Introduction

Following the research design, this section presents the technical details of the model development and representation. For this, the system decomposition will be presented first and followed by the SD modelling method.

4.4.2 Decomposition

As the first stage of the research, the system decomposition of the eGSFs provides a description of the eGSFs as a whole system which is decomposed into subsystems and sub-subsystems. Naturally, a system decomposition is a top-down process (Burton-Jones & Meso 2006). It starts with the system as a whole, identifies the subsystems according to a specific reference, then discovers further sub-subsystems if necessary. Analysts can conduct system decomposition and proceed in their own way (Paulson & Wand 1992). In spite of a lack of theory regarding the sufficient conditions for a good decomposition, modellers may decompose a system by referring to 1) functional, 2) behavioural, 3) organisational, and 4) informational aspects of the system (Curtis, Kellner & Over 1992). Among these references, the well-known Yourdon’s functional decomposition has been commonly implemented in the field of IS analysis and design (Burton-Jones & Meso 2006).

In this research, eGSFs and their relationships will be decomposed according to the functional criterion such that each subsystem, which consists of a set of closely-related success factors, represents a function or a role in achieving e-government success (Whitten & Bentley 2007). An organisation chart-like diagram will be drawn to represent this subsystem decomposition. This functional decomposition will proceed by decomposing each subsystem into sub-subsystems. A diagram depicting the decomposition will then be drawn for each subsystem and its sub-subsystem. Finally, these separate decompositions are combined into one whole diagram reflecting the system, subsystems and sub-subsystems. Because the aim of the decomposition process is to represent diagrammatically the top-down functional decomposition and system structure, the relationships among the factors will not be identified (Whitten & Bentley 2007). Therefore, the resulting diagrams do not inform the relationships among these subsystems.

4.4.3 Problem articulation

As with any other modelling effort, the SD modelling needs to identify and formulate the problem to be addressed clearly and purposefully (Coyle 1996; Maani & Cavana 2007; Sterman 2000). This includes:

- identification of the SD model purpose
- narrative description of the dynamic feedback relationships problem
- identification of the main variables and concepts
- specification of future and past time horizons
- graph development depicting problem behaviour over time (or reference modes) (Sterman 2000).

For this research, the formulation will begin with specifying the dynamic behaviour of e-government success and the time horizon. A time horizon will be set up for e-government system success and various possible dynamic behaviours over this time horizon will be identified. As specified by the scope of the research and as argued in Chapter 3, e-government system software sustainability represents e-government success. Therefore, e-government system software sustainability will be plotted over a two-dimensional graph in which the time range is set as the horizontal dimension and the level of e-government system software availability as the vertical dimension. As also argued in Chapter 3, sustainable software means a relatively high level of software availability over time. This graph illustrates the reference modes of e-government system success.

4.4.4 Causal loop diagram

A CLD consists of words or phrases which are linked by curved arrows, each of which has attached polarity and time delay symbols (Morecroft 2007; Sterman 2000). The arrow represents a causal relationship between two factors. The diagram visually conceptualises the dynamic relationships associated with the problem being studied. A CLD symbolises a framework emphasising the relationships instead of the things and focusing on the dynamic change instead of static snapshots (Senge 2006).

As a dynamic hypothesis, the diagram offers an explicit provisional presentation of the researcher's understanding and view of the causal feedback relationships and dynamics structure of the problem (McLucas 2005; Morecroft 2007; Sterman 2000). It is subject to modifications, improvements or even objections during model development and learning from the real world through the iterative process (McLucas 2005; Sterman 2000). Furthermore, Sterman (2000: 95) states "[a] dynamic hypothesis is a working theory of how the problem arose".

The CLD is formulated by referring to the relevant theories associated with the formulated problem. The formulation is also based on the researcher's mental model associated with the research problem. This hypothesis development puts together the main endogenous factors (that arise from within the system), reference modes, time delay, feedback structure and currently-used theory that explain the dynamic behaviour of the problem.

The development process begins with a simple causal feedback relationship structure originating from the main endogenous factors of the problem formulation. The formulation will be improved in detail through the inclusion of relevant endogenous factors or further elaboration of the current dynamic structure of the dynamic hypothesis.

Within the current context of this study, starting from the dynamic behaviour of e-government system software sustainability, endogenous success factors within e-government SM and IS staff motivation domains that cause this dynamic will be brought forth. Included within the identification process is the time delay taken by a success factor in influencing other success factor(s). A success factor of the eGSFs model being developed by this study can be a condition, situation, action, decision or physical element's condition within the domains that can influence and be influenced by other success factors; and both quantitative and qualitative success factors are possible.

The success factors within SM, the expectancy theory and IS human resource management are the endogenous ones, while those from three other high-level eGSFs (see section 3.2) and other stages of the e-government system life cycle (see section 2.2.3) are considered exogenous – that is, a variable's values or behaviours are assumed or given, and the chains of cause and effect that lead to an increase or decrease of the exogenous level are not pursued. The differentiation of the endogenous and exogenous success factors provides information about the boundary of the model. The decision on whether a success factor is exogenous or endogenous is dependent on the scope of the problem.

After the identification of the success factors, chains of pairs of cause and effect of the success factors that form a causal-loop relationship will be identified. The process starts from the key success factor. It will be then followed by identifying other relevant success factors that influence the key success factor. This process continues such that the last factor is influenced by the key success factor. Accordingly, this chain of causal relationships forms a feedback loop. The resulting causal-loop relationships provide a provisional explanation about how the success factors are linked together and how the dynamic behaviour of the problem can be explained.

- ***Technical aspects of the causal loop diagramming***

To construct the diagram, the following technical guidelines (Coyle 1996; Maani & Cavana 2007; Morecroft 2007; Sterman 2000) were utilised. To properly construct the diagram, the guidelines should be followed closely (Sterman 2000). The brief guideline is described as follows.

The causal relationship between a pair of success factors is depicted by a curved arrow with the arrowhead pointing to the effect factor. During the identification of the cause effect, the polarity of this causal relationship also needs to be identified to find out whether an increase or decrease of the cause factor causes an increase or decrease of the effect factor. A "+" or "—" sign is attached next to the arrowhead, where "+" means both success factors change the same direction while "—" means the pair changes different directions. During this identification, the name of the success factors is associated with a positive perception and is clearly associated with an increase or decrease of its value. The polarity of a cause effect depends on circumstances or assumptions adopted; therefore it is possible to have different polarities under different assumptions. However, any link between two success factors should not have ambiguous polarity. If such ambiguity exists, it usually means there are more causal paths connecting the two success factors and it is necessary to make all the paths explicit.

In addition, it must be noted that in many cases the effect of an action or decision or policy is not always immediate. There is a time delay in order for a cause to have a full effect and this delay creates dynamics. Therefore, within the causal-loop diagram modelling it is very important to identify and show the delay. A delay is represented by a double-slash drawn crossing the arrow.

Furthermore, once a feedback loop is revealed it is necessary to identify its type. There are two possible feedback types: reinforcing or positive and balancing or negative feedback loops. If initially an increase or decrease in one particular success factor within the loop results in a further increase or decrease in that success factor over time, then the loop is a reinforcing feedback loop; otherwise it is a balancing feedback loop. A reinforcing feedback loop creates a persistent increase or decrease, while a balancing feedback loop causes oscillation over time that leads to a particular level. Every feedback loop contains a story.

4.4.5 System Dynamics simulation

- ***SD simulation model construction***

Once the CLD is considered sufficient and able to represent the nature of the dynamic relationships of the system being modelled, the next step is to construct the SD simulation or the model. The main elements of the model are stock (or levels), flow (or rates) and converter (or auxiliaries); therefore, the model is also named as a stock-flow diagram. Table 4.1 provides a technical description of these elements.

The construction of the model will begin by drawing a tentative high-level sectors diagram of the system (Maani & Cavana 2007). The sectors show sub-models of the possible simulation along with linkages indicating physical and/or informational flows between the sectors. The linkages are added gradually as the SD simulation construction improves in detail.

Following the sector diagram is the identification and construction of stocks, flows, converters and feedback loop connections among these components (Sterman 2000). The stocks of the model will be identified and defined by looking up the elements of the system domain that can accumulate over time, irrespective of whether the elements are physical or not, while the flows of the model are the elements that change the level of the stock per unit time. During the initial model development, the value of stocks will be determined from the relevant literature if available, or from the researcher's plausible judgement. The rate of information or physical inflow or outflow of a stock will be formulated according to the SD simulation elements connected to the flow.

The relationship form of the model elements, the initial and parameter values of the model describing the dynamic structure of the system, will also be identified and defined. However, unlike the stocks and flows, the forms of the relationships between model elements and the parameter and initial values of the model cannot be identified or derived from the causal loop diagram; initially they will be identified and/or derived from information and data from previous relevant empirical research. It must be noted that the data and information to develop the SD simulation is not restricted to empirical research only; the sources of the data and information are diverse and include "observations, interviews, meetings with managers, company records, historical records and archives, statistical publications, survey responses, media reports and so on" (Maani & Cavana 2007**:** 66).

In addition to describing the dynamic structure of the system, the form of relationship between the model elements, the initial and parameter values will also be used to instantiate the model. The formulation of the relationships involves drawing two-dimensional graphs that represent causal non-linear relationships between two variables, or the effect of time delay on the level of a variable. The dependencies among model elements can also be represented as a mathematical formulation, such as a weighted average of two or more model elements. The relationships, initial values, constants or mathematical formulas in general are embedded within the converter. The model will be developed gradually from a simple to the more complex one.

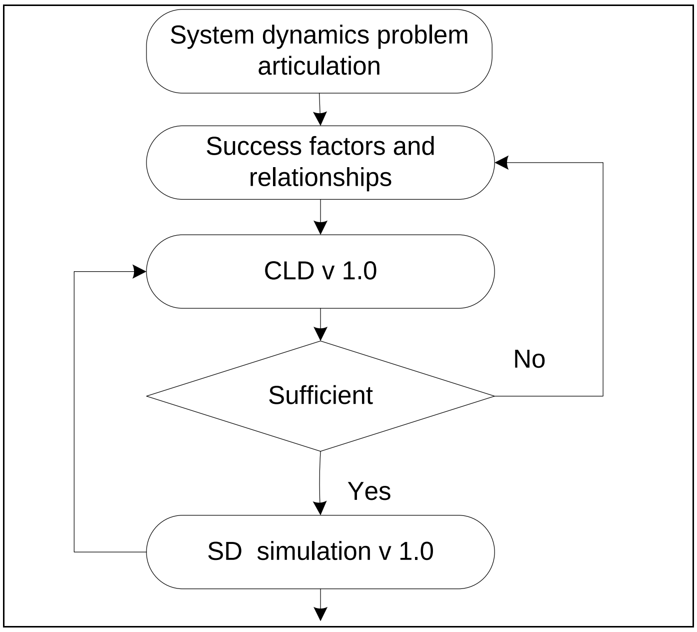


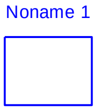
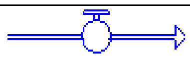
Figure 4.3 *Correspondence between CLD and SD simulations*

As the SD simulation (the model) provides much more detail and explicit description of the elements and their relationships, ambiguous concepts or representation within the CLD can be revealed, evaluated and refined. Correspondence between the CLD and the model is presented in Figure 4.3. This figure describes the iterative nature of SD model development between a causal loop diagram and SD simulation.

- ***Technical aspect of the System Dynamics simulation***

The elements of the SD simulation are represented as a collection of symbols within special computer software. Technically, Table 4.1 provides a list of symbols used to create a stock flow diagram for computer simulation using *STELLA*® software.

Table 4.1 *Elements of the SD simulation using STELLA® software*

Element	Notation	Description
Level or stock		A level or stock is an accumulation of things over time. It represents a container of physical or non-physical material, or information. Only inflow and outflow rates will increase or decrease the level. The level or stock represents the state of variables in a system and are determined by the rate of past accumulation.
Source/sink		A source and sink is an external source and repository where flows come from and go to. Their existence represents the accumulation that happened beyond the boundary of the modelled system. The capacity of source or sink is infinite.
Rate or flow		Rates or flows represent the "actions" in a system. They cause change in the level or stock. Rates may represent decisions or policy statements. Rates are computed as a function of levels, constants, and auxiliaries.
Converter or Auxiliary		A converter or auxiliary converts input to output. It provides detail of the structure of stock and flow, and informs a causal-loop relationship. It connects a level to a rate. Often, it represents a constant, mathematical equation or graphical relationship.
Information link		Information linkages are used to represent information flow. They link rates to levels or auxiliaries. To play their control mechanism, rates need a connection to other variables (levels or auxiliaries). Information links help to close causal relationships.

Source*: Adapted from Madachy (2008: 57)*

How these SD simulation elements work and what the SD simulation represents can be described by the following illustration, Figure 4.4.

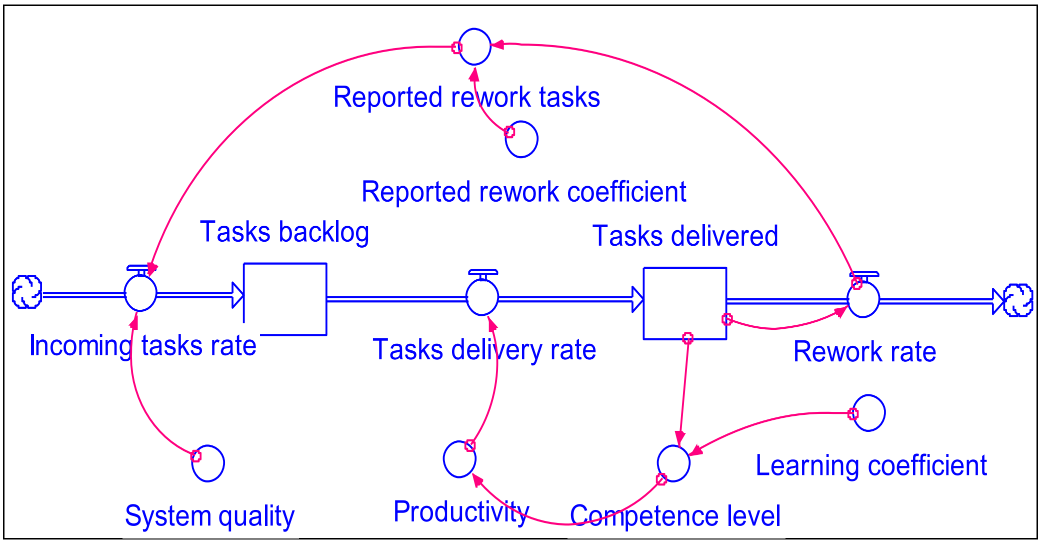


Figure 4.4 *A simple example of an SD simulation*

This simple stock flow model informs that there exist tasks flowing in at the rate of *Incoming_tasks_rate* and this causes a stock *Tasks_backlog* level to increase. The tasks can be physical or non-physical. The incoming rate is a function of *System_quality* and *Reported_rework_tasks*. The level of *Tasks_backlog* is reduced by *Tasks_delivery_rate,* which is, in turn, determined by a converter *Productivity* that keeps the value of, for example, employee productivity. This rate will increase the level of *Tasks_delivered*. As the delivery process may not be perfect, some tasks need rework. However, it is possible that only a portion of the required reworks are reported, which are specified by *Reported_rework_coefficient*. These reported reworks add to the rate of incoming tasks. Observation of this flow indicates the existence of a feedback loop. Another feedback loop can also be identified from this SD simulation, that is, the number of tasks being delivered can increase *Competence_level* by a certain amount determined by *Learning_coefficient*. The *Competence_level,* in turn, influences productivity and therefore closes the loop.

The converter, such as *Reported_rework_tasks, Productivity* and *Competence_level,* may contain a constant, equation or graph that represents a relationship structure between variables. The converters, stocks and flows are linked by connectors.

4.5 Model validation

4.5.1 Introduction

Prior to the use of the SD model for decision-making or for any other purpose, the model needs to be validated. In the SD method, model validation is meant to build the soundness and usefulness of an SD model and to ensure SD model users' confidence in using an SD model (Forrester & Senge 1980). It must be noted that this confidence is built up gradually in a series of tests which is naturally not sequential but iterative, following the iterative nature of the SD modelling process (Barlas 1996). In SD, "[m]odel validation is a gradual process of 'confidence building', rather than a binary 'accept/reject' division" (Barlas 1996: 188). This validation process is only applied to the SD simulation model.

4.5.2 Model testing

Model testing is a comparison between the model and empirical reality in order to confirm or disconfirm the structure and behaviour of the model. With respect to the scope and purpose of modelling being studied, this testing is needed to ensure that "there is nothing in the model that is not in the real system and nothing significant in the real system that is not in the model" (Maani & Cavana 2007: 70). By "empirical reality" is meant "information in many forms other than numerical statistics" (Forrester & Senge 1980: 210). This testing is a process for building confidence in the model, as stated by Forrester and Senge (1980: 209) that:

> There is no single test which serves to 'validate' a system dynamics model. Rather, confidence in a system dynamics model accumulates gradually as the model passes more tests and as new points of correspondence between the model and empirical reality are identified.

There has not been well-established formal model testing with formally established steps like in statistical methods (Barlas 1996). Forrester and Senge (1980: 209) stated that:

> The nature of SD models permits many tests of model structure and behaviour not possible with other types of models. Conversely, some widely used tests, such as standard statistical hypothesis tests, are either inappropriate or, at best, supplementary for system dynamics models.

Generally, there are three categories of SD model testing (Coyle 1983; Forrester & Senge 1980; Morecroft 2007). For example, Morecroft (2007**:** 377-8) presented the three categories of testing as:

- Tests of model structure – to assess whether the feedback structure and equation formulations of the model are consistent with the available facts and descriptive knowledge of the real-world system,
- Tests of model behaviour – to assess the fit of simulation to observed real-system behaviour,
- Tests of learning – to assess whether model users have gained new insight about system structure or learned something new about real system behaviour.

For this model testing, Forrester and Senge (1980) and Sterman (2000) have provided detailed tests to build confidence. Forrester and Senge (1980) identified 17 main tests for the SD modelling, the core of which is presented in Table 4.2 by Maani and Cavana (2007**:** 72).

- ***Test of model structure***

The structure test is a means to build confidence in the model by examining the variables and relationships structures of the variables and seeking their correspondence with the real world (Morecroft 2007 Ch. 10) of e-government systems. In doing this test, the association between the structure and the behaviour of the model is not evaluated (Forrester & Senge 1980). Considering the order of performing the validation process, this is the first test that will be carried out before the two other test categories (Barlas 1996), see Figure 4.5.

Table 4.2 *Testing for SD simulation models*

Test of model structure	
Structure verification	Is the model structure comparable with the structure of the real world?
Parameter verification	Do the model parameters (constants) correspond conceptually and numerically to the real system (i.e. real life)?
Extreme conditions	Are the rate (policy) equations plausible if imaginary maximum and minimum values of each state (stock) variable (or combination of state variables) on which they depend are inserted into the model?
Boundary adequacy	Is the level of model aggregation appropriate and does the model include all relevant structure?
Dimensional consistency	Are the rate equations dimensionally consistent and do they include 'scaling' parameters that have little or no real-life meaning?
Test of model behaviour	
Behaviour reproduction	How well does model-generated behaviour match observed behaviour of the real system?
Behaviour anomaly	Can a model assumption be defended if implausible model behaviour occurs if the assumption is altered?
Behaviour sensitivity	Can plausible shifts in model parameters cause a model to fail model behaviour tests previously passed?
Test of policy implication	
Changed behaviour prediction	Does the model correctly predict how the behaviour of the system will change if a governing policy is changed?
Policy sensitivity	To what extent are policy recommendations altered by plausible changes to parameter values, and what risk is indicated in adopting a model for policy making?

Source*: Maani and Cavana (2007**:** 72)*

Structure verification. This test will be performed to ensure that assumptions, feedback structure and relationships structure of the model do not contradict the essential elements and structure of the actual e-government system relevant to the model purpose (Forrester & Senge 1980). The significant aspects of the e-government system can be obtained from the knowledge of the people who have intimately dealt with or been important parts of the system (Barlas 1996; Morecroft 2007). This verification can also involve a comparison with the knowledge found in previous empirical literature or theories (Barlas 1996; Forrester & Senge 1980) and is highly qualitative (Barlas 1996; Luna-Reyes & Andersen 2003).

As this test is "intrinsically part of constructing a system dynamics model" (Forrester & Senge 1980: 226), therefore it will be performed once a first pair of stock and flow is constructed. To show that the model can be structurally verified, it will be necessary to indicate qualitatively that the model is consistent with the real system relevant to the modelling problems and purpose (Barlas 1996). As technical guidance, several aspects of the SD simulation will be examined that include conservation laws and physical laws: stock level that might go negative; practicality of the decision rules for the SD simulation user; and the existence of assumptions for nothing (Sterman 2000).

Parameter verification. This test is a mean to ensure that the parameters of the model can be justified conceptually and numerically in a real e-government system (Barlas 1996; Forrester & Senge 1980: 213). Conceptual verification means that the adopted parameters exist within the system, and numerical verification means that the choice of the numerical value of the parameters lies within an actual maximum and minimum interval. As with the structure verification, this verification can also involve a comparison with the knowledge found in previous relevant empirical literature or theories (Barlas 1996).

Extreme conditions. This test will be performed by introducing appropriate extreme values (negative infinity, zero or positive infinity) to the model (Forrester & Senge 1980). The plausibility of the resulting values will be compared with the knowledge of real situations when the same extreme conditions occur. Forrester and Senge (1980) stated two reasons for the importance of this test: it can disclose flaws in the model which can go unnoticed under normal conditions; and it is able to improve the model usefulness by evaluating the system when it is forced to operate beyond normal conditions.

Boundary adequacy. This test means to evaluate of the appropriateness of the model aggregation and the relevance of the model structure with the SD modelling's purpose which includes "developing a convincing hypothesis relating proposed model structure to a particular issue addressed by a model" (Forrester & Senge 1980: 215).

Dimensional consistency. The dimensional consistency test aims at ensuring the measurement units at the left-hand side and right-hand side of an equation in the model are equal. It is also useful in revealing whether a parameter included in the equation is meaningful as a component of structure of the model. A failure to pass this test indicates a faulty model structure (Forrester & Senge 1980: 216). For this study, this test will be performed by evaluating the result of the STELLA® software facility during the model development and refinement process.

- ***Test of model behaviour***

This test "evaluates adequacy of model structure through analysis of behaviour generated by the structure" (Forrester & Senge 1980: 217). The test is performed once confidence in the model structure has been built up. It will examine the closeness of the behaviour pattern produced by the model to that exhibited by an e-government system. Any revision suggested as a result of conducting this test involves only model parameter changes (Barlas 1996).

Behaviour reproduction. This test will be performed to evaluate the association between the time-series output representing the behaviour produced by the constructed model and the historical time-series data reflecting the behaviour of the real system in the past. Statistical tools such as Theil's Inequality Statistics might be useful to assess point-by-point agreements. However, tools of this type should be used cautiously because this type of fitness test cannot prove a model structure is correct and superior to other structures (Sterman 2000). Further, Sterman showed that it is possible for different structures to produce a set of time-series data which fits with historical data but deviates significantly outside the historical range.

Therefore, following Sterman (2000), to perform this test the time-series output produced by the model which has passed previous tests will be plotted and examined against the historical data of e-government success. The closeness of the modes of behaviour of these two sets of data, such as the phase, amplitude and frequencies that might reflect possible asymmetries and other subtle features, will be assessed qualitatively.

Behaviour anomaly. This test will be carried out to evaluate whether anomalous behaviour takes place if a structural relationship in the model is deleted or modified. The existence of an anomaly provides evidence of the significant role of the relationships (Sterman 2000). A loop-knockout analysis will be performed for this test, such as by setting an adjustment time of a flow to an infinite value, a non-linear function to a constant, or replacing a disequilibrium structure with an equilibrium subsystem (Sterman 2000).

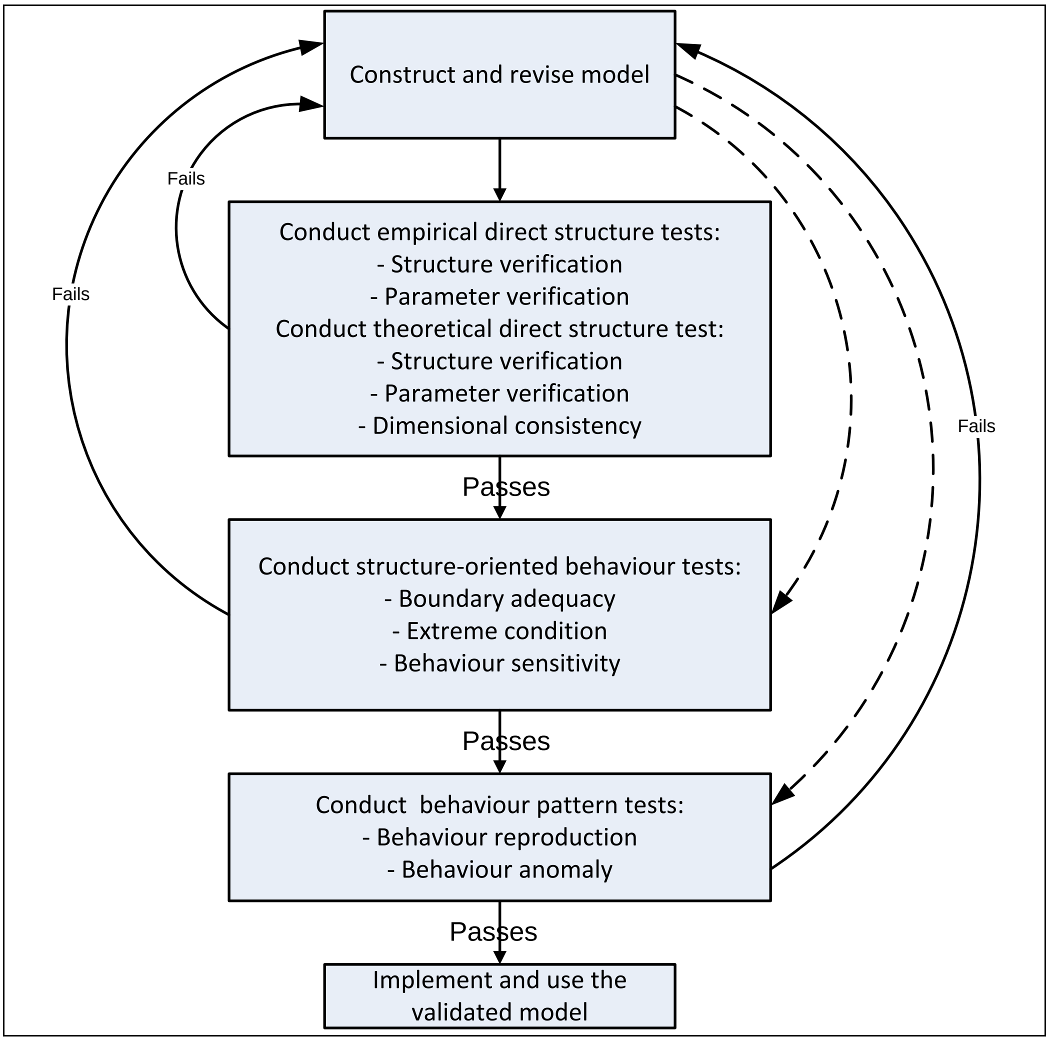
Behaviour sensitivity. This is to test if plausible changes in the model parameter cause unacceptable behaviour of the structure despite the structure having passed some tests previously. It aims at evaluating uncertain or imperfect assumptions about initial values, parameter values and structural relationships that are introduced to the model. It should be noted that at the beginning of the model development stage the initial values, elements and relationships are required to instantiate simulation. However, because they are initially uncertain, these values and relationships should be settled by subsequently altering the values within a reasonable range, running the model and inspecting the change of the dynamic system behaviour. If the parameter with such changes cannot be identified, confidence about the model structure is improved. The test will be performed through the facilities available within the STELLA® software by setting a range of values, such as 10% higher and 10% lower, of a selected parameter. The selected parameters are those that are suspected to be "both highly uncertain and likely to be influential" (Sterman 2000**:** 884).

- ***Test of policy implication***

This set of tests involves a comparison of the impact of policy changes between the model and the associated actual system. The tests try to judge whether the impacts as a result of policy changes in the real system are the same as the impacts predicted by the model. As these tests necessarily involves actual changes, which are beyond the aims and scope of the study, they will not be pursued.

The implementation of validation process, described in Table 4.2, will follow the sequence suggested by Barlas (1996) as presented in Figure 4.5. In this figure, the test of model structure is divided into two categories: direct structure tests and structure-oriented behaviour tests. It indicates that the validation process can involve a number of iterations and model revisions. Dashed-lines in the figure indicate the possibility of omitting direct-structure and structure-oriented behaviour tests for the subsequent test, only if the model revisions required by the subsequence tests are non-structural.

In addition to the tests mentioned above, several steps will also be applied in order to build confidence in the model (Coyle 1996**:** 96-7). They include the evaluation of the correspondence between the CLD and the SD problem articulation and the examination of the association between the CLD and the model.



***Source**: Adapted from Barlas (1996: 194)*

***Figure 4.5** The steps of SD simulation model validation process*

4.5.3 Case study

A case study will be undertaken to ensure that all elements and relationships of the SD model exist in a real-world system, and all significant and relevant elements of the real-world are properly modelled (Maani & Cavana 2007). The case study is performed to implement the model structure test of model validation, especially the structure verification and parameter verification tests. This empirical test is first a means of collecting from the system information of interest and using the findings to confirm or disconfirm the structure and parameter of the SD model (Barlas 1996). Second, it combines the empirical and theoretical verifications to modify and refine the SD model, such that the significant elements of the system are in the model properly, and that the dynamic behaviour of the system of interest can be reproduced by the model (Abdel-Hamid & Madnick 1991). As part of building confidence in an SD model, a single case study has been conducted by much SD research such as within the seminal work of Abdel-Hamid and Madnick (1991) and the doctoral dissertation of Fryling (2010).

- ***Case selection***

According to Yin (2003**:** 40-2) there are five possible reasons for conducting a single case study. The case may be the critical case in testing a well-formulated theory, an extreme or unique, a representative or typical, a revelatory, or a longitudinal one.

Following these possibilities for conducting a single case study, the selected case will be a successful e-government system in Indonesia. The case is believed to be representative among many e-government systems that can achieve success in a developing country context, and a revelatory one as there are many e-government systems in developing countries, including Indonesia itself, still experiencing various degrees of failure. In order to satisfy the requirements for a single case study, the case must satisfy a general criterion of this study: it must have attained success according to the working definition of e-government system success formulated in this study.

Therefore, referring to the working definition of e-government success, an e-government system case will be selected from the secondary data of the winner of the Indonesian eGovernment Awards. The case will be selected based on its sustainable achievement in the Indonesian eGovernment Award program in the sense that it won the award or was within the top three for several consecutive years. The use of the Indonesian eGovernment Award as one of the criteria is based on the fact that the program is organised by a non-government organisation which can provide objective judgement, it had been conducted as a national event for more than seven consecutive years, and the award has been considered by many Indonesian government units as a symbol of high-level achievement. As there are several award categories, a government unit at national level or ministerial level will be selected.

Before the selected case is contacted about its availability for participating in this research, information will be collected from its website to ensure that the case developed its e-government application systems as an in-house project, and the application systems are currently in an in-house maintenance stage. An e-government systems unit that does not meet these criteria is not suitable for this research. Once the selected case is considered appropriate, a letter will be sent by the researcher to the senior manager requesting their availability to participate in this research.

- ***Design for the case study***

Yin (2003**:** 21) underlined five components that should be prepared for a case study:

1. a study's questions
2. its propositions, if any
3. its unit(s) of analysis
4. the logic linking the data to the propositions; and
5. the criteria for interpreting the findings.

In accordance with these five components, the following descriptions are prepared to ensure that the case study is properly designed and appropriate evidence can be collected.

The question. The most appropriate questions for performing a case study are the ones that pertain to "how" and "why". As formulated in the research design section, this research seeks to answer how the eGSFs relate to each other and eventually cause the achievement of e-government success.

The study proposition. An SD model of version 1.1 serves as the proposition which guides the investigation of the elements within the scope of a case study. In addition to its role as a reflection of the adopted theory, the SD model of version 1.1 directs the study in collecting the empirical evidence.

The unit of analysis. This is a crucial aspect of a case study and should be specified at the outset of the study, as it is about defining what the "case" for the study is. A selected e-government system which satisfies a stipulated criterion is the unit of analysis for this study. The description of the unit of analysis has been presented in the Case selection subsection above.

The link between data and proposition and *the criteria for findings interpretation* is associated with how the data will be collected and interpreted. As mentioned in the previous section, SD model development allows the use of various types of data.

The data that will be collected about and from the case is also in a variety of forms. It includes verbal responses from interviews with internal IS staff within the case, documentation, software and software demonstration etc. For this, the following subsections present in detail how these activities will be performed.

- ***Respondent selection***

Respondents are an important source of data for SD model development and validation. As mentioned in the previous section, the respondents should be dealing intimately with the actual system of interest so that they have deep knowledge in association with their responsibility for the system. From them it is expected that the structure and parameters of the SD model can be confirmed or disconfirmed, and the significant elements of the case can be obtained to refine the SD model.

To select the respondents, the information available on the website of the selected e-government unit will be used as the first but provisional source. Three types of respondent will be chosen from the selected case study based on their managerial level position: senior managers, operational managers and senior staff.

The senior managers are expected to provide the overall description and policy of the e-government system in relation to the research focus. The operational managers are expected to provide data regarding success factors and the relationships being investigated by this study based on their experience in managing the maintenance of the e-government system. The senior staff who are involved in e-government SM are expected to give data based on their experience and views on running the system. The selection of the senior staff will be based on, especially, their involvement in the implementation and maintenance of the system software. It will also be based on permission being granted by the senior managers of the selected case.

In addition to the respondents, the documents relevant to the case, both internal and external, will also be collected to be used as another data source and studied to corroborate the interview findings. The internal data documentation requested from the e-government unit will include the application manuals, output forms, examples of requests, official letters etc. It is expected that the documentation will provide raw facts that reveal the dynamic behaviour of the system, its structural relationships or constants.

- ***Data collection techniques***

From the selected respondents, the data will be collected through interviews guided by open-ended, semi-structured questionnaires. Three different questionnaires will be prepared in accordance with the three different respondent categories. The conceptual CLD of version 1.1 and SD simulation structure of version 1.1 resulting from the SD model development stage will be used to prepare the questionnaires. One questionnaire will address a broad overview of the e-government system of the case and will be used to interview the senior managers. Another questionnaire, longer than those for senior managers, will be used to get information from operational managers. This second questionnaire covers the application systems maintenance management and human resource management of the maintenance staff. The third questionnaire will be used to interview senior staff and focuses on SM activity, motivation-related factors and competence. All questionnaires will be written in the Indonesian language.

In addition to the questionnaires, other data collection tools will be prepared. They consist of the proposed CLD and SD simulation model structures and graphical relationships. The CLD of the SD model version 1.1 will be presented stage-wise, starting from the simplest loop to the more complicated ones. Each stage is presented on one piece of paper. This CLD will be used to interview the senior managers by presenting and explaining the meaning of the diagram to obtain their confirmation or disconfirmation. The SD simulation model version 1.1 will also be presented and explained to the senior managers sector by sector. The senior managers' views towards this model structure will also be pursued.

Prior to their interview, all respondents will be provided with general information about the research and the voluntary and confidential nature of the interview, and asked to sign the letter of consent to their participation in the interview and to the use of a recording device during its course. In addition, a concise overview of the research, some technical explanations and definitions related to the SM will be provided before the interview begins. This is to establish common meanings for some SM terminology. The interviews will be conducted based on respondents' time availability and in the Indonesian language.

- ***Data analysis techniques***

Data gathered from various sources can be qualitative or quantitative. The collected data from the case study can be verbal responses of the respondents, documentation and graphs.

To analyse the case study data, the data that provides a general profile of the selected e-government system will be presented first. This is to show that the selected e-government system is significantly appropriate within the scope and boundaries of the research problems being studied and satisfies the working definition of e-government success. The descriptive presentation covers the general profile of the selected e-government system, the implemented e-government system and services provided, the e-government application system sustainability, the in-house application systems maintenance management and the management of the internal human resources with specific competence in ICT.

As the aim of the case study data collection is to verify the structure and parameters of the SD model and the questionnaires are designed based on the SD model of version 1.1 that represents the proposition of the SD model, therefore the elements, relationships, quantitative values and time delay of the models will be used as the referents to analyse the collected data (Yin 2003). Thus, guided by both the CLD and the model structure, the data analysis will be undertaken.

It starts with the presentation of the CLD as a result of confirmation or disconfirmation from the senior managers of the case. Addition, deletion or modification of the success factors and the causal relationships will be carried out first, followed by identification of new important causal loops, or reconstruction of the existing essential causal loops if necessary. The findings from the senior managers regarding the CLD will be corroborated by the data obtained from the operational managers and senior staff. The data analysis from these respondents will be organised in accordance with the success factors of the models and the causal relationships associated with those success factors.

The qualitative analysis will be conducted by understanding and comprehending the meaning and context of the qualitative verbal responses of each respondent, in the sense that the data will be understood and comprehended from the definition given to that element, and identification of what and how other elements relate to the elements within the models. Any indication from the interviews in relation to the success factors and their relationships, verbatim or derived, will be based on what the interviewees say and how they express the factors (Burchill & Fine 1997). Conclusions will be drawn from agreement as to the meaning of the responses among the majority of the respondents. Any disagreement will be brought forth and further explanation will be pursued from the response provided by that respondent.

On the other hand, the quantitative numerical data provided by the respondents will be averaged. Further explanation will also be pursued from outlier data. In addition, the quantitative graphical data will be analysed based on their agreement and a conclusion will be drawn for each topic.

- ***Model refinement***

For the model structure, the stock and flow and structural relationships will be compared with the case study findings and the model will be reconstructed based on the findings. In addition, the quantitative data for the initial values and parameter values of the model elements obtained from the case study will be used to replace the initial values obtained from the literature; the constant values, structural relationships represented by graphs and mathematical formulation will also be reconstructed.

The case study will result in CLD and SD simulation model version 2.0. As this refined model contains structural and parametric changes, the desk validation process will be undertaken. The output of this validation is SD model version 2.1.

4.6 Summary

This chapter has presented the research and modelling methodology to be employed in order to achieve the research aims. The presentation includes the adoption of the system approach, which is implemented as system decomposition and SD modelling.

The research design consists of a modelling process describing the essential steps of model development: eGSFs literature review, system decomposition, initial SD model development, case study and SD model refinement and validation.

The SD model comprises a CLD and an SD simulation model of the eGSFs and relationships. By considering the SD simulation model development and refinement process, it can be concluded that the model is constructed from a combination of knowledge obtained from previous relevant studies, a case study and the researcher's mental model.

The validation process is applied to the SD simulation using a set of model testing procedures. It serves to build confidence in the use of the computer simulation model. The validation only deals with the test of model structure and model behaviour.

A case study will be implemented to collect data from a real successful e-government system in a developing country and used as part of the model validation and refinement.

In the following chapter, Chapter 5, the methods will be implemented to develop the system decomposition of the eGSFs, the initial representation of the CLD and the SD simulation model. The case study realisation will be presented in Chapter 6, which describes the realisation of the case study data collection, the selected e-government system and the SD model validation and refinement.

Chapter 5 A Model of the Success Factors for an E-government System

5.1 Introduction

This chapter presents the development of the model in accordance with the research approach, modelling methods and techniques presented in Chapter 4. The construction of this initial model refers to the success factors and their relationships discussed in Chapter 3.

The chapter begins with the decomposition of the eGSFs system. This method will decompose the eGSFs system into subsystems and sub-subsystems of success factors, which is expected to provide a holistic view of those success factors. In this section, SM and IS staff subsystem selection for the focus of study is also presented.

In the next section, the implementation of the SD modelling method to develop a formal eGSFs relationships model is described. The presentation is divided into three subsections following the SD modelling process. First, the SD problem articulation is presented, including descriptions of model purpose, problem formulation for SD modelling, reference mode, time horizon and key factors. Second, a CLD of the success factors from the SM and IS staff subsystem is drawn and explained. This CLD qualitatively explains how the eGSFs relate to each other in feedback-relationships in influencing e-government success. Third, an SD simulation model of those success factors is constructed by referring to the CLD. The construction is categorised into five modelling sectors: maintenance process; motivational factors; staff competence; staff allocation and staff development. In contrast to the decomposition of the eGSFs that provides a broad view, this SD model presents the detailed relationships of elementary levels of the eGSFs.

5.2 The e-government success factors system decomposition

5.2.1 First level subsystems

The implementation of the functional system decomposition method, as presented in Chapter 4, requires identification of the subsystems and their role in e-government success. Referring to the extensive literature review provided in Chapter 3 section 3.2, which has identified a wide range of domains of eGSFs, categorised them into four high-level success factors and indicated their functions in achieving e-government success, this research argues that the eGSFs system can be decomposed into four eGSF subsystems, as depicted in Figure 5.1. This figure reveals the subsystems and their possible complex relationships within the eGSF system**.** However, how they actually influence each other will not be explored at this stage. It is expected that the relationships will emerge from the development of the success factor relationships at lower levels. This figure also conceptualises feedback relationships between the eGSFs and the success of e-government. In the next section, each of these first-level subsystems will be decomposed into its second-level subsystems or sub-subsystems.

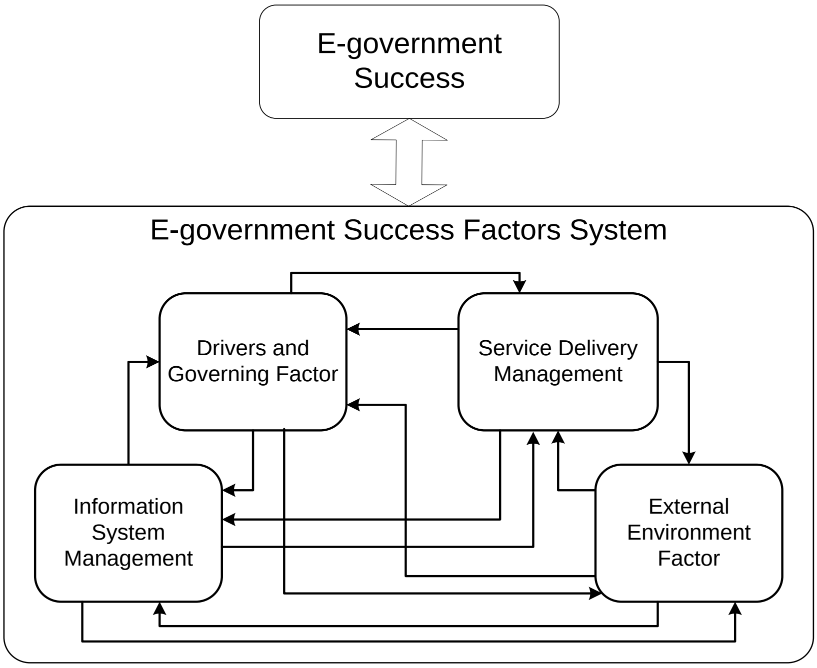


Figure 5.1 *The e-government success factor subsystems and their possible relationships*

Driving and Governing Factor subsystem. The function of this success factor subsystem is to ensure the successful ignition of an e-government initiative, to unite the various elements involved, to provide proper direction during the planning and design phases, to equip the e-government project with all the necessary resources, to maintain the required operation and growth level, as well as to assure e-government sustainability.

Information Systems Management subsystem. This subsystem deals with the management of software, hardware, network technology, data and information, human resources specialised in ICT, and integration of these elements. A successful impact of this subsystem is indicated by a highly reliable and available information system, such that it can sustainably support the human activities of e-government to deliver values and services to citizens and businesses.

Service Delivery Management subsystem. This subsystem covers all government entities that use and operate the e-government information system to deliver government values and services to citizens and businesses. A successful impact of this subsystem is indicated by highly available, relevant and accurate information and reliable, responsive and assured services of the e-government systems.

External Environment subsystem. This subsystem consists of entities or resources that are not part of the government, as well as activities or processes associated with them. The role of this subsystem is to ensure that the exchange of influence between the government and external environment leads to and supports e-government success.

5.2.2 Second-level subsystems

A further functional decomposition of the first-level subsystem is also based on the literature review provided in Chapter 3 section 3.2. A complete description of the eGSF system, subsystems and sub-subsystems as a result of this two-level-down functional decomposition process is presented in Figure 5.2. Each sub-subsystem cohesively contains factors and relationships, and is loosely coupled with other subsystems.

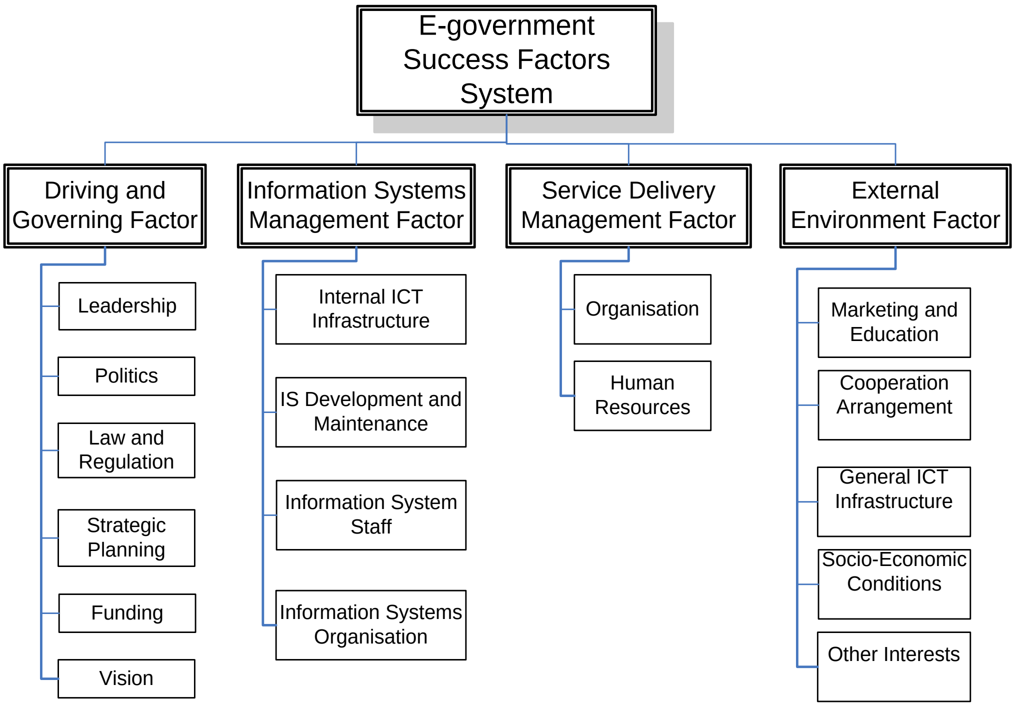


Figure 5.2 *An overall high-level representation of the eGSFs system*

The presentation of the detailed view of the eGSF system visualises the description of the broad range of success factors. It facilitates decision-makers in managing effort and harnessing resources to achieve e-government success, such as by focusing on (a set of) particular success factor subsystem(s), while at the same time being aware of the variability of the success factors that can influence the level of achievement of e-government development and implementation. In other words, the decision-makers can focus on a particular aspect while still adopting a holistic approach to attaining e-government success. A brief description of each of the sub-subsystems of Figure 5.2 is provided as follows.

Leadership. This sub-subsystem covers all level of government organisation leaders, their attributes and activities. For example, a top political leader and other lower-level leaders of a government organisation are required to exist in a government initiative, provide real support in the development, and give direction and necessary examples in the use of e-government system services.

Politics. This sub-subsystem includes success factors and relationships associated with the fact that a government is a political entity. Therefore any government decisions and activities always involve political aspects – for example, decisions to initiate an e-government system development, to establish an e-government unit and to provide sustainable funding – need political support.

Law and Regulation. This sub-subsystem consists of success factors and relationships relating to law and regulation required to govern the e-government system. For example, establishment of e-government units and the way the government provides services via an e-government system should be based on law. In addition, the usefulness of the information as an e-government product depends on the legal value of the information.

Funding. This sub-subsystem comprises the fund resource, its attributes such as levels of availability and sustainability, and processes and activities dealing with funds allocation and distribution.

Strategic planning. This sub-subsystem consists of the existence of e-government strategic planning at various levels, the involvement of stakeholders during its establishment, and its implementation.

Vision. This sub-subsystem covers vision success factor as resources. To attain e-government success, e-government system leaders should have a clear, achievable, shareable and realistic vision that will be used as guidance by all e-government system stakeholders.

*Internal ICT infrastructure***.** This sub-subsystem is formed by internal ICT infrastructure resources which consist of hardware, software, database system, computer network and communication systems. As success factors, they should be highly available, sufficient, reliable and sustainable. This depends, among other things, on analysis and design, acquisition, distribution, construction, maintenance, and replacement.

*IS development and maintenance***.** This sub-subsystem is specifically associated with complex activities consisting of IS planning, analysis, design, and construction within a set-up time period, a specific set of objectives and purposes, and a specified amount of allocated budget. It also involves activities to ensure and realise sustainable IS.

IS staff. This sub-subsystem consists of human resources who have specialised skills and knowledge in ICT or IS. As a success factor, full-time IS staff should be competent, available, motivated and productive, which can be affected by the hiring process, learning and training activities, job allocation and organisational rewards.

IS organisation. This sub-subsystem covers the organisational units that manage IS resource and activities. In addition to the basic management functions of IS resources, coordination and collaboration with other units to establish e-government strategic planning and supporting system users are examples of other essential functions.

Organisation. The organisation sub-subsystem consists of the government's unit or organisation resource that uses the IS to deliver government services. Organisational culture, structure and diversity, as well as coordination and collaboration among government units are important success factors. Another important success factor is organisational change management.

Human resources. In this case, the sub-subsystem covers the human resources who use and in practice operate the application system of the e-government system to deliver service to citizens or business. Willingness to reduce their authority when providing government services to citizens and eagerness to learn new technology as a new and better way to deliver government services are examples of the support they can provide to the e-government system.

Marketing and education. This sub-subsystem consists of marketing the existence of an e-government system to citizens and educating them on its benefits and the advantages of using it. This is required to gain full support from them.

Cooperation arrangement. This sub-subsystem covers success factors related to cooperation between government and private companies. Resources and expertise available in private companies could help the government to attain e-government success.

General ICT infrastructure. This sub-subsystem includes success factors associated with the national electronic communication and electricity infrastructure, and the ever-changing nature of IT resources available in the market. This will affect the ability of citizens to access e-government system services.

Socio-economic conditions. This sub-subsystem consists of potential economic levels and population of the region, which can exercise “pushing and pulling” forces on e-government growth.

Other factors. This sub-subsystem covers a variety of other success factors such as the “me too” attitude among government leaders in deciding on e-government adoption, which can cause the success or failure of an e-government system initiative.

As indicated, the aforementioned description of the eGSF system decomposition does not provide information about what the eGSFs are and how they relate to each other at an operational or low level within and between sub-subsystems. In the next sections, this level of the success factors and their relationships will be explored in depth within SM and IS staff subsystems.

5.2.3 Subsystems selection

The eGSF system decomposition constructed in the previous subsections has illustrated the subsystems and sub-subsystems of the eGSFs. However, this decomposition has not yet indicated explicitly where the SM and IS staff subsystems (sub-subsystems) reside. Although justifications for focusing the study on SM and IS staff subsystems have been provided in Chapter 3, this subsection will describe a process of moving from a system consisting of subsystem success factors into the detailed or elementary success factors within the SM and IS staff subsystems. This process puts these two subsystems and their elementary success factors within the broader context of the eGSF system decomposition.

As can be seen in the description of the eGSF system decomposition, the system does not differentiate between whether the success factor is a resource or a process. For example, quality ICT infrastructure and competent human resources are resources, while e-government service delivery and IS development are processes. The resources and processes within the eGSFs system will be identified and presented using an Entity Relationship (ER) diagram – a database modelling tool commonly used during system analysis and design of IS development. Figure 5.3 gives an example of an ER diagram which consists of two tables or entities, Resources and Processes, and many-to-many relationships between the two tables. The first table consists of resources, for example, ICT Infrastructure, Funds and Human Resources, which can be detailed into specific resources. For instance, Human Resources can be detailed into General Human Resources and IS Human Resources. Further, the latter resource can be broken down into resources associated with it – for instance, Knowledge Resource and Motivation. Using the same reasoning, the second table comprises processes such as e-Government Service Delivery and IS Development and Management; and the latter can be further detailed into IS Construction and IS Maintenance. This many-to-many relationship between these two tables means one specific resource in the Resources table is associated with 0, 1 or more members within the Processes table and vice versa. For example, IS Human Resources is needed for e-Government Service Delivery and IS Development and Management, and ICT Infrastructure needs IS construction and IS Maintenance. On the other hand, IS maintenance needs Knowledge and Motivation of IS Human Resources, and e-Government Service Delivery requires ICT Infrastructure and Human Resources.

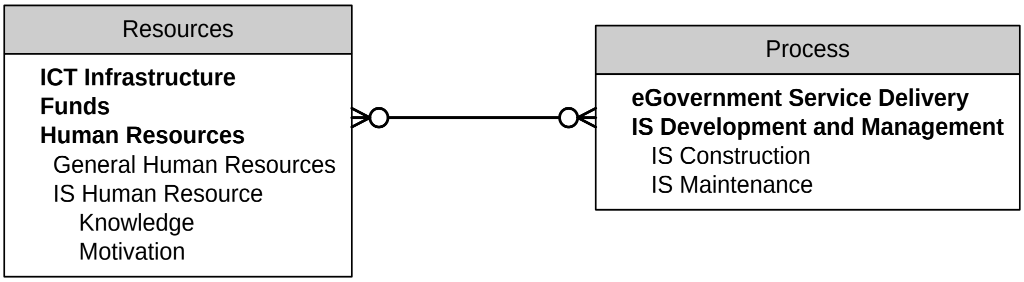


Figure 5.3 *An example of many-to-many relationships between Resources and Processes*

By differentiating resources and processes within the eGSFs system and using an ER diagram along with many-to-many relationships as shown in the aforementioned examples, the resources and processes will be explored and identified by referring to the representation of the eGSFs system decomposition. They are listed in the Resources and Processes tables accordingly, although certainly the lists are not exhaustive. Based on this description, the elementary success factors of SM and IS staff subsystems can be viewed from the whole context of the eGSF system.

Figure 5.4 shows a number of resources and processes within the context of the eGSFs system. The shaded areas in the tables indicate the subsystems within which more detailed success factors and relationships will be pursued using the SD modelling method. As is clear from the figure, the scope of the success factors being modelled with this method is e-government SM and IS staff management.

The reasons for focusing on these subsystems have been provided in Chapter 2 and Chapter 3. In short, this selection is based on the fact that e-government sustainability, among other reasons, is highly dependent on e-government SM, while SM performance is influenced by IS staff's motivation and competence as well as IS staff management.

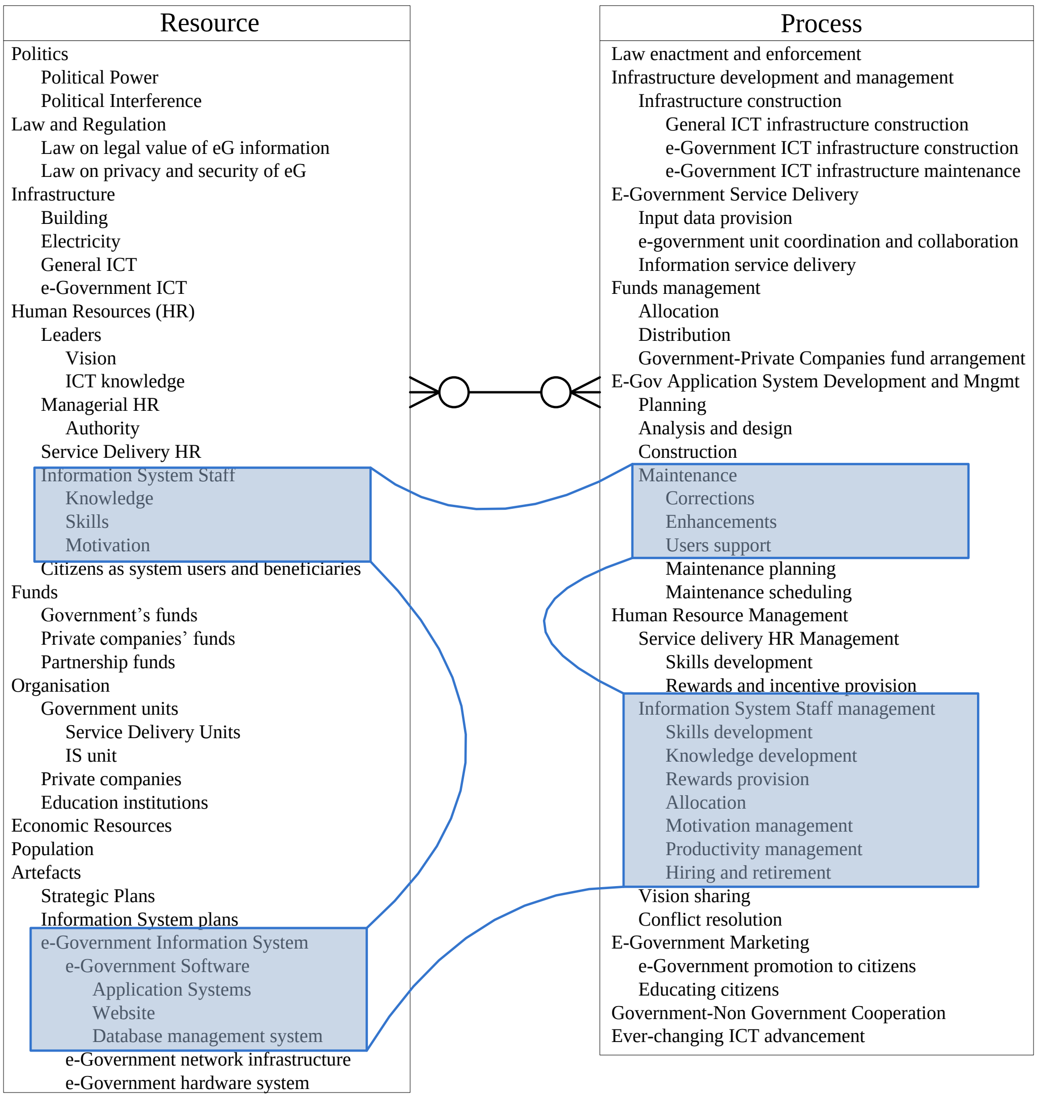


Figure 5.4 *A visual representation of the selected domains within the overall eGSF system*

5.3 The e-government success factors system dynamics

5.3.1 Introduction

This section describes the development of the formal model of the eGSFs relationships within SM and IS staff subsystems using the SD modelling method, at the same time being aware of the existence of various other eGSFs and relationships that can affect e-government success, as depicted by the eGSFs system decomposition.

Referring to the research design in Chapter 4, the model constructed in this chapter is the eGSFs SD model version 1.1. During the construction of the simulation model some desk validation also takes place, such as a dimensional consistency test.

5.3.2 Problem articulation

Model purpose. The purpose of the eGSF SD model is to give insight into the importance of the complex feedback relationships of the eGSFs in influencing e-government success, and to show why and how these relationships lead to e-government success over time. In particular, the model can assist e-government decision-makers or managers to ensure e-government software sustainability from the perspective of SM and the IS staff management. The model can also help e-government stakeholders, especially those from e-government units that still experience failure, to learn how to achieve e-government success within the scope of these two subsystems.

Problem formulation for SD modelling. Referring to the e-government success working definition and based on the selected eGSF subsystems i.e. SM and IS staff management subsystems, this study operationalises e-government success as e-government software sustainability. The study further defines software sustainability as the availability of the software to perform its intended purpose for a specified period of time after it is developed and implemented. To be available, the required software must exist, the software's features to perform the intended functions or to be used to deliver a (set of) particular service(s) must also exist, and the existing features must be able to be used to deliver its intended purposes as required. Because this study focuses on maintenance, how to create the software will not be investigated.

Software engineering literature has indicated that persistent maintenance in order to preserve the required software availability level is necessary. Several reasons exist for this persistent requirement: error-free software is non-existent, ever-changing business rules, regulations and organisation requirements, IT innovations, new users' requirements, and even the maintenance itself. If the organisation is unable to perform the required SM the functionality of the software will degrade, which means the ability of the software to deliver a particular level of services also degrades. This degradation will eventually lead to abandonment of the software much earlier than its intended lifetime, which in turn jeopardises the service delivery of the e-government system. In the worst situation, this problem can cause e-government system users to leave the system and hence lead to e-government system failure. The pace of software performance degradation depends on SM performance.

Software maintenance performance, that is, the ability of the organisation to cope successfully with the software performance degradation, depends mainly on the performance of the IS staff. This performance in turn depends on the willingness of IS staff to exert the necessary level of effort. There are many factors that can motivate IS staff to maintain high-level performance or improve to a high-level of performance, such as received organisational rewards, competence, jobs characteristic and the maintenance environment. The expectancy theory has shown that there is a dynamic feedback relationship between staff's effort, performance and rewards.

Information systems staff competence largely determines their maintenance performance. However, relative to ever-advancing IT, the level of competence tends to become obsolete over time and therefore a continual update of competence is essential. This can be achieved through effective and periodic training as well as effective jobs allocation. Effective job assignment can also improve the competence level through effective learning by doing.

Organisational rewards can also significantly influence the degree of IS staff turnover as well as absence level, because IS staff will judge whether or not the received rewards are in balance with their effort and performance. Both turnover and absence level directly affect the total organisational effort exerted for SM. However, it should also be noted that the provision of organisational rewards certainly depends on staff performance.

Reference mode. Referring to the problem formulation for the SD modelling described above, the reference mode of e-government success is defined by a high-level of e-government software availability over time. Focusing on the steady-state performance, the availability is measured by the ratio between the number of successful maintenance deliveries and the number of maintenance requests at a particular time.

This availability depends on the software errors occurring during service delivery, new requirements for software enhancement to maintain or improve service delivery, and SM performance. The ratio is formulated by assuming that all critical errors and enhancement needs materialise as maintenance requests and are always given the highest maintenance priority. By using this measure, e-government success is indicated by steady-state ratio values close to 1 over time.

A two-dimensional diagram in which the ratio and time are represented as the vertical and horizontal dimensions respectively is drawn to represent the reference mode. The diagram is depicted in Figure 5.5. This figure also describes the availability level of unsustainable software.

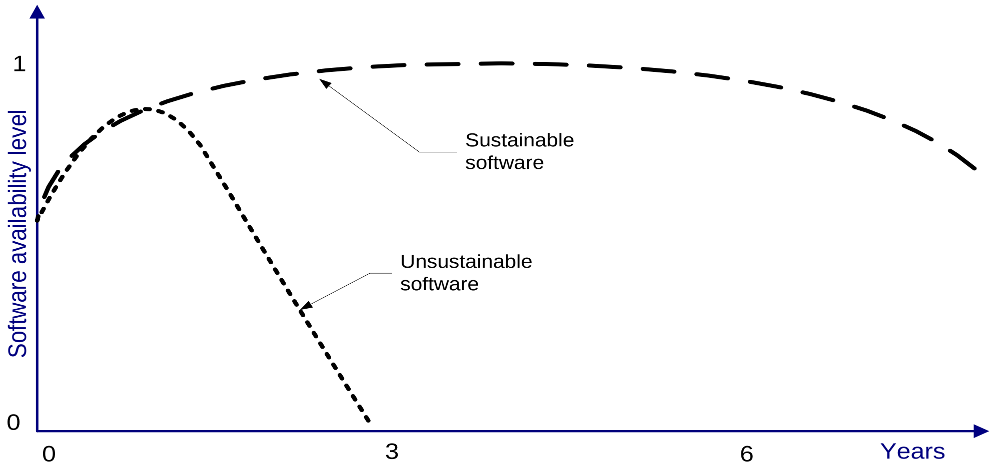


***Figure 5.5** Behaviour over time of steady-state software availability*

The years that label the horizontal dimension are for illustrative purposes, showing that unsustainable software can only be used to deliver e-government services over a relatively short time, while sustainable software can operate consistently to deliver services for a long period, at least as long as its designed lifetime. It is assumed that actual new functionality requirements continue to arise and need to be addressed. However, at a particular point of time the cumulative level of implemented maintenance, from many perspectives, is such that it is no longer feasible to maintain the software but rather it must be replaced.

Time horizon. Several previous studies, for example Krishna and Walsham (2005), defined e-government sustainability as if it can operate and grow for at least four years, while Burch and Kung (1997) showed that software replacement is taking place after five years of the software in operation. Based on this data from previous studies, the time horizon is set to five years. This setting also assumes that the normal length of time of government administrators hold their administrative power in developing countries is five years. Beyond this length of time, the problem of sustainability is more likely to depend on political than IS related factors, in the sense that the e-government software is unsustained because the political leader ceases its operation.

Key variables. Related to the problem formulation of SD modelling, several key or main factors can be inferred and are listed in Table 5.1.

Table 5.1 *Main factors of SD modelling*

Endogenous	Exogenous
Software availability level	Software quality resulting from a software development project.
SM performance	Environmental pressure for enhancements
SM requests	Ever-advancing IT
IS staff effort and competence	
Training and learning by doing	
Rewards	
IS staff performance	
IS staff turnover and absence level	
SM characteristics and environment	

5.3.3 Qualitative representation of eGSF relationships

Figure 5.6 presents the CLD of the key success factors of the eGSF system, conceptualising their complex feedback relationships. This is of version 1.0. The development of this conceptual CLD is based on the problem formulation presented above and the literature review described in Chapter 3, sections 3.3 and 3.4. Therefore, this CLD proposes how the eGSFs relate to each other in a qualitative manner that eventually results in e-government success, and how the success influences back to the success factors.

To facilitate the presentation describing the relationships, the logic behind the construction of this diagram will be explained based on its sub-diagrams. The way to understand the diagram refers to the technical aspects presented in Chapter 4, section 4.4.4.

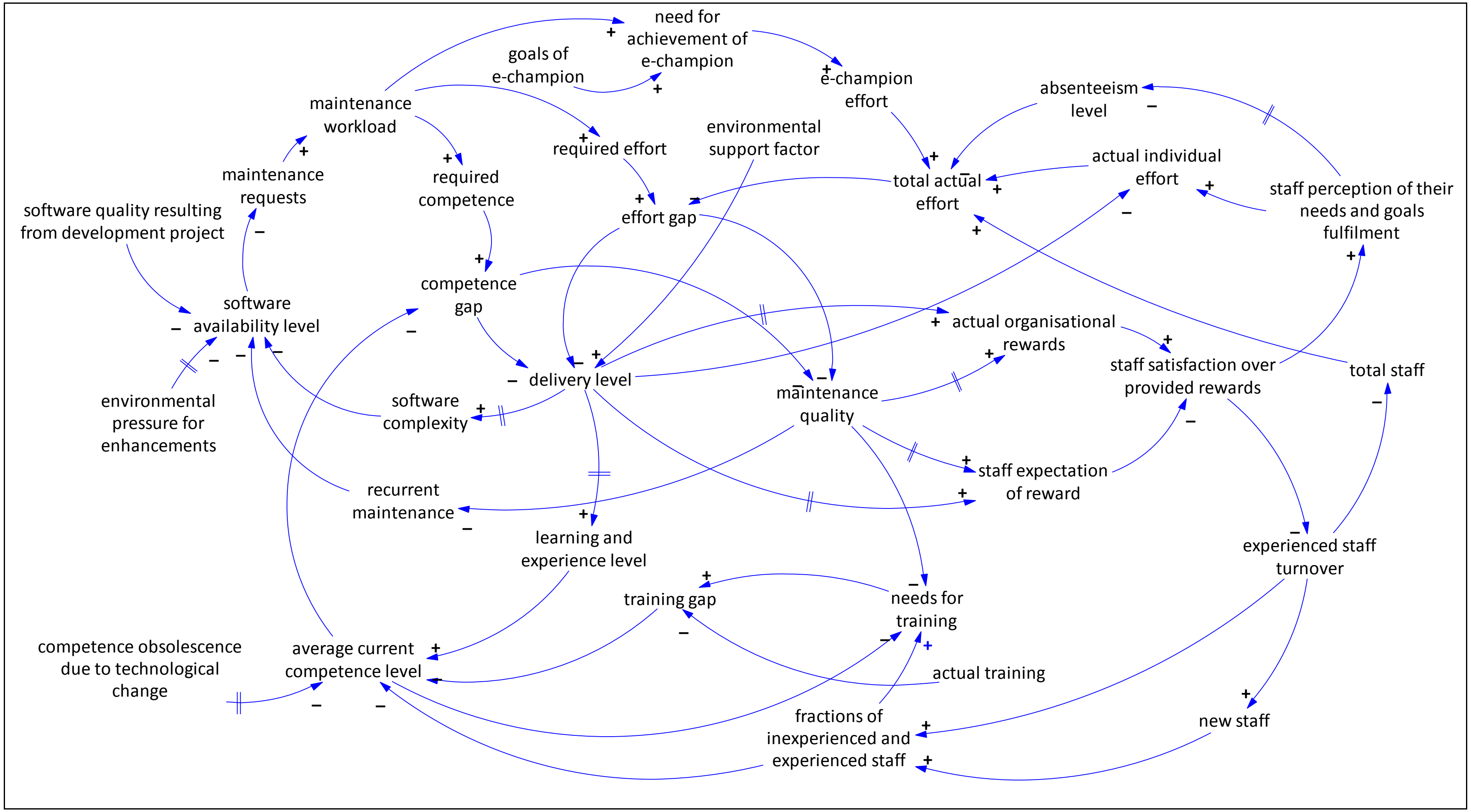


Figure 5.6 *The initial causal loop diagram of the eGSF system*

Maintenance process. Figure 5.7 presents a subset of Figure 5.6 relating to the e-government maintenance process.

In essence, the level of software availability is influenced by three factors. First, it is positively influenced by software quality resulting from the development project. Software with a lack of quality produces more errors and does not satisfy its designed requirements; therefore, it is less available to deliver its intended services. Second, it is also affected by environmental pressure for enhancements but in a negative direction. If there are many changes in the organisation which affect the delivery of services, many new requirements by users' and changes in IT, then there are more requirements that need to be accommodated by the software, which in turn reduces its availability level. Third, the level of software availability is also negatively influenced by the maintenance itself, through the increasing software complexity and recurrent maintenance requests. In turn, the software availability level will negatively influence the MRs.

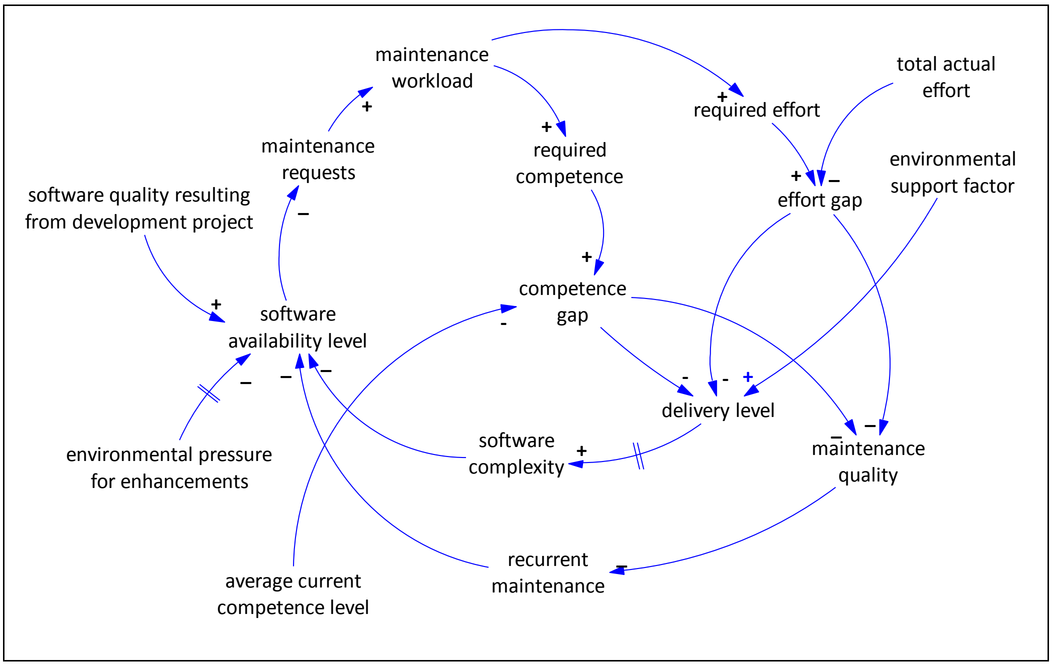


Figure 5.7 *Causal loop relationships between success factors of the maintenance process*

Any e-government SM is always initiated by a maintenance request, which can originate from users, either internal or external, or from the IS staff itself. The maintenance requests occur randomly and a large number of maintenance requests means large workloads of maintenance. The workloads depend on both the degree of difficulty of a maintenance problem and the number of workloads at a particular time unit.

Therefore, solving any kind of maintenance problem requires two things simultaneously: staff effort and competence. However, as the required effort and competence cannot always be fulfilled, then the differences between the required and actual cause gaps between these two pairs. If the average current competence level of IS staff is high, then the competence gap will be narrow and if the total actual effort exerted by IS staff is high the effort gap is also small. These gaps affect completed maintenance delivery level and quality. The delivery level will also be positively affected by smoothness of communication with the environment (other competent colleagues and users) to obtain their support. The higher the gaps, the lower the delivery level and the maintenance quality.

The level of delivery then has a positive influence on the level of software complexity, which itself accumulates over time. However, we have to be cautious in interpreting this relationship, because a lower level of delivery does not mean a lower level of software complexity. It is as in the case of the relationship between birth rate and population number: a reduction in birth rate does not mean a reduction in population number. Any maintenance delivered always adds to software complexity irrespective of whether the delivery level is high or low. Thus the relationship means that an increase or decrease in delivery level increases or decreases the rate of change of software complexity.

On the other hand, the quality level of maintenance has a negative effect on recurrent maintenance. This recurrent maintenance takes place when a fault caused by delivered maintenance becomes evident at some point of time in the future, or when the requesters repeat their request due to cancellation, incomplete delivery or rescheduling.

In turn, software complexity and recurrent maintenance negatively affect the software availability level. It has been indicated by SM literature that more complex software, especially that with long SLOC, is more prone to error than less complex software; therefore, this results in the reduction of software availability. In terms of recurrent maintenance, the more recurrent maintenance occurs the lower the software availability.

Here, the software availability level, as a result of implemented SM, also measures SM performance.

Motivational factors. Figure 5.8 is another subset of Figure 5.6 which presents feedback causal relationships of the motivational factors. The CLD in Figure 5.8 is a representation and implementation of the expectancy theory and other related factors in an e-government SM context.

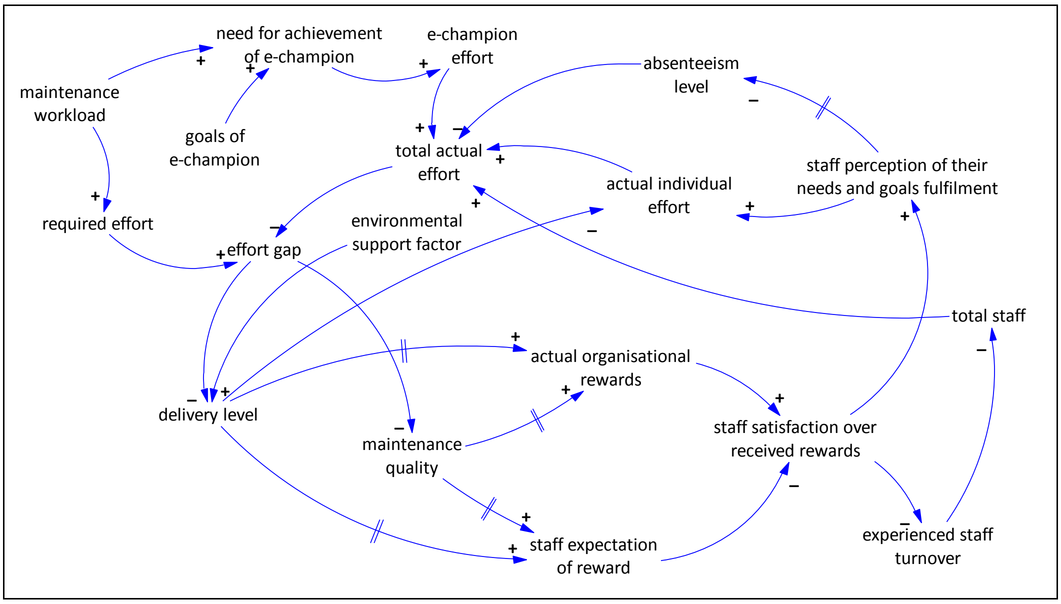


Figure 5.8 *Causal loop relationships of the motivational success factors*

In this case, the effort factor of the expectancy theory is represented by actual individual effort, total actual effort and the effort gap. Individual effort is defined as the amount of time (hours) for a period of time (such as one day) spent by IS staff to work on an assigned maintenance task. Total effort represents the organisational effort in terms of its staff spent in performing maintenance tasks for a specified period of time. It is a multiplicative function of individual effort and total staff. As the actual individual effort increases, so the total actual effort also increases, assuming the total staff is constant, which in turn reduces the effort gap. It is also assumed that staff only perform maintenance tasks.

The delivery level and maintenance quality, which represent maintenance performance, positively influences both actual organisational rewards and staff expectation of rewards. Any discrepancy between these two will affect staff satisfaction over the rewards provided. As underlined by the expectancy theory, the value of the actual rewards depends on how the IS staff perceive their value. In turn, this satisfaction will positively affect staff perception of their needs and goal fulfilment and negatively influence experienced staff turnover. A high level of staff perception of their needs and goal fulfilment improves actual individual effort and reduces the absenteeism level, which in turn increases the total actual effort. On the contrary, the staff may be present in the workplace but not exert their full effort in maintaining the software as a result of their perception that their effort does not give rise to the fulfilment of their needs and goals. An increase in the absenteeism level may be assumed to be the staff's attempt to fulfil their needs and goals from other places, as has occurred in many developing countries.

In addition, much research has shown that unsatisfied and experienced staff tend to leave the organisation. This is so because they can see opportunities to obtain more rewards, considering their current level of experience and competence. In other words, if staff satisfaction over received rewards decreases, then experienced staff turnover will increase. This eventually negatively influences the total actual effort through the total staff factor.

The expectancy theory also mentions that staff understand that the rewards can only be obtained by performance. An observation made by staff towards their performance could lead to a change in their exerted effort level, which in turn influences the performance level. The staff tend to raise their effort – for example, by increasing the amount of working hours solely dedicated to maintenance – if their normal effort does not lead to their expected performance, and the other way around. These relationships are expressed as a balancing loop of delivery level → actual individual effort → total actual effort → effort gap → delivery level.

Some research has indicated that the e-champion plays an important role in e-government success. An e-champion is a staff member who has specific personal goals, the goals of an e-champion. In this case, the maintenance workload factor positively influences the need for achievement of an e-champion, which in turn also positively affects e-champion effort and eventually this positively influences the total actual effort.

Competence development. Figure 5.9 is another subset of Figure 5.6 which presents feedback causal relationships of IS staff competence-related factors. In this subset, the key success factor is the average current competence level.

The average current competence level expresses the average nominal amount of maintenance tasks that can be completed for a particular period of time by a staff member when there is no loss factor. In this situation, the staff is fully motivated, the supporting facilities to perform maintenance is perfect, etc. It is assumed that this level is associated with the level of formal education of the staff, in the sense that staff with higher levels of formal education are assumed to have a higher level of competence than those with a lower educational level.

The average current competence level is negatively influenced by an exogenous factor called competence obsolescence due to IT advancement, but the decrease in the competence level takes time. There is a reinforcing loop which links the average current competence level, competence gap, delivery level and learning and experience levels. To a degree, any completed maintenance delivery level improves staff's learning and experience levels. Through time, as the exposure to various challenges of maintenance increases in the course of the increase of the delivery level, then the staff's experience also increases. The increase in staff experience causes the average current competence level to increase as well.

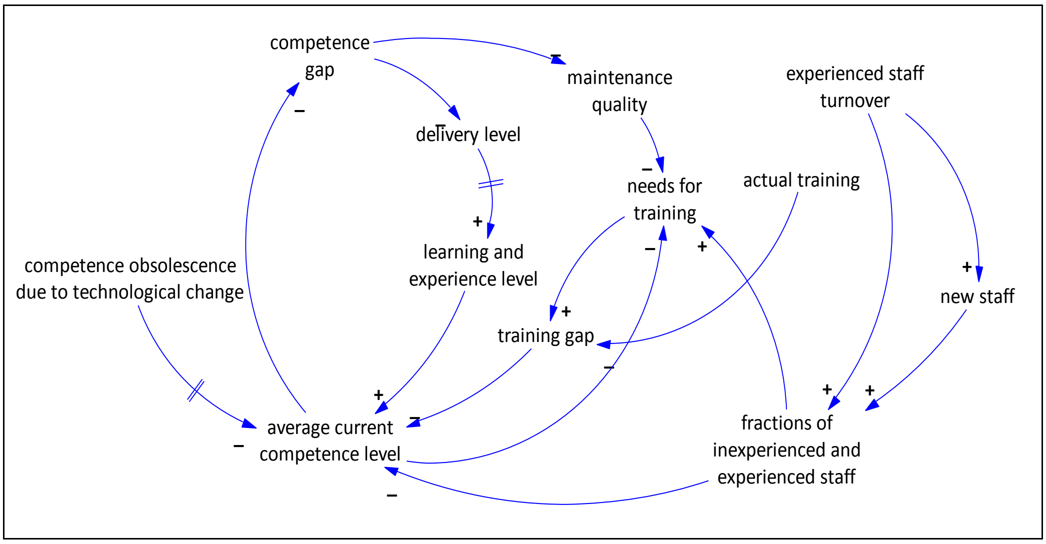


Figure 5.9 *Causal-loop relationships of IS staff competence-related success factors*

The average current competence level also relates to training, especially in response to the competence obsolescence due to IT advancement. The increase in the average current competence level, through narrowing the competence gap, improves maintenance quality which in turn reduces the needs for training or, at least, the level of training needed. The need for training and the actual training, which responds to the need for training, will determine the training gap which represents the value or effectiveness of training to the improvement of the average competence level. Assuming the actual training and the competence obsolescence factors are at a constant level, a higher need for training will widen the training gap which indicates a decrease in the average current competence level. On the other hand, the need for training is negatively influenced by the average current competence level.

The fractions of inexperienced and experienced staff negatively influence the average current competence level and positively affect the need for training. Experienced staff turnover and new staff determine the fractions.

5.3.4 Simulation model of eGSFs relationships – Introduction

The previous section has presented the development and explanation of the conceptual CLD of the eGSFs based on the existing research. This diagram has shown how, conceptually, the success factors relate to each other in influencing e-government success in a qualitative way. Therefore, the diagram at this stage of research attempts to provide a perspective on why and how the eGSFs affect e-government success.

In the following section, the SD simulation model will be developed. The model is an initial and conceptual representation of the eGSFs and their relationships in a quantitative way. The development will be based on the understanding of the eGSFs CLD and the literature review presented in Chapter 3, section 3.3 and 3.4. In the following model development, the success factors and the relationships that have not been described explicitly in the CLD will be formally constructed, such as the range of the level of a factor and the form of the relationships. However, it must be noted that it is not possible to achieve one-to-one correspondence between the factors in the CLD and those in the simulation model, because the model provides more explicit detail of the factors and their relationships.

During the model construction, constants, initial values, graphical relationships and mathematical equations are introduced which are used to represent structural relationships among factors. Constants, initial values and forms of graphical relationships are based on previous relevant empirical studies wherever possible; however, plausibly estimated values and graphical forms are used for those elements for which the empirical data is hard to obtain. The simulation time unit is set to one quarter and the simulation time length is set to five years or 20 quarters.

During the construction, the structure and parameters of the model are cross-checked with the existing literature; the introduction of parameter values and relationships forms is evaluated through the examination of the dynamic behaviour of the model at that particular stage of model construction, and the dimension consistency of each of equation being formulated is also checked. This checking is part of the desk validation.

Referring to the conceptual CLD, the model is organised into five sectors: maintenance process; motivational factors; staff competence; staff development and staff allocation. The sectors are created such that each sector contains relatively coherent factors. Arrows shown in the figure indicate the relationships of between sectors. These sectors are depicted in Figure 5.10 which is obtained from STELLA® software.

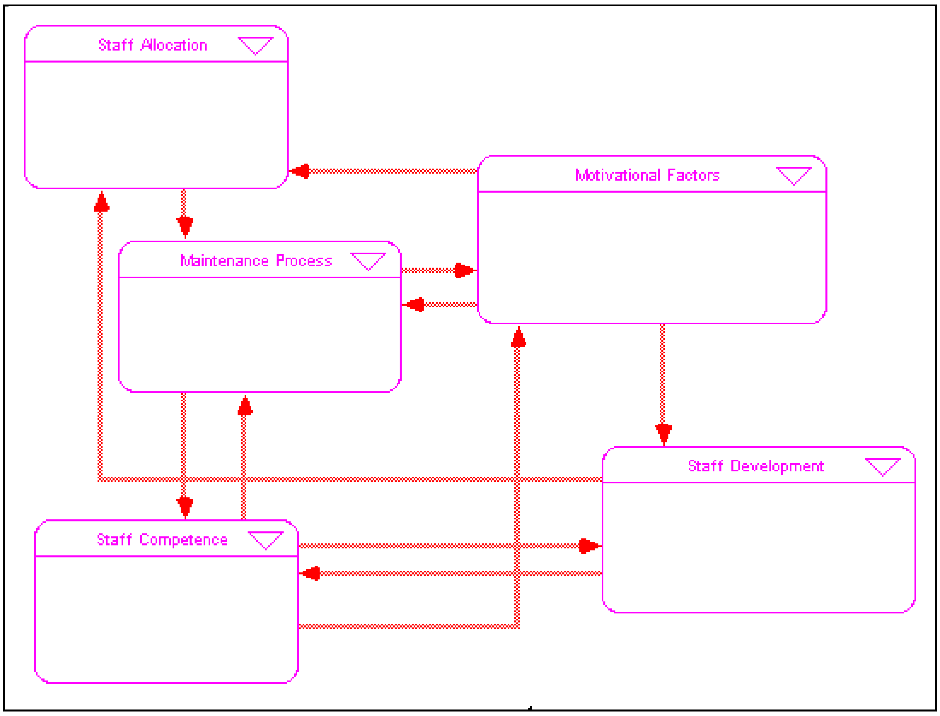


Figure 5.10 *The simulation model sectors of the eGSF relationships*

The model development and presentation follow this five-sector sequence. The visual representation of the model will be given first, followed by a description of the construction of the model elements.

5.3.5 Simulation model of eGSFs relationships – Maintenance process

Figure 5.11 presents the initial and conceptual model of the maintenance process sector. The maintenance process begins when a new maintenance request is accepted by an IS management unit. The maintenance requests (MRs), which occur randomly over time, originate from exogenous and endogenous sources. Any accepted MRs will be classified, prioritised and assigned to a staff member or team of staff. The classification of MRs also means they will undergo different maintenance processes: corrective maintenance and enhancive maintenance. At the end of the process, successfully delivered maintenance certainly will increase system size or complexity and may generate further requests. The proportion of successfully completed maintenance tasks relative to the MRs, named the *Delivery_and_request_ratio*, measures the SM performance and software availability level.

Because this study deals with government units that develop their own software to deliver their specific services, it focuses only on closed-developed software.

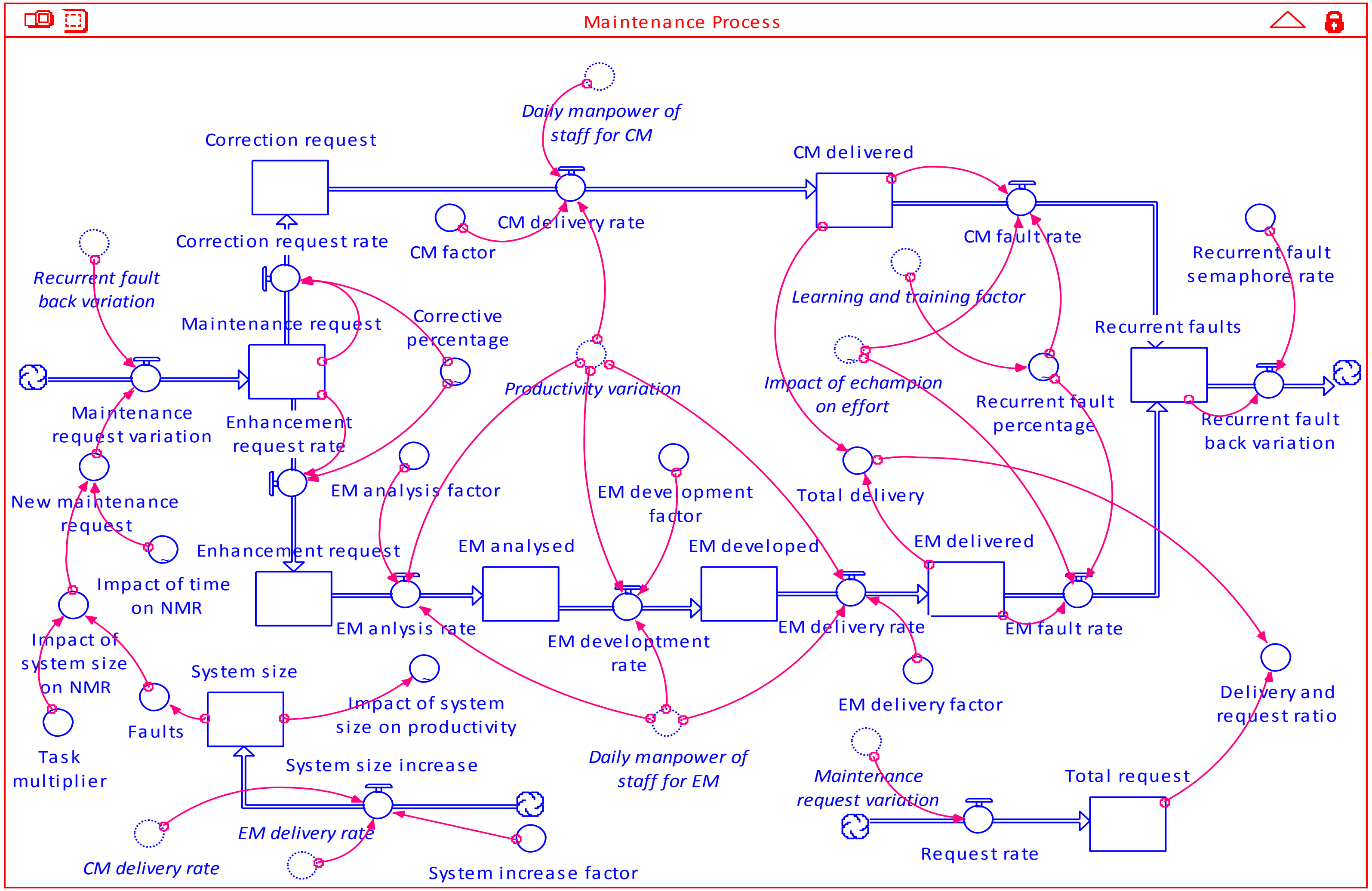


Figure 5.11 *The maintenance process sector of the model*

Incoming MRs. Referring to Figure 5.11, the dynamic number of MRs is symbolised by a stock named *Maintenance_request*. The level of this stock is determined by the incoming flow of the requests and the outgoing flow of the request classification. This stock also indicates the number of maintenance tasks, or workload. Accordingly, the unit for this stock is *tasks.* The rate of the incoming flow is represented by *Maintenance_request_variation* which is formulated as:

Maintenance_request_variation = New_maintenance_request + Recurrent_fault_back_variation.

The *New_maintenance_request* factor of this equation is formulated as:

*New_maintenance_request = Impact_of_system_size_on_NMR * Impact_of_time_on_NMR,*

where the first factor represents the impact of system size on new maintenance requests (NMRs) and the second represents the impact of time on NMRs. On the other hand, the *Recurrent_fault_back_variation* is the variation of incoming MRs classified as recurrent maintenance due to the ripple effect, incomplete maintenance delivery, or cancelled or postponed maintenance. The unit for the *Maintenance_request_variation* is *tasks/quarter.*

Software size or complexity. As shown by much literature reviewed in Chapter 3, software size or complexity influences software faults that can turn into MRs if reported. Much of the existing research used the SLOC to represent software size or complexity to predict software faults for its simplicity and concluded that the larger the software module, the higher the number of faults that may be contained in that module. Among various models that use the SLOC to predict the number of faults, this study uses a highly simplified negative binomial model:

*the number of faults = exp(-1.636 + 1.047*log(SLOC))*

to describe the relationship between the SLOC and the number of faults. Graphical representation of this formulation is given in Figure 5.12.

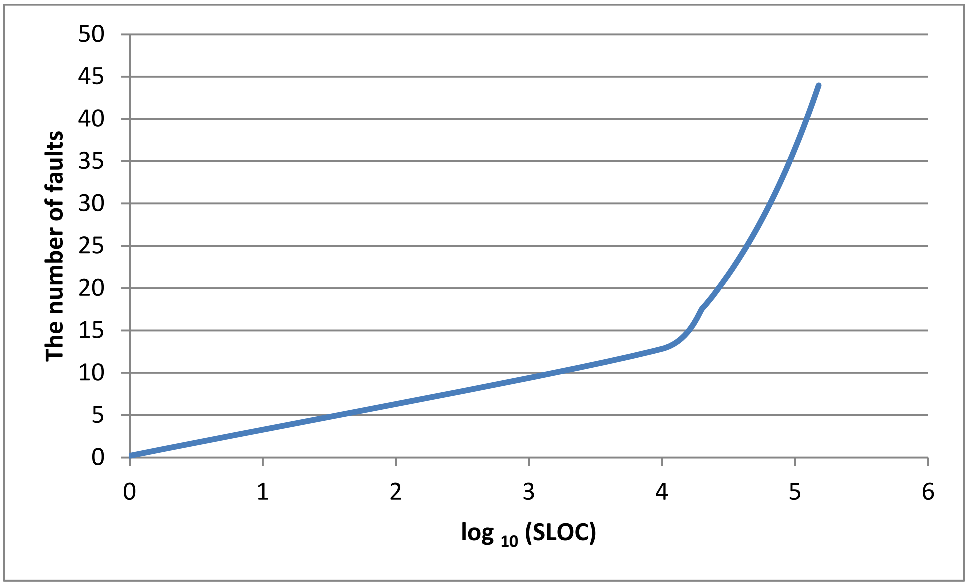


Figure 5.12 *The impact of the system size on NMRs*

Although Figure 5.12 is built on the idea of a relationship between system size and the number of faults, it is assumed that all faults turn into MRs. Therefore it represents the impact of the system size on the number of MRs. In addition to representing the faults, it is assumed that other type of MRs also follow this relationship. Based on this assumption, the equation represents the impact of the system size or complexity on the number of MRs which leads to NMRs or *Impact_of_system_size_on_NMR*. It can be noted that the relationship between system size and NMRs is not linear. The unit for this converter is *tasks/quarter.*

Further, by referring to Figure 5.11, this equation is formulated as:

Impact_of_system_size_on_NMR = (*Task_multiplier* * *Faults*).

The units for these are *tasks/quarter* and *unitless,* respectively. The *Task_multiplier* represents that all faults turn into MRs, while *Faults* represents the faults produced by the software as a result of its degree of complexity. The software size or complexity is symbolised by a *System_size* stock which is assumed to have initial value of 650,000 SLOC. The unit for this stock is *SLOC.* The size or complexity of the software increases as a result of completed maintenance. The rate of increase is modelled by:

System_size_increase=(CM_delivery_rate+EM_delivery_rate) System_increase_factor.*

The average additional SLOC of completed maintenance from Jorgensen & Sjoberg (2002), that is, 206 SLOC, is used as input for the *System_increase_factor.* The unit for *System_size_increase* is *SLOC/quarter* and for *System_increase_factor* is *SLOC/tasks.*

Time-series behaviour of MRs. The *Impact_of_time_on_NMR* represents the time-series behaviour and randomness of MRs. This effect of time representation is to accommodate the MRs caused by quality of the software resulting from a development project and the environmental pressure for the software enhancement. The form of the relationship between time and NMRs is depicted in Figure 5.13.

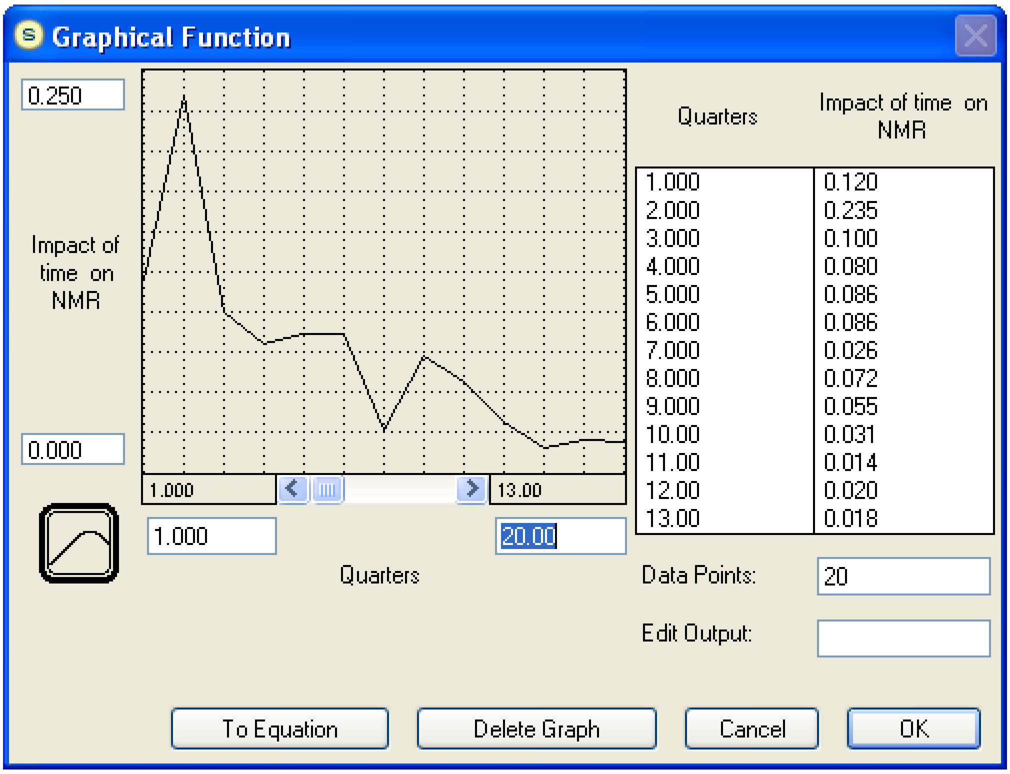


Figure 5.13 *The relationship between time and NMRs*

To formulate this relationship, this study utilises the time-series pattern of MRs from a specially developed software application for a particular organisation, which was not open for public participation. Especially, the longitudinal pattern of monthly total maintenance requests (corrective, enhancement and user-support) reported by Burch and Kung (1997) will be used. The graph in Figure 5.13 is reproduced from monthly total request number for the first 60-month period out of a 67-month period (the original) of Burch and Kung's graph. It is represented as 20 quarterly percentages of the total requests. The value for each quarter is a percentage and is calculated from the total requests for each quarter divided by total MRs. It is assumed that the two types of requests (CM and EM) inherit the same longitudinal pattern from the monthly total maintenance request. The unit of *Impact_of_time_on_NMR* is *unitless*.

Classification of MRs. Previous empirical research has shown that the percentages of CM and EM are significantly different. Therefore, this study adopts this difference and represents the percentage of each maintenance classification into *Corrective_percentage*. Thus, in this case the percentage of the maintenance classified as corrective is determined by *Corrective_percentage,* while those classified as enhancive are calculated from (*1 – Corrective_percentage*). These settings also indicate that corrective MRs are given higher priority than enhancive ones. This study only considers CM and EM, while user support activities will be treated as other duty. The unit for this converter is *1/quarter*.

This study will use the percentage of maintenance internally-originated requests reported by Burch and Kung (1997). There is no specific reason for this selection except that they provided the actual time-series data of MR and the data is needed for the simulation instantiation. Further, the percentages of corrective MRs and enhancive MRs also vary over time following four stages of about nine months each (Burch & Kung 1997). In the first stage, both corrective MRs and enhancive MRs have low percentages, which are then followed by a high corrective request percentage in the second stage. In the third stage, the percentage of request is dominated by EM, while in the last stage both are about equal. Therefore, the *Corrective_percentage* variable is modelled as a line graph that varies over time, as is shown in Figure 5.14. Burch and Kung did not provide the explicit percentage of corrective MRs after month 36; so it is assumed that the corrective MRs percentage is lower than the enhancive MRs percentage and keeps decreasing towards zero.

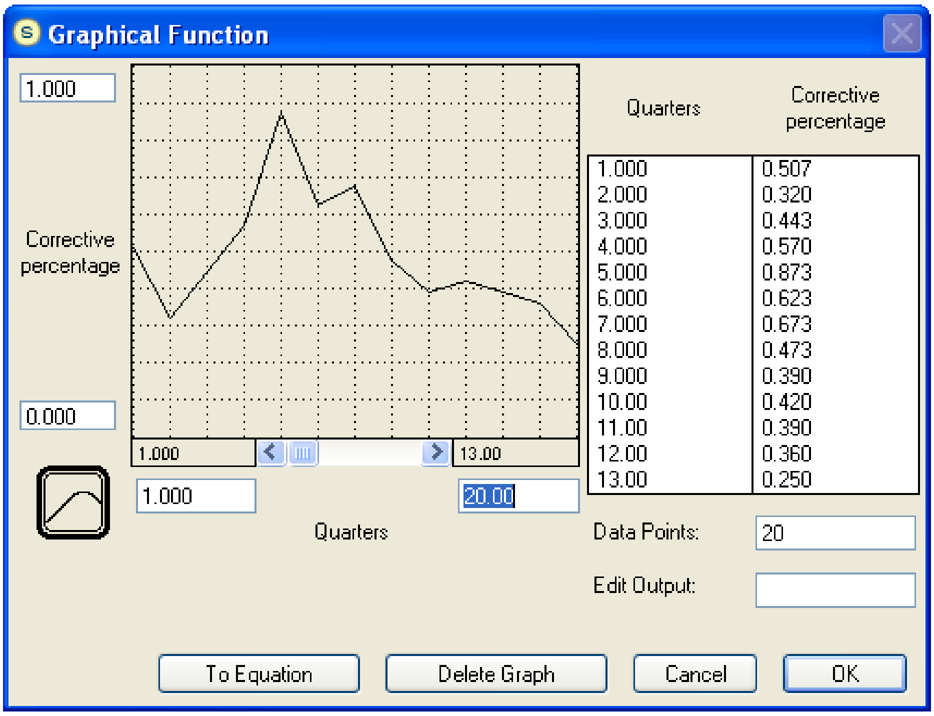


Figure 5.14 *The quarterly percentage of the corrective MRs*

A flow and cumulative number of the corrective MRs are, respectively, represented by *Correction_request_rate* and *Correction_request,* while those of the enhancive MRs are modelled by *Enhancive_request_rate* and *Enhancement _request*. The rates of the flow are functions of the *Corrective_percentage* and the *Maintenance_request*. The presentation for these is depicted in the left-hand side of the maintenance process model shown in Figure 5.11. This representation also indicates that classification of the requests as CM or EM and assignment to the staff take a certain amount of time. By introducing this part of the model, it is assumed that any MR will undergo a classification and assignment process prior to the maintenance work.

Corrective maintenance phase. Once an MR is classified as a corrective one, it will undergo a CM process for a period of time and results in a CM delivered to users. Each of the requests becomes a task assigned to a CM team. However, as observed by previous research, this team can consist of only a small number of staff, perhaps only one. Also, the completion time is significantly shorter than for the average of the overall maintenance tasks. Accordingly, the CM phase is set as a one-phase process. Each CM request is considered as one *task.*

By adopting the software development rate model in Abdel-Hamid and Madnick (1991**:** 78), the completion and delivery for CM is formulated as:

*CM_delivery_rate=Productivity_variation * Daily_manpower_of staff_for_CM * CM_factor*.

The first factor represents variation of staff average maintenance productivity over time. It is the number of *task*s that can be completed per person per quarter or *task/person-quarter*. The second factor represents actual average number of staff solely allocated for CM. This factor will be elaborated more in the Staff allocation sector. The third factor is an adjustment factor for the *Productivity_variation.* The adjustment is necessary because the variable used to represent the maintenance productivity is not a function of CM rate, but rather a function of the average maintenance rate; and these two rates have different values.

To determine the value of *CM_factor,* empirical SM data that indicated the average CM completion time is about one-third of the average maintenance completion time will be utilised. More specifically, the data showed that a CM task needs, on average, 1.7 person-days, or about 0.026 person-quarters, while overall maintenance needs, on average, 5 person-days, or about 0.077 person-quarters. The resulting figure from the comparison between these two values, which is about 3, is then used. The *CM_factor* is *unitless* and will be used to multiply the *Productivity_variation*. The use of this value means that the CM completion rate is higher than the average of the overall maintenance completion rate.

Enhancive maintenance phase. The maintenance process for an EM comprises three phases as described in Figure 3.2 of Chapter 3: enhancement analysis; enhancement development; and enhancement delivery.

As there are three phases, the EM is formulated by using three pairs of stocks and flows. The rate of flow for each of the enhancement phases is determined by the multiplication of the *Productivity_variation* and *Daily_manpower_of_staff _for_EM,* which is also adopted from Abdel-Hamid and Madnick (1991**:** 78) as the case of CM. Because the same value of the *Productivity_variation* is used for both CM and EM, it is necessary to multiply the rate of flows of both CM and EM with a factor to reflect that they have different rates. In addition, as the amounts of time taken for each enhancement phase are also different, this difference is accounted for by introducing three different multiplying factors. It has been shown by previous literature that more than half of staff time is allocated to the first phase, and only a relatively very short time is spent for the last phase. The multipliers for each phase are *EM_analysis_factor, EM_development_factor,* and *EM_delivery_factor*. All these three factors are *unitless.* The rate for each phase is formulated as:

EM_analysis_rate= Productivity_variation Daily_manpower_of_staff_for_EM * EM_analysis_factor,*

EM_development_rate= Productivity_variation Daily_manpower_of_staff_for_EM * EM_development_factor,*

EM_delivery_rate= Productivity_variation Daily_manpower_of_staff_for_EM * EM_delivery_factor.*

To determine the values of the multipliers, it is necessary to consider the fact emerging from previous studies that, on average, an overall EM task requires 7.5 person-days (Jorgensen 1995) or about 0.115 person-quarters. Accordingly, if there is one maintenance staff member who takes 7.5 days to carry out and deliver one EM task, then it can also be assumed that the staff take 4 days for one analysis, 2 days for one modification development and 1.5 days for one delivery of one EM task. Based on this assumption, one EM task takes about 0.062 person-quarters for the analysis plus about 0.031 person-quarters for the development plus about 0.023 person-quarters for the delivery. From these settings, the value of *EM_analysis_factor* is 1.25, *EM_development_factor* is 2.5 and *EM_delivery_factor* is 3.3. In addition to the assumption about the time required for each phase, it is assumed that the numbers of staff allocated for EM are distributed equally over the three phases, meaning that each phase will be allocated one-third of the EM staff.

The resulting analysis, development and delivery processes are accumulated in *EM_analysed, EM_developed* and *EM_delivered,* respectively. The unit for these three stocks is *tasks.* Unlike the level of *EM_delivered*, the levels of *EM_analysed* and *EM_developed* will never consistently increase as they will be drained by the modification development and the delivery rate.

Recurrent maintenance. The SM process itself can cause maintenance requests. There is a proportion of MRs that cannot be fulfilled for various reasons.

There are also some of the delivered maintenance that might not satisfy the requirements or trigger other software faults. All of these possibilities can be used to reflect the maintenance quality process. It must be noted that these possibilities do not always materialise into MRs and in general, maintenance ripple effects have not been able to be predicted with confidence.

Assuming that unfulfilled requests have been part of the *Impact_of_ time_on_NMR*, then recurrent maintenance consists only of the requests caused by lack of quality maintenance: incomplete delivery or faulty maintenance. The percentage of these types of requests can also be assumed to go down as the maintenance staff learn more lessons from the increasing number of the completed maintenance tasks and receive effective and relevant training. The recurrent maintenance is also considered to be influenced by the involvement of the e-champion.

Within the model, recurrent maintenance is represented by the *Recurrent_faults* stock, the level of which is determined by incoming flows: *CM_fault_rate* and *EM_fault_rate*, and by outgoing flow *Recurrent_fault _back_variation*. The unit of this stock is *tasks*. Each of these two rates is:

*CM_fault_rate = CM_delivered * Recurrent_fault_percentage * (2 – Impact_of_echampion_on_effort),*

*EM_fault_rate = EM_delivered * Recurrent_fault_percentage * (2 – Impact_of_echampion_on_effort)*.

The unit for both rates are *tasks/quarter*.

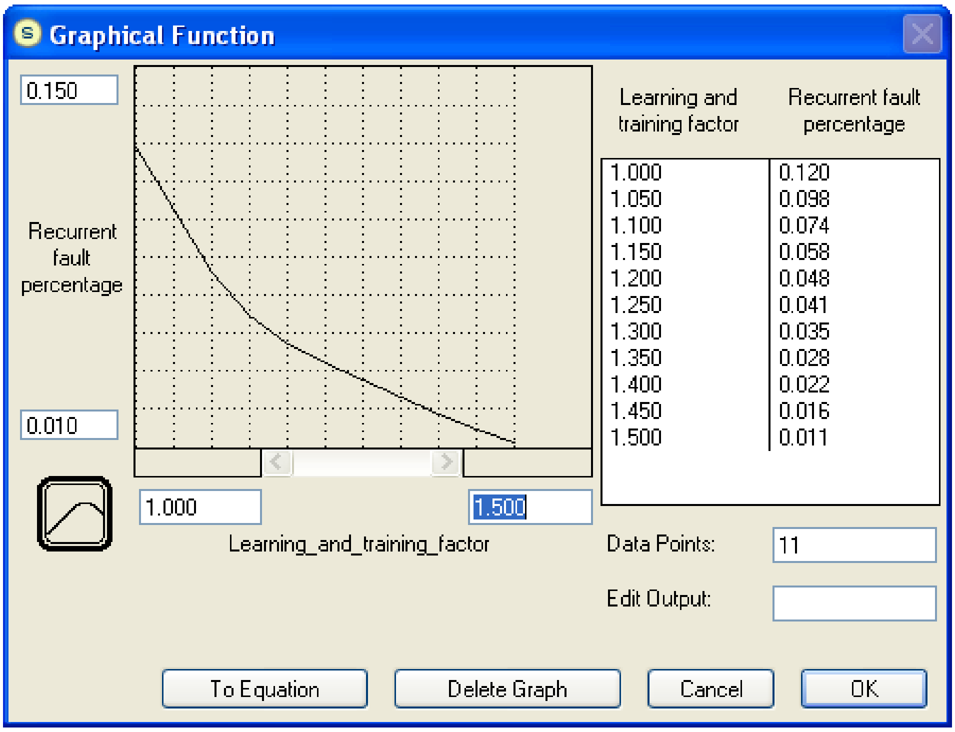


Figure 5.15 *The impact of learning and training on recurrent faults*

The *Recurrent_fault_percentage* itself is constructed as a monotonically decreasing function, describes a decreasing recurrent maintenance task and is determined by the *Learning_and_training_factor*, as depicted in Figure 5.15. To determine the initial value of the *Recurrent_fault_percentage*, it is assumed that CM and EM create the same level of recurrent maintenance. Using Jorgensen's data, the value is set to 12%. The graph of the *Recurrent_fault_percentage* assumes that when the *Learning_and_training factor* is 1, the percentage of recurrent faults is about 12% out of the total delivery. This is the case when the maintenance staff is new.

The *Recurrent_fault_back_variation* describes the rate of the recurrent faults transformed into MRs. The unit of this rate is *tasks/quarter.* Although the MRs caused by recurrent faults cannot be predicted in advance, nevertheless a value is required to represent this type of MR. For this, it is assumed that half of the recurrent faults materialise as a maintenance request per quarter. The value of this rate is determined by:

*Recurrent_fault_back_variation = Recurrent_faults * Recurrent_fault_semaphore_rate.*

The latter factor is set to 0.5 to reflect that half of the recurrent faults come to evidence in a quarter, the unit of which is *quarter*.

Software maintenance performance. As mentioned in the reference modes formulation in section 5.3.2, software availability depends on the level of successful maintenance delivery relative to the level of MRs. Successful maintenance delivery means that all necessary maintenance processes in each subsequent phase can be satisfied. The number of successful deliveries is the sum of the cumulative successful CM and EM deliveries at any time. From this, the level of maintenance performance can be formulated as the ratio between the number of maintenance delivery and request and denoted by *Delivery_and_request_ratio.*

Delivery_and_request_ratio = Total_delivery / Total_request
Total_delivery= CM_delivered + EM_delivered

However, it should be noted that as this ratio is calculated based on the value of the *Total_delivery* and the *Total_request* in the same quarter, then the value will be slightly different from the actual ratio value. This is so because a maintenance delivery in a particular quarter, say *t*, is a completed maintenance of a maintenance task requested at quarter *(t–a),* where *a* is a positive value.

It must also be noted that the value of the *Total_delivery* at a particular time will be affected by the recurrent fault rate, that is, the *CM_fault_rate* and the *EM_fault_rate*. These two factors reduce the *Total_delivery* as they cause outflows. A high reduction in this delivery also means low quality of maintenance; therefore, a low value of the *Total_delivery* can also be interpreted as the maintenance processes delivering much lower maintenance output.

In sum, the *Delivery_and_request_ratio* measures maintenance performance and therefore the software availability level. It is *unitless*. A low level of this ratio can be interpreted as low deliveries, low quality or both. This measure can also be interpreted as the performance level of overall maintenance staff.

5.3.6 Simulation model of eGSFs relationships – Motivational factors

This study adapts and extends McGrath and More's (1998) SD model of the expectancy theory into the SM of e-government systems. The extension of the model is undertaken by causally linking the model to the model's maintenance process sector in order to elucidate the causal feedback relationships of various motivational factors over time towards SM performance. The extension is also performed by causally connecting to the staff turnover and absenteeism factors. This elaboration is depicted in Figure 5.16.

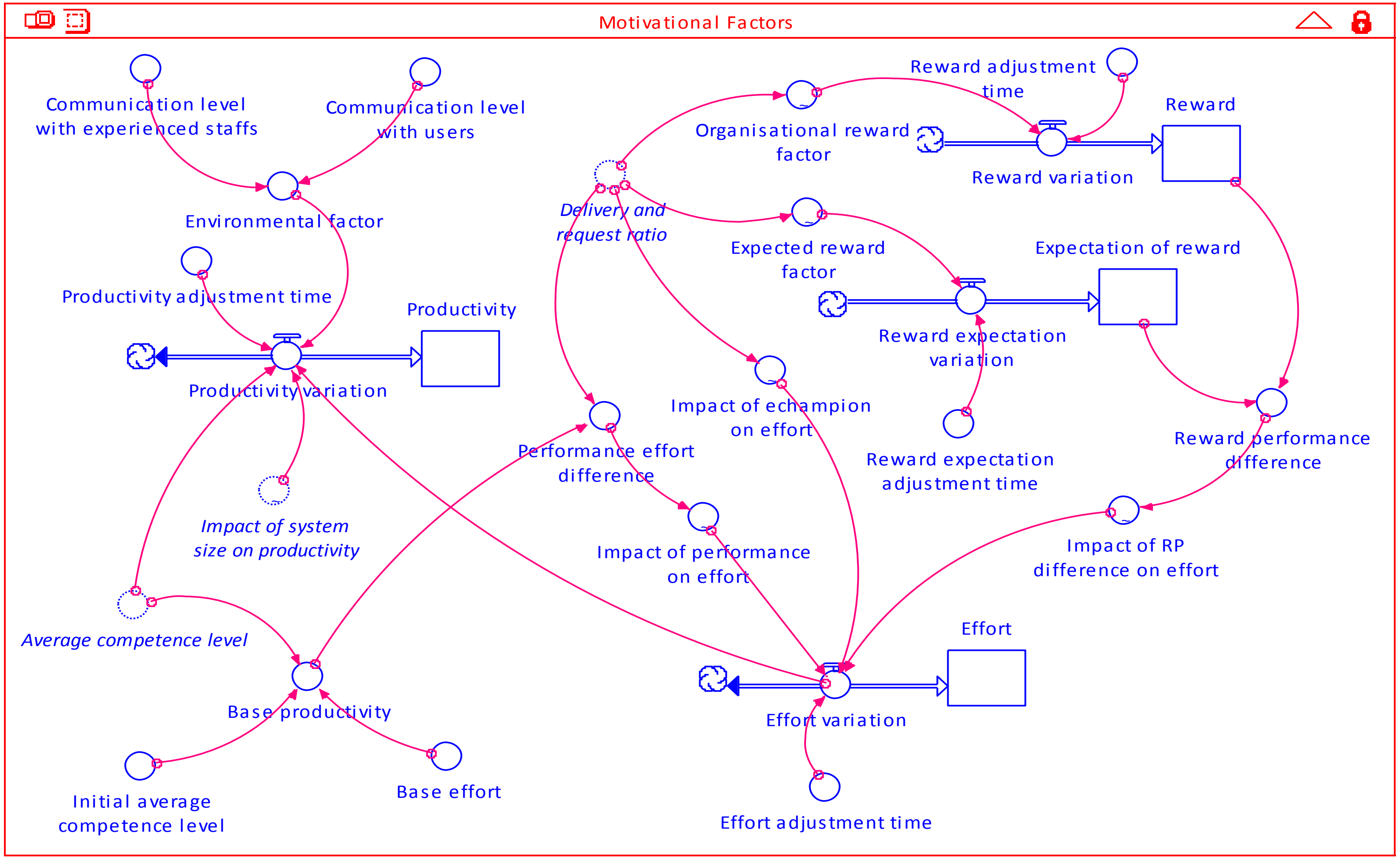


Figure 5.16 *The motivational factors sector of the model*

The links in Figure 5.16 can be identified from the fact that the *Productivity_variation* determines the delivery level of the completed maintenance which, in turn, is used to calculate maintenance performance. Eventually, the level of maintenance performance influences the rewards, both organisational and expected. The difference between these two eventually affects staff turnover, absenteeism and effort levels.

Staff effort. In software development project modelling, Abdel-Hamid & Madnick (1991**:** 85) defined staff effort as "actual fraction of a man-day on project". Implementing this definition to the SM, staff effort means the actual fraction of time spent by one staff member to carry out assigned maintenance tasks in one working day (official working hours). The actual amount of time spent to undertake the SM task can vary from time to time because of a variety of motivational, personal, environmental or social factors over time. So the actual amount of time can be less than, equal to or more than official working hours. For example, if the staff member spend six hours in a working day to carry out the maintenance task assigned to him/her, although the staff are officially in the workplace for eight working hours, then the staff effort is 0.75. On the other hand, if the staff spend 12 hours in a day to carry out the assigned maintenance, irrespective of where they do the task, then the staff effort is 1.5. Here, it is assumed that the staff can spend a maximum of up to 16 hours a day; hence, the staff effort could be as high as 2.

For the staff effort, a bi-flow and a stock will be implemented as in McGrath and More (1998) and Abdel-Hamid and Madnick (1991) to represent, respectively, the staff effort variation and cumulative effort over time. They are denoted by *Effort_variation* and *Effort*, respectively (see Figure 5.16). Referring to the definition of staff effort, the *Effort_variation* is the variation of the actual fraction of a staff member's working hours solely dedicated to maintenance activities in a working day. The bi-flow of the *Effort_variation*, which means it can be negative or positive, causes the *Effort* level to change. The variation causes staff productivity to fluctuate by assuming that other influencing factors are constant. A negative value of this variation can occur when the total value of the influencing factors is negative, namely, when there is a reduction of the effort due to the task completion time being faster than scheduled, and/or the rewards received less than expected. The reduction of effort means the staff prolongs the completion time.

The *Effort_variation* in the model is influenced by three factors, namely, staff performance through *Impact_of_performance_on_effort*, the e-champion factor through *Impact_of_echampion_on_effort*, and the rewards difference via *Impact_of_RP_difference_on_effort*. This variation is formulated as:

Effort_variation = (0.5 * (*Impact_of_performance_on_effort* + *Impact_of_RP_difference_on_effort*)) * *Impact_of_echampion_on_effort* / *Effort_adjustment_time*

The range of *Effort_variation* is set from 0 to 2, which means 0 to 16 working hours a day. The unit of this variation is *1/quarters.*

The description of each of these three factors will be provided in the next sub-subsections. Before describing these factors, it is first necessary to define *Base_effort* as it will be used in formulating those factors.

The staff effort is first determined by normal actual effort and is denoted by *Base_effort*. It is the fraction of actual normal working hours in one working day spent by a staff member on an assigned maintenance task. Following Abdel-Hamid and Madnick (1991), the value of the *Base_effort* is set to 0.6 which can be translated as 312 maintenance working hours out of an official 520 working hours per quarter. This is by assuming that the official working hours are eight hours a day and 65 working days in a quarter. The value of 0.6 is also used as the initial level of the *Effort*. Additionally, this value of 0.6 is utilised as a reference value in setting up the *Impact_of_performance_on_effort* and *Impact_of_RP_difference_on_effort*, each of which is represented as a line graph. The *Base_effort* is *unitless.*

The *Base_effort*, the staff *Average_competence_level* and the *Initial_ average_competence_level* (which will be described in the Staff competence section) are used to determine staff *Base_productivity*. The *Base_productivity* describes the productivity of staff when they exert normal effort using their current competence levels. It is formulated as:

*Base_productivity = (Average_competence_level *Base_effort) / Initial_average_competence_level*

As the staff *Average_competence_level* changes over time, then the *Base_productivity* also changes from time to time. The division is intended to scale the staff *Average_competence_level* and it causes the *Base_productivity* to be *unitless*. Theoretically, the range of the *Base_productivity* is 0 to an infinite value. This is so because the range of the *Base_effort* can be 0 to 2, and the *Initial_average_competence_level,* theoretically, can be 0 to infinity.

Impact of performance on effort. The relationship between staff performance and effort is presented in Figure 5.17 and is constructed as follows. As described in the CLD, the *Effort_variation* is influenced directly by the *Delivery_and_ request_ratio* level as the maintenance performance measure of the staff. The effort is a result of their judgement on the comparison between the exerted normal effort and the maintenance performance. There is a tendency to increase effort to a certain limit if the performance is lower than its required level; and, on the contrary, there is also a tendency to slow down (reduce) the effort if the performance is higher than expected (positive difference) (Abdel-Hamid & Madnick 1991). This concept is modelled by *Performance_effort_difference* and *Impact_of_ performance_on_effort.* The former variable is to formulate the difference between the actual performance of the staff and the staff *Base_productivity*, while the latter is to model the impact of this difference towards the staff effort.

Performance_effort_difference = *Delivery_and_request_ratio* –
Base_productivity

The range of the *Performance_effort_difference* is –1 to 1. The difference is –1 if the ratio is 0 (no delivery) and the *Base_productivity* is 1; and the difference is 1 if the ratio is 1 and the *Base_productivity* is 0. This is *unitless*.

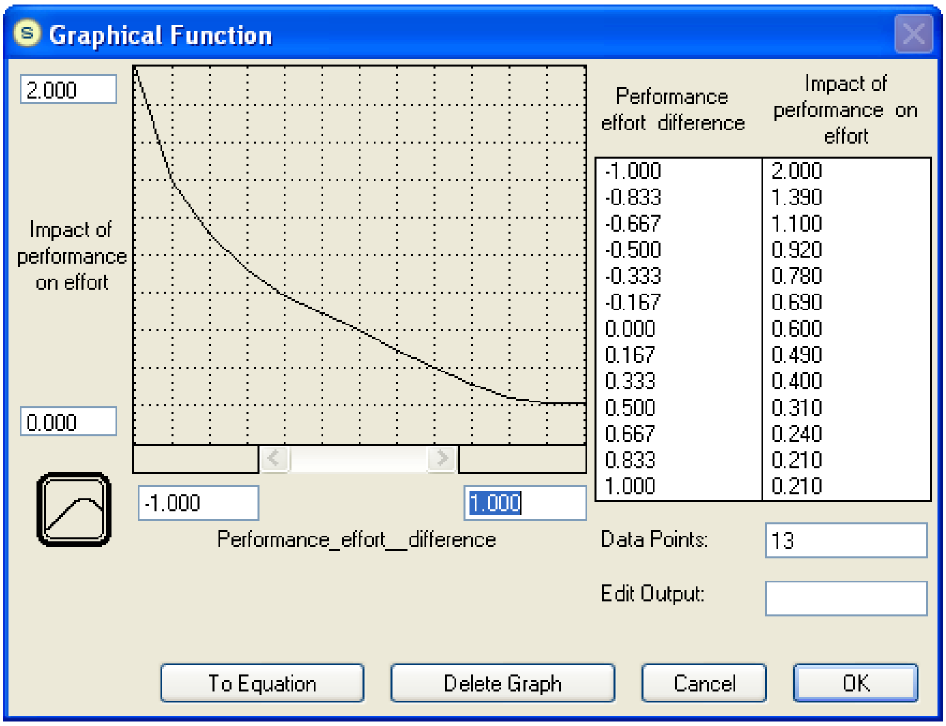


Figure 5.17 *The impact of performance effort difference on staff effort*

The relationship between performance and effort is represented explicitly as a line graph in Figure 5.17. For this initial model, the form of this graph serves for simulation instantiation. This graph gets input from *Performance_effort_difference,* which is set as abscissa, and provides output as *Impact_of_performance_on_effort,* which is set as ordinate. The interval value of the ordinate is 0 to 2. The value of the *Base_effort*, which equal to 0.6, is used as a reference in the graph in the sense that if the value of the *Performance_effort_difference* is equal to 0 then the value of the *Impact_of_performance_on_effort* is equal to 0.6. The impact on effort is about 0.2 (or 1.6 hours) out of official working hours a day when the difference is 1, and about 2 (or 16 hours) out of official working hours a day when the difference is –1. In other words, staff will reduce their working hours for maintenance if they observe that their current performance is higher than their performance-based normal working hours, or the other way round, by assuming that other factors are held constant.

Impact of rewards on effort. As described in previous subsections, staff effort is also motivated by the existence of organisational rewards over their performance and staff's perception of the value of rewards. The structural relationship between rewards and effort is represented in this modelling by a line graph describing the impact of the difference between the provided and expected rewards on the level of staff effort. The construction of this relationship is now described and refers to the expectancy theory and its SD representation, Figure 3.4.

With respect to the same maintenance performance level, it is very likely that there is a difference between the provided and expected organisational rewards. Before formulating the difference, denoted as *Reward_performance_difference,* it is necessary to create two representations of the relationships: first, between maintenance performance and organisational rewards; and second, between maintenance performance and staff expectation of organisational rewards.

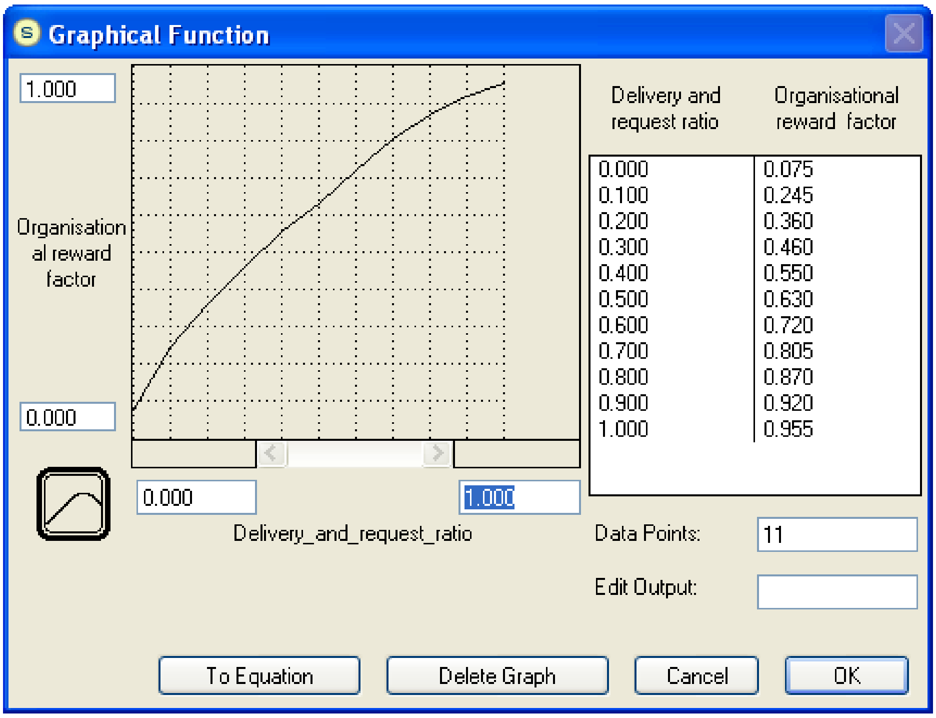


Figure 5.18 *The impact of actual performance on organisational rewards*

As shown in previous sections, the maintenance performance is represented by the *Delivery_and_request_ratio.* The impact of this performance on organisational rewards is described graphically by a line in Figure 5.18. This graph assumes that as the *Delivery_and_request_ratio* increases then the *Organisational_reward_factor* also increases, and vice versa. The graph pattern is presented to serve simulation instantiation and the detail curve pattern is somewhat arbitrary, although its characteristic as a monotonically increasing graph is quite plausible. The *Organisational_reward_factor* value is scaled ranging from 0 to 1 and represents total rewards provided by the organisation; it may include intrinsic and extrinsic rewards. A value of 1 represents maximum possible rewards that can be provided by the organisation. It is *unitless*.

Similar to organisational rewards, the impact of maintenance performance on staff expectation of organisational rewards is also described as a line graph, shown in Figure 5.19. This graph assumes that as the *Delivery_and_request_ratio* increases, the *Expected_reward_factor* which represents the staff expectation of rewards also increases, in S-form, and vice versa. Figure 5.19 is presented to serve simulation instantiation, and the curve pattern of the graph is somewhat arbitrary but plausible. The range value of the *Expected_reward_factor* is 0 to 1 and scaled, and may include expected intrinsic and extrinsic rewards. A value of 1 represents expected maximum possible rewards. This factor is *unitless*.

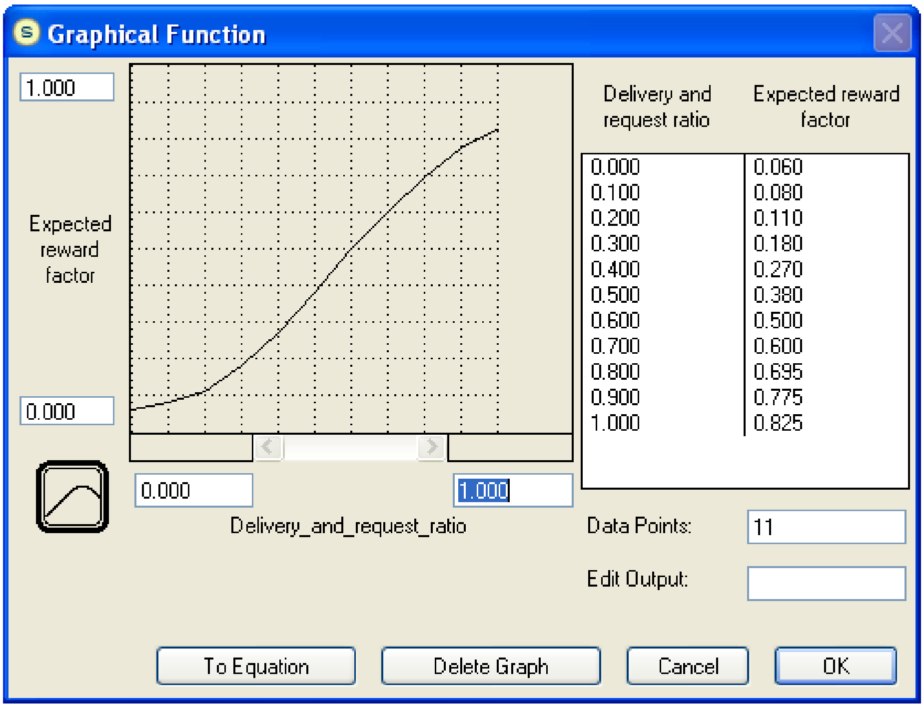


Figure 5.19 *The impact of actual performance on expected rewards*

Considering their relationship with maintenance performance, both the *Organisational_reward_factor* and *Expected_reward_factor* values will vary depend on the performance level. However, it must be noted that the provision of the rewards and the expectation are not always immediate; rather there are some delays. The comparison between provided and expected rewards that may lead to a difference may be based on the long term or in a cumulative way, especially for permanent staff. To facilitate the comparison, two pairs of flows and stocks will be constructed to reflect the variation of the provided and expected rewards and their cumulative values.

The rate of flow or the variation of the provided rewards is represented by *Reward_variation,* and its cumulative value is denoted by *Reward* stock. The units of these are *1/quarter* and *unitless*, respectively. The delay of the rewards provision is modelled by *Reward_adjustment_time,* the unit of which is *quarter*. On the other hand, the rate of flow or the variation of the reward expectation is represented by *Reward_expectation_variation,* and its cumulative value is the *Expectation_of_reward* stock*.* The units of these are *1/quarter* and *unitless*, respectively. The delay is denoted by *Reward_expectation_adjustment_time.* By using the *Reward* and *Expectation_of_reward* stocks, the difference of the provided and expected rewards at any time can be calculated. For this:

Reward_performance_difference = (Reward – Expectation_of_Reward)

In turn, the difference will be used to construct a representation of the effect of rewards on staff effort. The effect is represented by the *Impact_of_RP_ difference_on_effort*. The formulation of this effect utilises a line graph with the *Reward_performance_difference* as the abscissa and the effort exerted by the staff as the ordinate. The value of this effort factor ranges from 0 to 2. The value of 0 means the staff will reduce their effort by 100% of the base effort and 2 means the staff will increase their effort up to 16 hours work per day. When the value of the *Reward_performance_difference* is 0, then the *Impact_of_RP_difference_on_effort* is set to 0.6. This setting means that if the rewards provided by the organisation equal the staff expectation, then they work as normal. This is *unitless.* The graph is depicted in Figure 5.20.

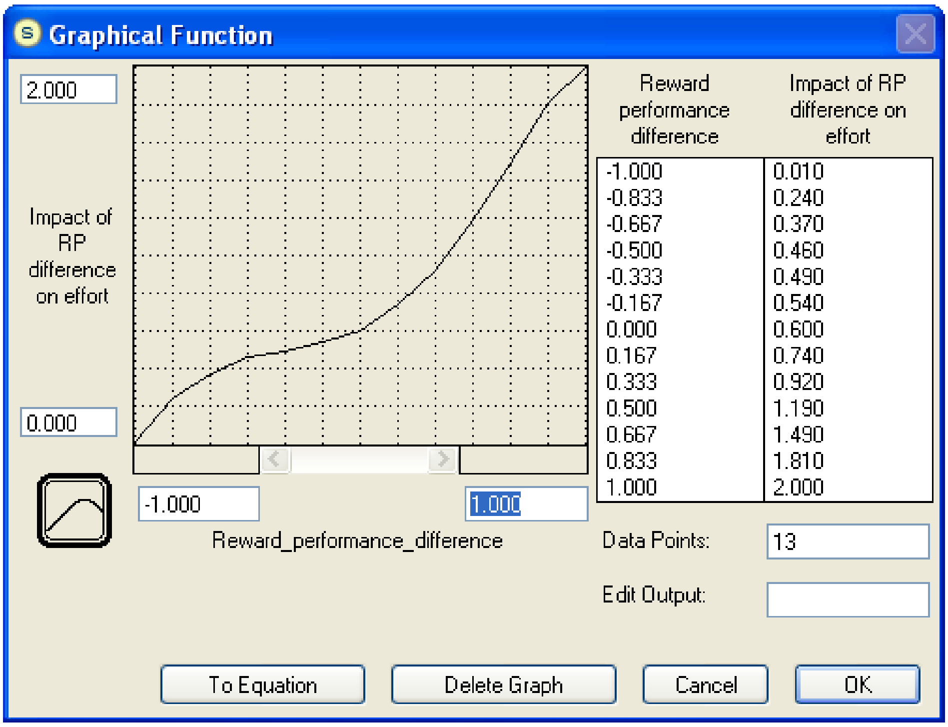


Figure 5.20 *The impact of reward performance difference on staff effort*

Impact of e-champion on effort. As with the aforementioned, in some cases the e-champion plays an important role. The e-champion can be interpreted as those staff members who have a high need for achievement. In this study it is assumed that they play their role if the maintenance delivery level is lower than a particular level. The impact of the e-champion is reflected by an increase in the staff effort variation. To represent this factor, the *Impact_of_echampion_on_effort* is introduced. It uses the *Delivery_and_request_ratio* as an input, and is utilised as a multiplier for the *Effort_variation*. The formulation of the *Impact_of_ echampion_on_effort* is depicted in Figure 5.21. The form of the graph is constructed for the simulation instantiation purpose. It is assumed that the maximum increase of effort is 50%, and there is no increase when the maintenance performance achieves 70% of the MR.

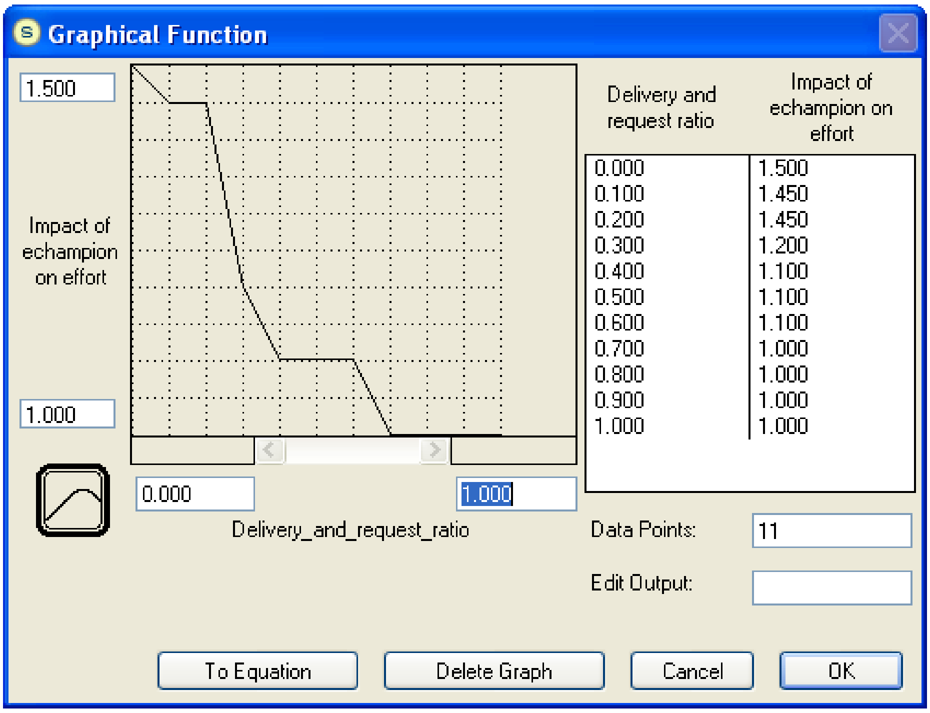


Figure 5.21 *The impact of the e-champion on staff effort*

Staff productivity variation. Staff productivity will be constructed by using potential and actual productivity concepts. The potential productivity is defined by the competence level of the staff, while the actual productivity is defined by potential productivity minus losses. Therefore, the actual productivity is always less than or equal to the potential productivity. The competence level concept will be introduced in the next section. Referring to this description, it can be interpreted that the actual productivity equals the potential productivity when staff motivation and working environment are perfect. Therefore, actual productivity also depends on the level of motivation and working environment. From an operational perpective, the potential productivity is the number of maintenance tasks that can be completed by one person in one quarter when there is no loss due to any kind of factor.

The SD representation of the staff productivity consists of a pair of flow and stock, named *Productivity_variation* and *Productivity* respectively. The flow, which describes the staff productivity variation over time, is a multiplicative function of four factors:

*Productivity_variation = Effort_variation * Average_competence_level * Environmental_factor * Impact_of_system_size_on_productivity * Productivity_adjustment_time*

These represent four factors that determine actual productivity: potential productivity; staff motivation; environment; and job characteristics. The staff productivity is also assumed to be time-dependent, that is, on *Productivity_adjustment_time*. The dimension of the *Productivity_variation* is *tasks/person-quarter*.

Two variables will be used to represent the environment contribution to the staff productivity: communication level with experienced staff and communication level with users. The job characteristic is the system size and complexity. The inclusion of the first variable is based on the fact that some literature indicated that, for non-experienced staff, the output of maintenance work is tested by experienced staff. This requires smooth communication. In other cases, although the task is assigned to one staff member, it is common that this staff member will seek more expert advice within the office unit when conducting CM. Accordingly, the level of this type of communication determines the staff productivity. This type of communication is represented by *Communication_level_ with_experienced_staff.* The value of this variable is set to 0 to 1, or from totally imperfect communication to totally perfect. To instantiate the simulation, the *Communication_level_with_experienced_staff* is set to 0.95, meaning that there is 5% reduction in the staff competence level. In many cases, staff have to communicate with the users who request maintenance when performing the required maintenance task. Communication with users is modelled by *Communication_level_with_users* and an instantiation value of 0.9 is assigned to this variable.

The average value of these two factors represents *Environmental_factor*. The value ranges from 0.0 to 1.0 and the factor is unitless. The value of *Environmental_factor* is set as such because this factor represents the staff productivity loss.

The system complexity is measured by the software size in SLOC. The more complex the software, the more difficult to solve the maintenance tasks. Accordingly, an increase in system size will reduce staff productivity.The impact of the system size towards the staff resolution ability, which is denoted by *Impact_of_system_size_on_productivity,* is represented by a line graph in Figure 5.22.

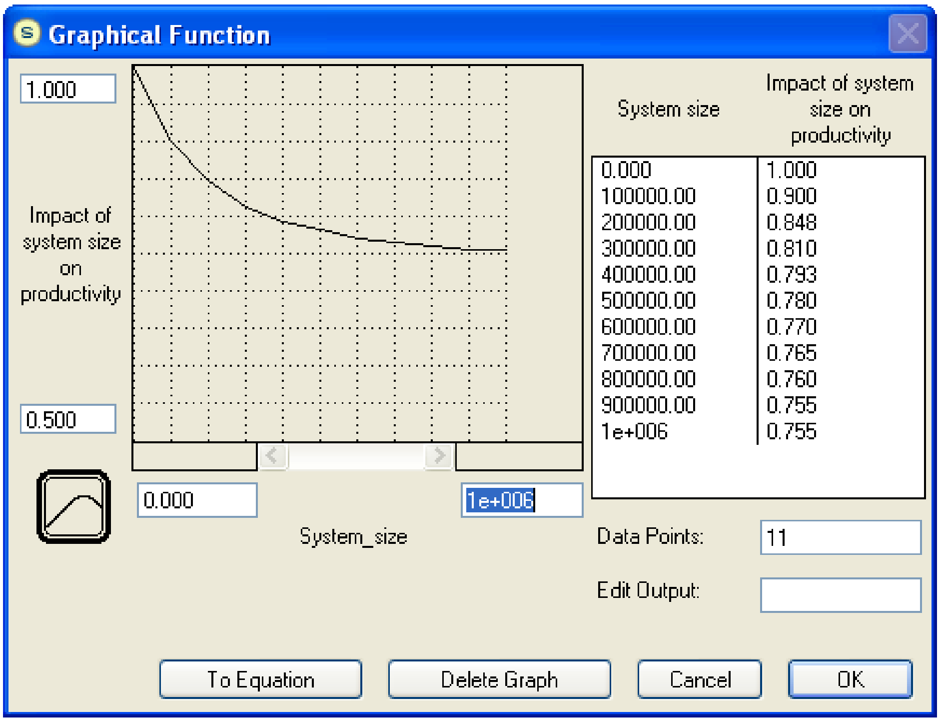


Figure 5.22 *The impact of system size on staff productivity*

The *Productivity_adjustment_time* factor represents the time adjustment required by the influencing factors to take effect. As this factor is multiplied by the other factors that influence the *Productivity_variation*, so a higher value of the *Productivity_adjustment_time* means quicker flow of the variation.

5.3.7 Simulation model of eGSFs relationships – Staff competence

The level of staff competence has been shown by much research to be an influential factor for SM performance. This level, which has dynamic characteristics, has been used to define the potential productivity of staff. The CLD has indicated that the competence level of IS staff always decreases over time, mainly because of the ever-changing advancements in IT. However, over time their competence level also increases due to experience gained by doing their jobs, self-learning and training attended. These two different directions of factors compensate for each other. From an organisational perspective, staff recruitment and turnover affect the overall level of staff competence level.

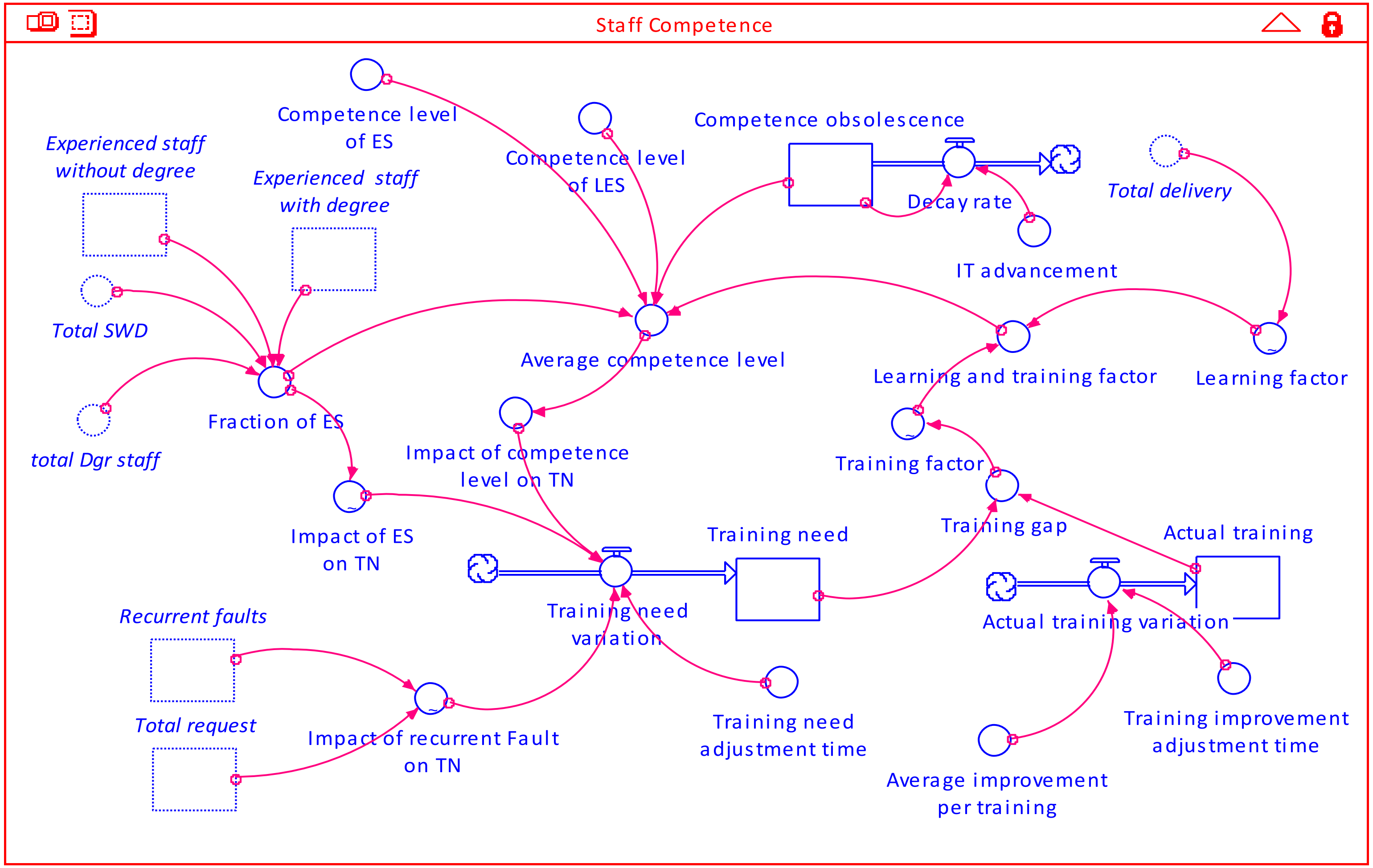


__Figure 5.23__ The staff competence sector of the model

Average competence level. Within an SM unit, the average staff competence level depends on the staff's level of education and experience, the fractions between experienced and less experienced staff, the learning and training process during their involvement in system maintenance activities, and the rate of IT advancement.

Technically, referring to Figure 5.23, the average staff competence level is denoted by:

*Average_competence_level= ((Fraction_of_ES * Competence_level_of_ES) + (1 –Fraction_of_ES) * Competence_level_of_LES) * Learning_and_training_factor * Competence_obsolescence*

The *Competence_level_of_LES* and *Competence_level_of_ES* are the level of competence of less experienced staff and the level of competence of experienced staff, respectively. The dimension of the *Average_competence_level* is *tasks/person-quarter*.

To determine the value of the *Average_competence_level*, this study refers to Jorgensen's (1995) data in which, on average, one staff member requires 5 days or 0.077 quarters to complete a maintenance task. In particular Jorgensen (1995) also reported that, on average, one CM task takes 0.026 quarters (or 1.7 days) and one EM task needs 0.115 quarters (or 7.5 days) of maintenance effort. Using the average of overall maintenance task completion time, which equals 5 days, the level is equivalent to 13 *tasks/person-quarter*. It is assumed that this value refers to the time required by an experienced staff member to complete one maintenance task. In comparing experienced and less experienced staff, Abdel-Hamid and Madnick (1991) considered that the level of competence of experienced staff is two times higher than that of less experienced staff. This data is used to set the (initial) value of the *Competence_level_of_LES* to *6.5 tasks/person-quarter* and the *Competence_level_of_ES* to 13 *tasks/person-quarter*. These settings represent the number of tasks that can be completed by less experienced and experienced staff, respectively, in a quarter. Furthermore, the *Fraction_of_ES* is defined as:

Fraction_of_ES = Experienced_staff_with_degree + Experienced_staff_without_degree) / (total_Dgr_staff + Total_SWD)

It describes the proportion of experienced staff to total staff. The value on the right-hand side of this equation depends on the values of the associated factor in the Staff development sector.

As noted from the formulation of the *Average_competence_level*, the second component that influences the value of this level is the *Learning_and_ training_factor*, which is a multiplier to the *Average_competence_level*. As shown by previous research, the impact of this factor is an increase in the *Average_competence_level;* therefore the minimum value for this factor is 1. The *Learning_and_training_factor* is the sum of *Training_factor* and *Learning_factor,* each of which will be explained in the following sub-subsections.

Learning factor. The *Learning_factor* accommodates the increase of the staff competence level due to knowledge gained from maintenance experience, which is indicated by the number of maintenance deliveries. As in Abdel-Hamid and Madnick (1991), this study adopts an S-type graph to represent this increase. The graph describing the effect of total delivery of completed maintenance tasks in a quarter on learning is depicted in Figure 5.24. This variable is *unitless* and, along with the training factor, is a multiplier to the average competence level and is set to have values range from 1 to 1.4. The value 1 means the staff is new and the number of the maintenance tasks delivered is 0. It is estimated that 1.4 is the maximum value, because the value will level off at a particular point as a result of the staff dealing with the same software over time.

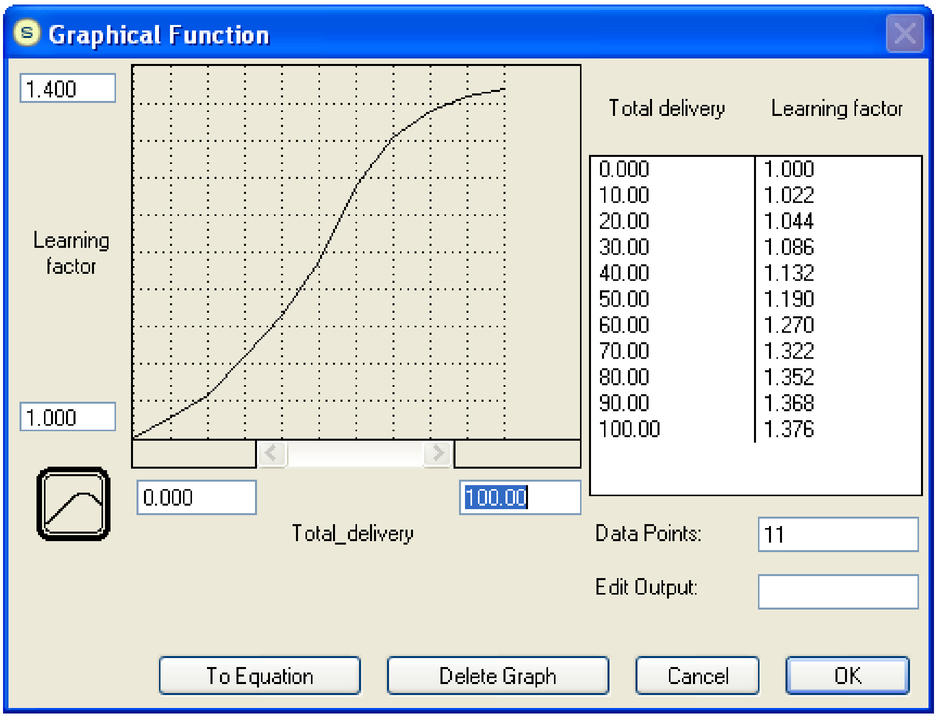


Figure 5.24 *The relationship between total maintenance delivery and staff learning*

Training factor. As shown in the literature review and CLD, training is a critical way to overcome degradation in either individual or overall staff competence levels. Training can take various forms: formal, informal, structured or unstructured; it is normally based on specific needs and has a specific level of actual effectiveness. In this SD simulation construction, the training factor is denoted by *Training_factor* and is *unitless.*

To formulate the *Training_factor*, there are two aspects that need to be considered: the training level needed and the actual training. These two factors are formulated as pairs of stock and flow. The first is a flow called *Training_need _variation* and a stock named *Training_need,* the units of which are *1/quarter* and *unitless* respectively; and the second is represented by *Actual_training_variation* and *Actual_training,* the units of which are *1/quarter* and *unitless* respectively*.* The difference between the cumulative value of the *Training_need* and the *Actual_training* is used to determine the improvement of the competence level caused by the training through *Training_gap,* which is also *unitless*. It assumed that any training will improve the competence level, but the level of impact of the training on the competence level depends on the gap value. In this case, if the gap equals 0 then the improvement level is 0.2. This value is derived from Murthy et al. (2008)

who found a 20% improvement in a performance measure of the staff at a call centre as a result of appropriate training. This data is used as a proxy for simulation instantiation. The *Training_gap* is set to –1 to 1. The relationship between the *Training_gap* and and the *Training_factor* is depicted in Figure 5.25.

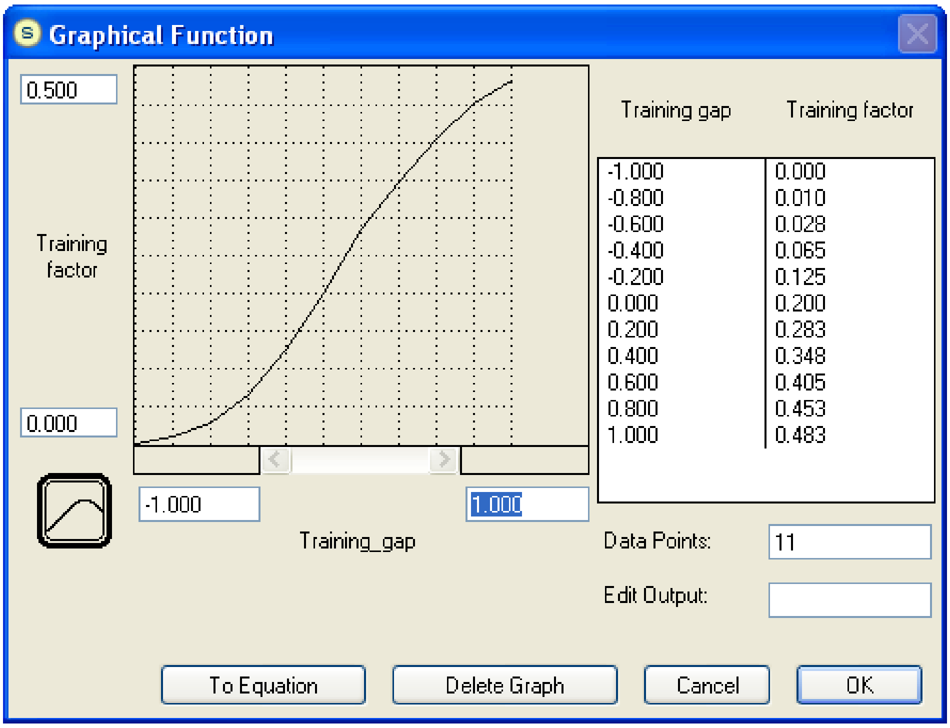


Figure 5.25 *The impact of training gap on the level of training factor*

The graph describes that if the actual training is much higher than the needed, then the impact on the improvement level is high. On the other hand, if the actual training provided to the staff cannot fulfill the needed training then the impact is low. It can be seen on the graph that if the *Training_gap* value is 0 then the *Training_factor* value is 0.2.

The level of the *Actual_training* stock is determined by the *Actual_training_variation* flow. In turn, this flow is determined by *Average_improvement_per_training,* which is *unitless,* and *Training_improvement_adjustment_time,* the unit of which is *quarter*. In this case, it is assumed that every quarter there is one training run, each of which can improve the staff competence level by 20%. So, if the results of the actual training match with the training need i.e. the *Training_gap* is 0, then the *Training_factor* contributes to the improvement of the competence level by about 20%.

As in the case of the actual training, the cumulative level of the *Training_need* is determined by the *Training_need_variation* flow. The flow of training need is affected by the declining competence level of staff from time to time due to advances in IT; the recurrent faults level that indicates the quality of completed maintenance; and the fraction of experienced staff as a result of the increase in the number of new less-experienced staff. Each of these factors may cause different degrees of training need. These three factors are then modelled by *Impact_of_competence_level_on_TN*, *Impact_of_recurrent_fault_on_TN* and *Impact_of_ES_on_TN* respectively, and all of which are *unitless*. Then, the *Training_need_variation* is a weighted average of these three factors divided by *Training_need_adjustment_time.* The unit of this adjustment time factors is *quarter*. The weighted average is set such that the total of these three factor is equal to 1.

The effect of the natural decline of the competence level due to IT advancement on the training need is depicted in Figure 5.26. The graph describes that to maintain an appropriate competence level, training is needed to compensate for the decline. It is assumed that when the number of maintenance tasks that can be completed per quarter per person is 1 then the training needed is 0.33 and, on the other hand, if it is 65 then the level of the training needed is 0.

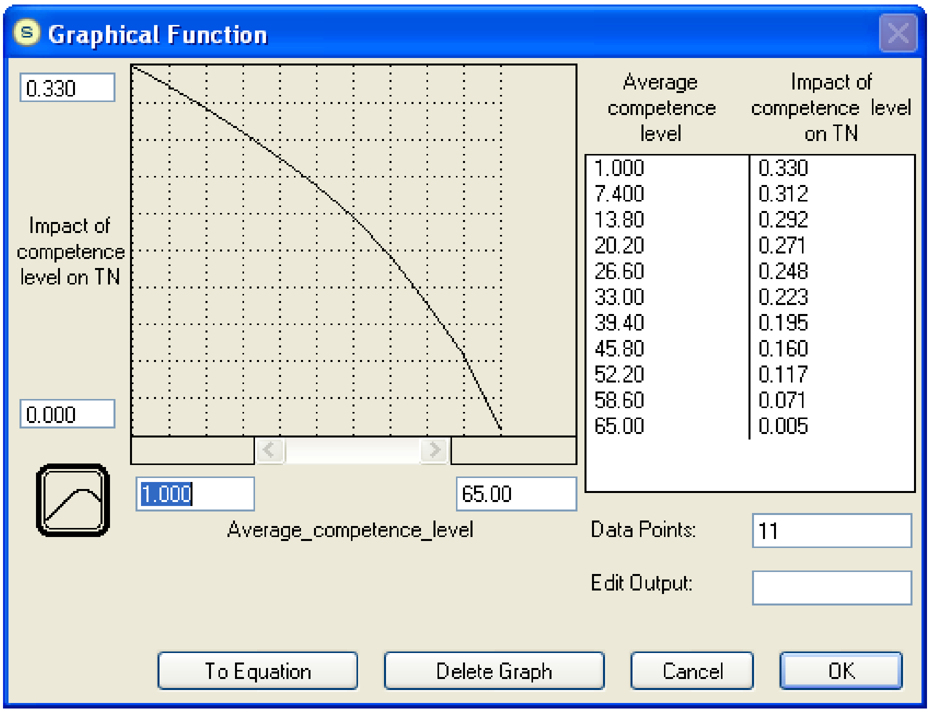


Figure 5.26 *The impact of average competence level on level of training needed*

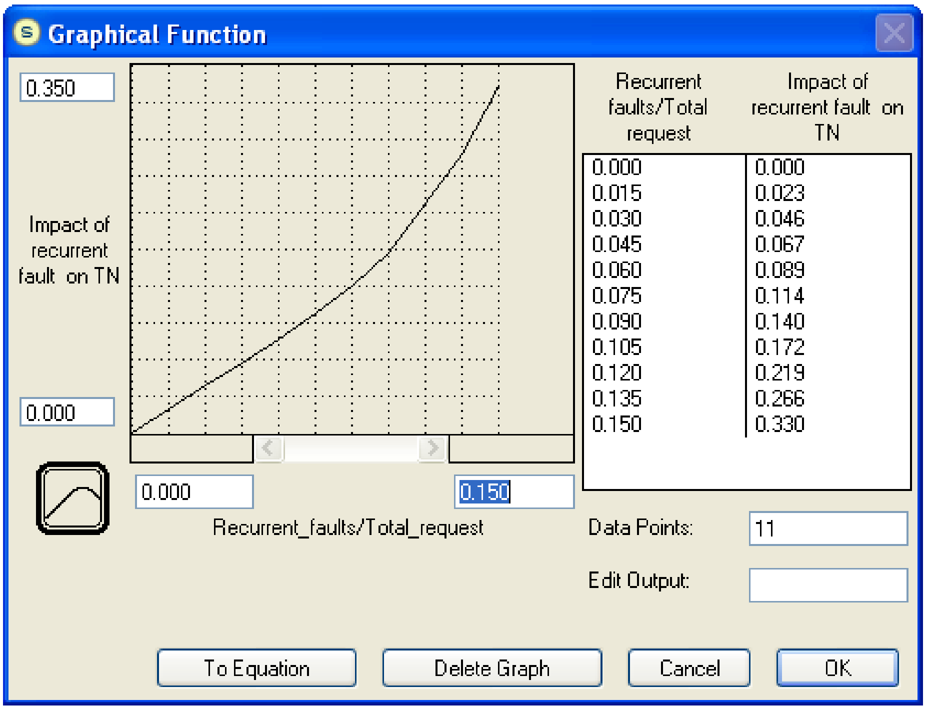


Figure 5.27 *The impact of recurrent faults on level of training needed*

The impact of the recurrent faults on the training need is presented in Figure 5.27. The graph informs that as the recurrent faults percentage increases, the quality of the maintenance delivery decreases, which can also mean the competence level decreases. This eventually causes a need for an appropriate level of training.

The last factor is determined by incoming new staff or outgoing experienced staff, because these increase the proportion of less experienced staff over total staff, which also means a decrease in the average of the overall competence level. This in turn causes the training need. The level of training needed as a result of the increase in this proportion is depicted in Figure 5.28.

To give a meaning to the quantitative value of the variation of the need for training, two aspects need to be considered: frequency of the training per year, and the degree of need compared with the average staff competence level.

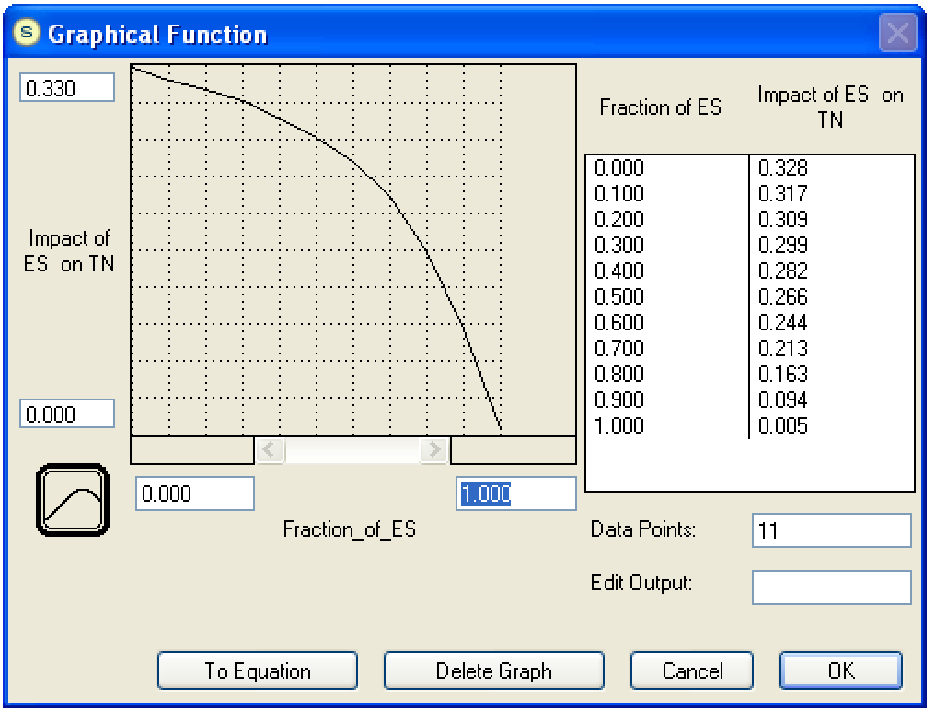


Figure 5.28 *The impact of the fraction of experienced staff on level of training needed*

Information technology advancement. The ever-improving IT naturally causes staff competence to become obsolete. However, there is no such figure that can be used as a general measure for this decline. For this reason, Moore's Law will be used as a reference. Moore's Law of the computing power of a computer central processing unit stated that this power is approximately doubled every two years (Mazor 2009). Therefore it may be assumed that, from the other point of view, the competence level of SM staff will be halved every two years. And from the SM tasks need perspective, at a particular point the decrease is such that the level of staff competence is unable to cope with the need of the SM tasks. If the average decrease is about 50% per two years, then per quarter the staff loses their competence roughly by about 0.0625. To represent this factor, a multiplier of the staff competence level named *Competence_obsolescence* is constructed as a stock, the level of which is drained by an outflow called *Decay_rate*. The unit of these two factors are *unitless* and *1/quarter*. This rate is:

*Decay_rate = Competence_obsolescence * IT_advancement*

The *IT_advancement* is set to 0.0625, and its unit is *1/quarter*. The result of this outflow is the level of the stock which decreases according to a negative exponential pattern.

5.3.8 Simulation model of eGSFs relationships – Staff development

The review of SD literature has indicated that many stock and flow models of the dynamic of human resources management have been developed or proposed. For example, within the context of a software development project, Abdel-Hamid and Madnick (1991) constructed one layer of software staff consisting of two phases of competence improvement, and Morecroft (2007) presented a two-layer model of health professional planning within the public sector.

By referring to Abdel-Hamid and Madnick's (1991) and Morecroft's (2007) models as well as the data from PUSDATIN (2006), IS staff development is modelled using two layers, each of which consists of two phases. In short, the model starts with staff recruitment, which results in less-experienced staff and ends with staff retirement. The first layer accommodates staff members without degrees, while the second one models the staff with degrees. A periodic recruitment for both types of staff is conducted considering the gap between the number of staff needed and the number currently available. The model accomodates the possibility that a staff member without a degree can leave the organisation to pursue a formal academic degree while retaining their employment status; this is the case, for example, when they are awarded a scholarship. It is also assumed that staff can quit or be transferred to other government units at any phase. Modelling in this sector is in association with several factors previously described in the CLD: total staff; experienced staff turnover; new staff; and fractions of inexperienced and experienced staff.

The numerical values used in this sector are mainly for simulation instantiation.

Figure 5.29 presents the model for this IS staff development sector. The name of the model elements for staff without degree is indicated by *SWD* and those with degree are indicated by *Dgr*. Starting with staff recruitment, referring to Figure 5.29, the element of the model of the recruitment phase of staff without degree is:

SWD_recruitment_rate = SWD_recruited / SWD_recruitment_delay

The value of *SWD_recruitment_delay* is 4 which means recruitment occurs once in a year. The incoming new non-degree staff accumulates in *Less_experienced_staff_ without_degree* stock, the initial value of which is 2. Those staff may quit or be transferred to other units during this early phase of employment. The rate of this outflow is denoted by:

*Less_ESWD_transfer_rate = Less_experienced_staff_without_degree * less_ESWD_transfer_fraction*

The initial value of this fraction is 0.001. Through improvement that takes time, the rate of the staff who become experienced staff is denoted by:

SWD_improvement_rate= Less_experienced_staf_without_degree / SWD_improvement_time

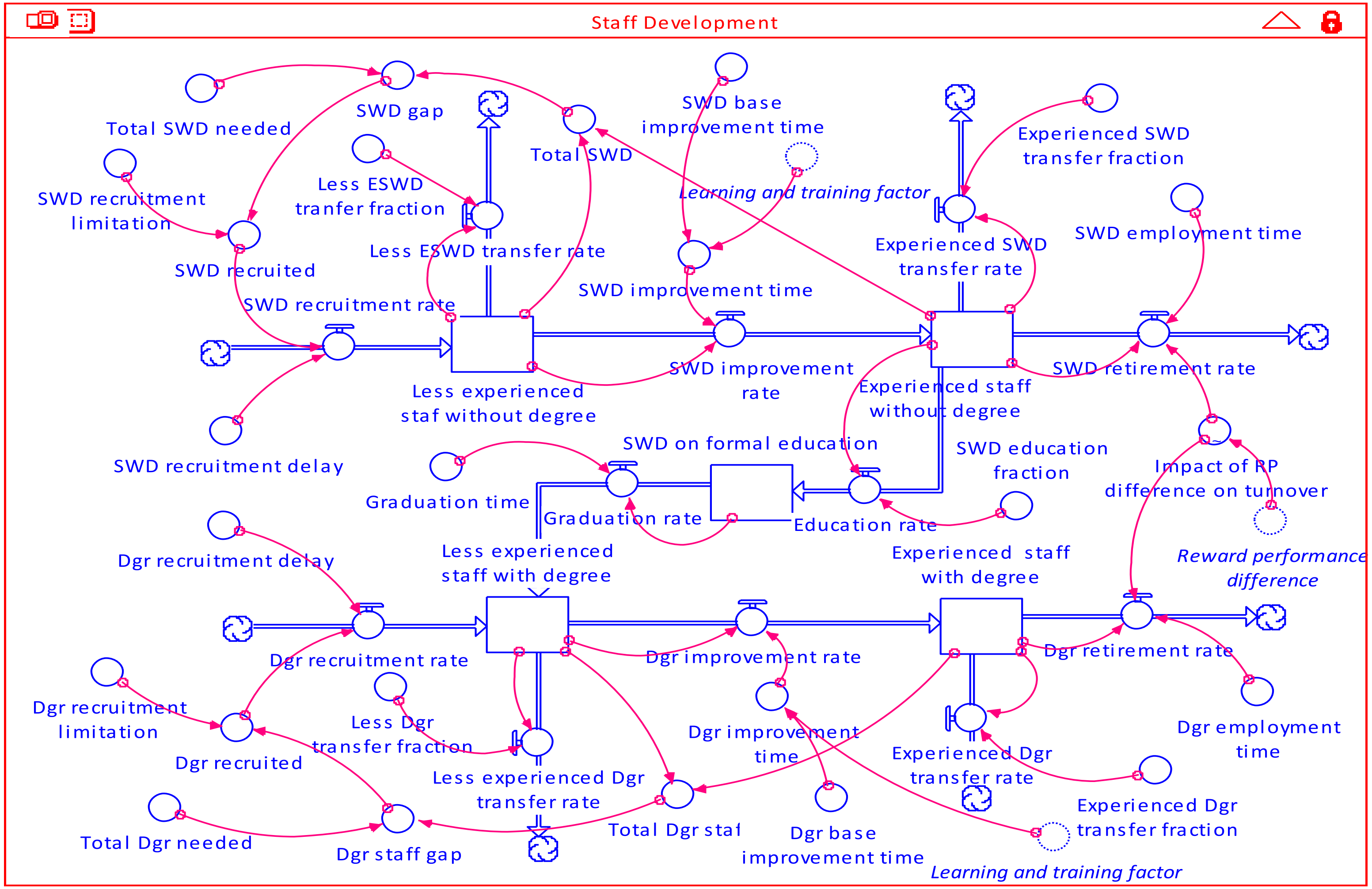


***Figure 5.29** The IS staff development management sector*

In turn this divisor is determined by:

SWD_improvement_time = SWD_base__improvement_time / Learning_and_training_factor

This base improvement time is set to 4, which means a staff member requires 4 quarters to improve to become experienced. The involvement of the latter factor is to accommodate the fact that to become experienced the staff should gain experience from learning and improve their competence through training. The cumulative number of experienced staff without degree is denoted by a stock named *Experienced_staff_without_degree*, the initial value for which is 2. There is also a possibility that an experienced staff member will quit or be transferred to other units at this stage and this is reflected by:

*Experienced_SWD_transfer_rate = Experienced_staff_without_degree * Experienced_SWD_transfer_fraction*

The value of this fraction is 0.001. In addition to this transfer rate, the level of the *Experienced_staff_without_degree* also decreases as a result of staff retirement or turnover. The retirement and turnover rate is formulated by:

*SWD_retirement_rate = Experienced_staff_without_degree/ (SWD_employment_time * Impact_of_RP_difference_on_turnover)*

The employment time is set to 20 quarters; in other words, staff members keep their jobs in this unit for five years. This formulation describes that the retirement rate is affected by the difference between reward and performance. The formulation of the *Impact_of_RP_difference_on_turnover* is depicted in Figure 5.30.

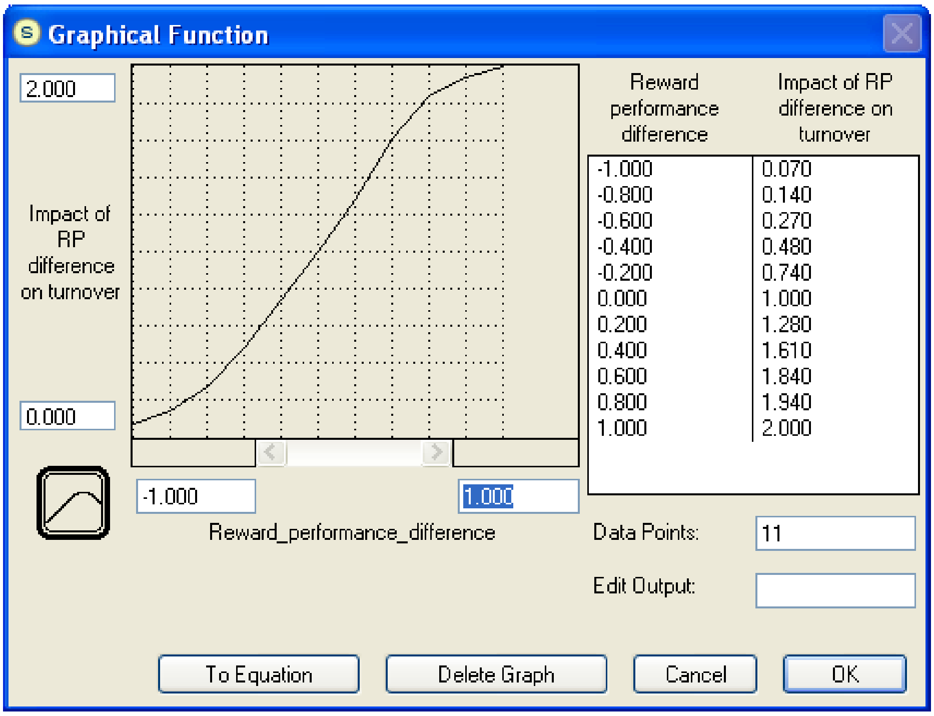


Figure 5.30 *The impact of rewards performance difference on retirement rate*

In addition, it is assumed that only experienced non-degree staff can pursue further formal education leading to a degree and temporarily leave their employment status. This reduces the experienced non-degree staff number by:

*Education_rate = SWD_education_fraction * Experienced_staff_without_degree*

The value of the fraction is set to 0.01. Time needed to complete their study is *Graduation_time* which determines *Graduation_rate*. The value of this time factor is set to 8 quarters or 2 years. As they graduate, they join less experienced degree staff. The reason for this is that during the long education time they leave their maintenance job completely. By this fact, they are also not part of the total maintenance staff during their period of study.

In this sector, the total staff without degree needed is determined by the maintenance management and represented by *Total_SWD_needed*. It is assumed that the value for this needed staff is 5. The difference between the number of current staff and needed staff equals *SWD_gap*. So,

SWD_gap = Total_SWD_needed – Total_SWD

In turn, this gap triggers staff recruitment. However, it must be noted that the number of staff needed may not always be satisfied because of budget limitations. Therefore, *SWD_recruited* is determined by:

SWD_recruited = if SWD_gap>0 then
min(SWD_gap,SWD_recruitment__limitation) else 0

The limitation is assumed to be 3 persons.

In this sector, the units of all stocks are *persons*, all rates are *persons/quarters*, all fractions are *1/quarters*, all improvement or employment times or delays are *quarters*, and all recruitment limitation and staff needed factors are *persons.*

The construction for the staff with degree layer mirrors the staff without degree layer. The stocks, flows and converters of the staff with degree layer are indicated by *Dgr* or *degree* within their factor names instead of *SWD*. The values assumed for *Less_experienced_staff_with_degree* is 4, *Experienced_staff_with_degree* is 5, *Dgr_recruitment_limitation* is 5, *Total_Dgr_needed* is 10, *Dgr_base_improvement _time* is 2 and *Dgr_employment_time* is 20.

Both the non-degree and degree staff are considered by this study to have the same role in SM but with different ranges of responsibility. Those staff members who have degrees and experience are also responsible for supervising less experienced (degree and non-degree) and experienced non-degree staff in addition to carrying out maintenance jobs. Supervision activities are considered on-the-job training for those being supervised.

5.3.9 Simulation model of eGSFs relationships – Staff allocation

The model sector for staff allocation depicted in Figure 5.31 is derived from Abdel-Hamid and Madnick (1991); however, this study includes staff absence level and CM urgency level. The allocation includes the assignment of more experienced staff to supervise less experienced ones. The eventual output of this sector is the number of staff allocated for CM and EM.

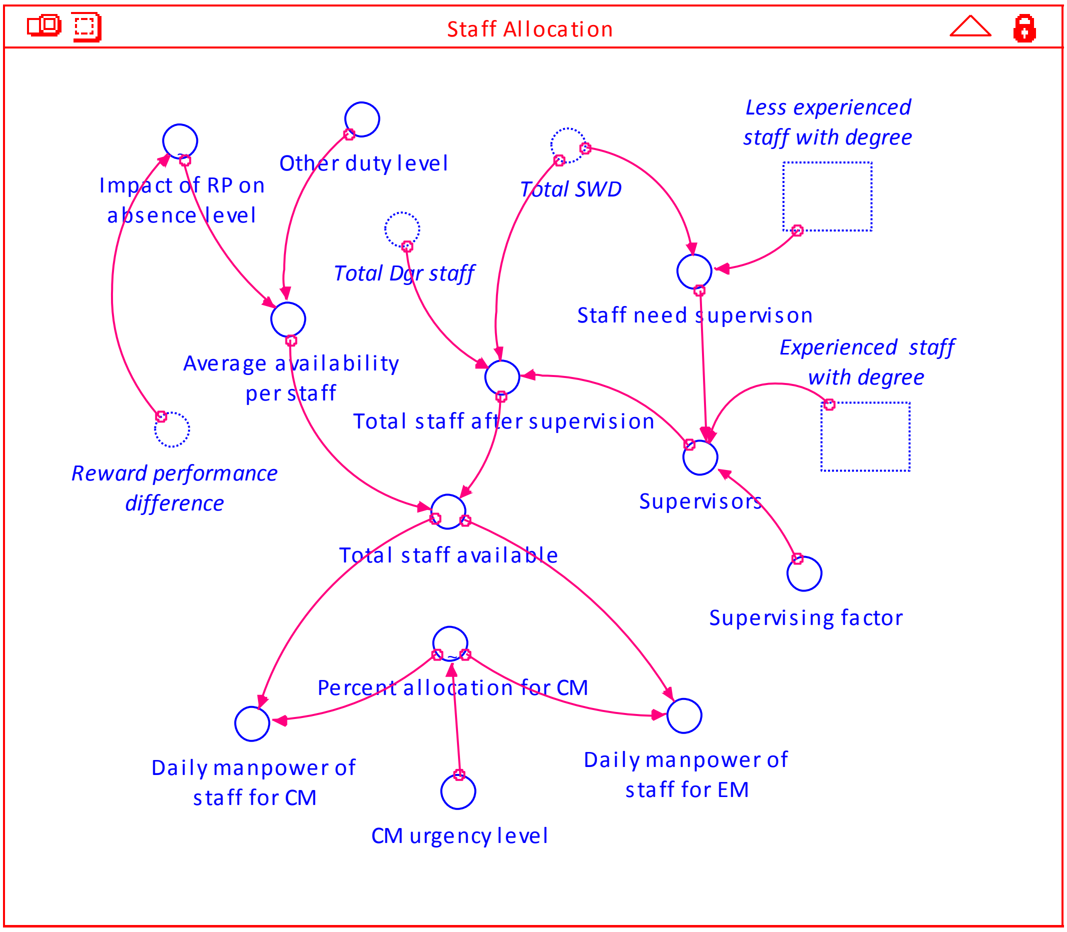


Figure 5.31 *The IS staff allocation sector*

Absenteeism and other duties. The absence level affects average staff availability and is affected by the difference between provided and expected rewards as shown in CLD, Figure 5.6. The absence level is defined as the percentage of not being at the office without planning for personal reasons out of the total number of working days in one year. Staff may be absent for various reasons including sick leave, moonlighting or other motivation-based reasons. As skillful employees, staff with a significant competence level in software development and maintenance are assumed to be likely to behave like other skillful workers such as teachers, nurses or physicians. In developing countries, they have chances to earn more income through moonlighting while at the same time retaining their employment status. The organisational rewards and external opportunities might have a significant influence on the level of staff absence.

In addition to absenteeism, other duties also affect overall the level of staff availability for SM. Staff may also be assigned to other duties such as user support services, new system development, administrative jobs and organisational meetings in addition to the main task of maintaining the software of the e-government system.

In this model sector, staff absence and other duty levels are denoted by *Impact_of_RP_on_absence_level* and *Other_duty_level* respectively. Both of these levels are *unitless.* The level of absence of the staff is assumed to be influenced by motivational factors and is represented as a graph shown in Figure 5.32; and the *Other_duty_level* is set to *20%,* a constant. The average availability of the staff is modelled by the *Average_availability_per_staff* which is formulated as:

Average_availability_per_staff = MAX(0, 1 – (Impact_of_RP_on_absence_lvl + other_duty_lvl))

This availability factor is *unitless.* The use of *MAX* is to prevent the staff availability from having negative values.

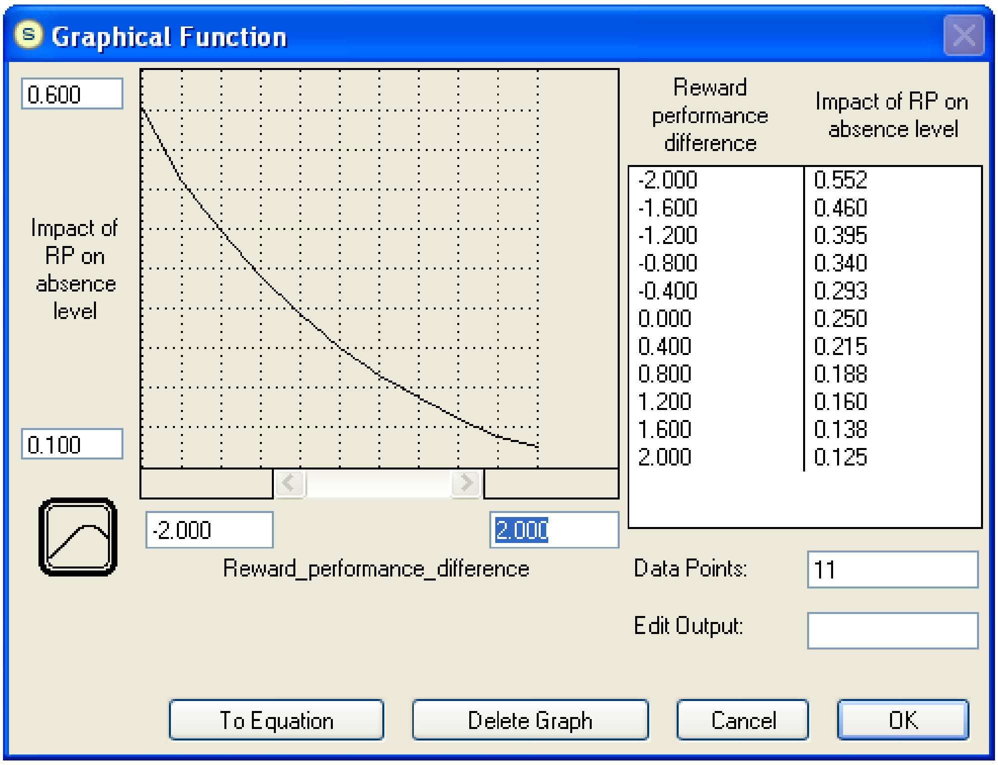


Figure 5.32 *The relationship between reward performance difference and staff absence level*

Supervision. In addition to absenteeism and other duties, staff supervision may also affect the total availabaility of staff. The reason for this is that a proportion of the maintenance staff needs supervision in doing their maintenance jobs, which could be considered as a process of on-the-job training for less experienced staff. The proportion consists of, referring to the previous sector, all non-degree staff and less experienced degree staff. The total number of these staff members is modelled by *Staff_need_supervision*. Its unit is *persons.* To determine staff availability, it is assumed that only experienced degree staff can supervise. In this case, they are called supervisors. To calculate the number of supervisors needed, a supervising factor is introduced. In this case, it is assumed that one supervisor can supervise up to four staff. If the supervisor number needed is larger than the number of experienced degree staff, then all the experienced degree staff are assigned the supervisor job. The number of supervisors is denoted by the *Supervisor* and its unit is *persons.* It must be noted that the value of the *Supervisor* factor cannot be larger than the number of experienced degree staff available.

Supervisors = if *(Staff_need_supervison/Supervising_factor) > Experienced_staff_with_degree* then *Experienced_staff_with_degree* else *(Staff_need_supervison/Supervising_factor)*

In turn, the value of the *Supervisor* variable will be used to calculate the *Total_staff_after_supervision*. This factor is formulated as:

Total_staff_after_supervision = total_SWD + total_Dgr_staff – supervisors

This *Total_staff_after_supervision* factor is used to accommodate the fact that the supervision activities also further reduce the availability of maintenance staff. The unit for this factor is *persons.*

Further, by considering both the *Average_availability_per_staff* and the *Total_staff_after_supervision*, the *Total_staff_available* for maintenance can be formulated.

*Total_staff_available = Total_staff_after_supervision * Average_availability_per_staff*

This represents the actual total availability of staff to perform SM and its unit is *persons*.

Staff allocation. Once the value of the *Total_staff_available* is obtained, the staff are then allocated for udertaking either CM or EM. For this allocation, the proportion of staff for CM and EM is initially 50% each. But, as CM has higher priority than EM then the proportion of staff allocated for correction can be higher. To determine the proportion for each maintenance type, a factor called *Correction_urgency_level* is introduced to the model. It is *unitless* and its value is determined by the maintenance management. Based on the value of this factor, the *Percent_allocation_for_correction* value is calculated. Figure 5.33 shows the percentage value of the staff allocated for the CM. The value ranges from 50% to 89.5% of the total staff available for maintenance, which depends on the level of the CM urgency assigned by maintenance management. This allocation results in the *Daily_manpower_of_staff_for_CM* and the *Daily_manpower_ of_staff_for_EM*. The first represents the average actual number of persons for CM and the second represents the average actual number of persons for EM. The unit of these two factors is *persons*.

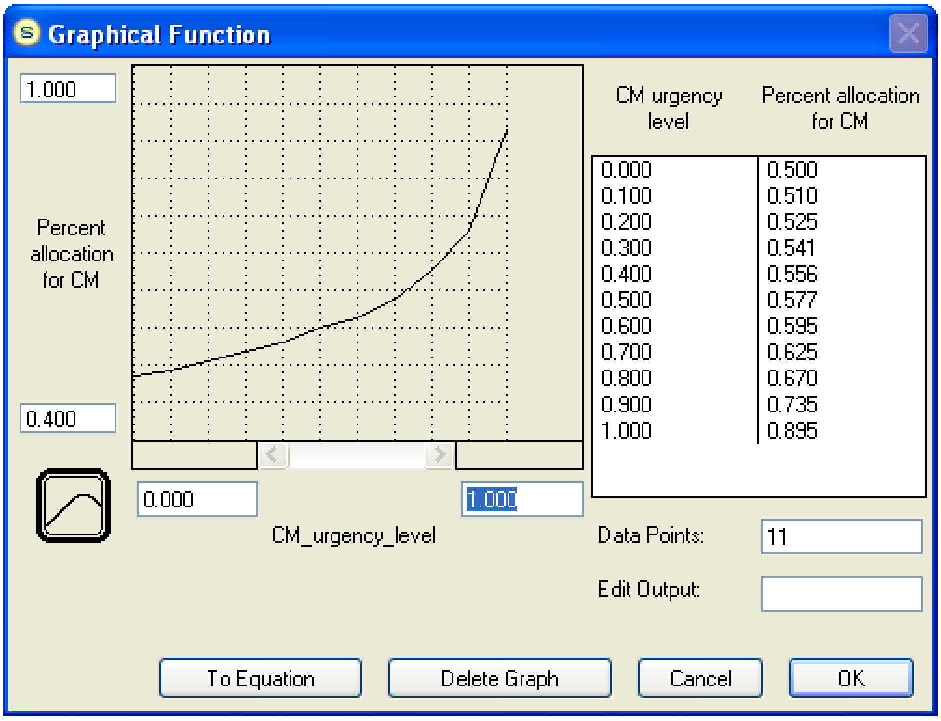


Figure 5.33 *The relationship between level of urgency of CM and percentage of staff allocated for CM*

5.4 Summary

This chapter has presented the development of a conceptual SD model of the eGSF relationships based on previous research and researcher's assumptions.

Prior to the development of the conceptual SD model, a system decomposition of the eGSFs was constructed. It provides a broad view and organises the myriad success factors into a manageable number of subsystems and sub-subsystems. This decomposition allows e-government stakeholders to focus on what kinds of factors should be considered in developing and sustaining e-government success in a particular situation, while still embracing a holistic view of the eGSF system. It is within this holistic perspective that the relationships of the eGSFs within e-government system SM and IS staff management are modelled.

The development of the conceptual SD model was begun by articulating the research problem in accordance with the SD modelling process. This included spelling out the model purpose, stating the dynamic of the problem, developing a reference mode and identifying key variables.

By referring to the articulation of the problem, a conceptual CLD was constructed. This diagram presents the relationships of the eGSFs in a qualitative way, providing insight into how the success factors of SM and IS staff management sub-subsystems relate to each other and lead to the achievement of e-government success. Reinforcing and balancing feedback-loop relationships can be identified from the diagram, showing the roles of success factors. This diagram also serves as the dynamic hypothesis.

Furthermore, an SD simulation model of the eGSF system has been developed based on the CLD. This model accommodates qualitative and quantitative factors of the eGSFs and makes the factors and their relationships explicitly and formally described. The model is organised into five sectors: maintenance process; motivational factors; staff competence; staff allocation; and staff development. To express the dynamic relationships between factors, a number of line graphs and simple mathematical equations were constructed based on the available and relevant previous research, as well as the researcher’s assumptions.

During model development, some SD validation processes have been conducted to ensure that the conceptual model does not produce implausible behaviour such as negative or indefinitely increasing value of delivery and request ratio, average competence level, productivity variation etc. It is also to verify that the equations dimension is consistent. Referring to the research design, this stage results in an SD model version 1.1, which can be used to construct questionnaires to collect data for a case study.

In order to build confidence in the soundness and usefulness of the model before it is used to assist e-government decision-makers or managers, a series of further validation processes still needs to be performed, as guided by Table 4.2 and Figure 4.5 of Chapter 4. This validation process includes an empirical model structure and parameters confirmation through a case study. Data collection and analysis of the case study are guided by the eGSF SD Model version 1.1 and will be presented in Chapter 6.

Chapter 6 Results and Realisation

6.1 Introduction

This chapter presents the realisation and results of data collection for a case study and SD model validation according to the method presented in Chapter 4. The case study data along with the validation process will be utilised to ensure that all elements and relationships of the model can be verified in an actual, successful e-government system in a developing country, and that significant elements and relationships of that system have been well represented in the model. The validation processes is required in order to be able to use the model confidently in understanding the importance of the relationships of the eGSFs and assisting e-government decision-makers and managers to achieve and maintain a successful e-government system. The following diagram, Figure 6.1, which represents a partial view of Figure 4.2 and a subset of the overall research design, provides the context of the case study and validation process presented in this chapter within the overall research design and indicates the order of presentation of this chapter.

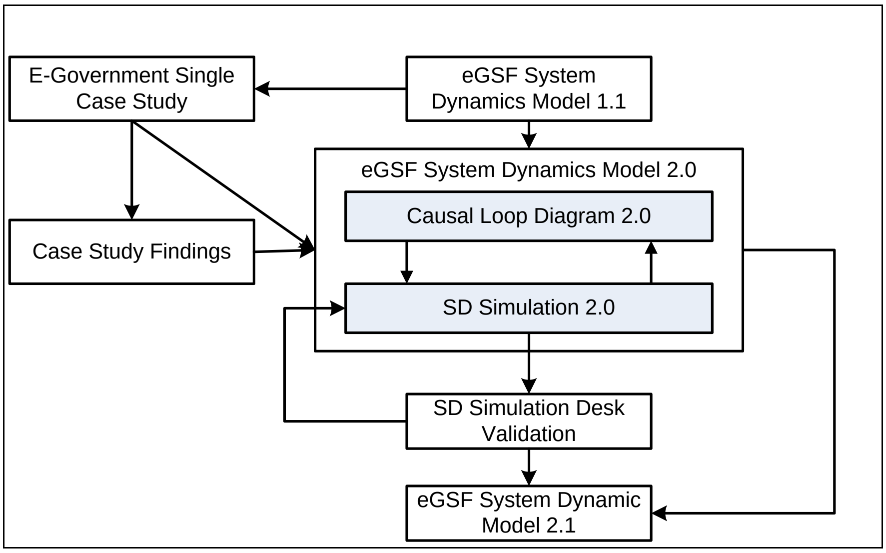


Figure 6.1 *A subset of the overall research design describing the context of the case study and validation process*

The model that will undergo the validation process is the SD model version 1.1 presented in Chapter 5. This version, which has undergone some desk validation processes, was used as a reference to develop questionnaires which were implemented to collect data from a selected case study.

A successful e-government system in Indonesia, a developing country, has been selected to represent successful e-government systems in developing countries. The data from this case is utilised to refine the causal loop diagram and validate the relationships structure and parameters of the SD simulation model version 1.1. Once these processes have been undertaken, further desk validation processes over the model are performed which eventually result in the SD model version 2.1.

The case study validation process and its results are now described. First, the process of case study selection and the description of the selected case will be presented. Second, the data collection process from the selected case will be explained. Third, a validation for the model version 1.1 which consists of eight steps will be undertaken and presented in accordance with Chapter 4, section 4.5.

6.2 Case study selection and description

6.2.1 Introduction

This section describes the process of the case study selection. It begins with preliminary research through the internet facility and ends with the choice of a case for study. The decision was based on the case's fulfilment of the criteria stipulated by this study and the availability of the top manager to participate in this research.

In this section, details of the selected case will be presented. These include the appropriateness of the case within the working definition of e-government success, e-government services provided by the case, IT infrastructure and application systems, SM management and IS staff management. The description is based on the data collected from the case.

6.2.2 Selection

Preliminary data collection about the case candidate. To select the case, a preliminary search was performed at the end of 2008 via the internet to obtain preliminary descriptions about the winners of the Indonesian eGovernment Award. The internet search was carried out again in early 2010 to obtain more current data about the award winners. A list of case candidates was compiled consisting of the Indonesian central government units which had won the award. Among the case candidates, the Ministry of Agriculture (MA) won the award for two consecutive years, 2008 and 2009. Further research indicated that the Ministry had also achieved a high ranking for the award in previous years. This preliminary search result hinted that the most likely case would be the e-government system of the Ministry of Agriculture (eGSMA).

Reasoning for eGSMA selection as the case candidate. The data resulting from the preliminary data collection about the eGSMA indicated that the stipulated criteria for the case candidate as mentioned in Chapter 4, section 4.5.3, were being fulfilled. This, along with the researcher's acquaintance with one of the senior managers of the unit managing the eGSMA, were the main reasons behind the decision to select the eGSMA as the case candidate. The contact with one of the senior managers within the Ministry was expected to smooth the researcher's communication with the top manager of the eGSMA.

Initial data collection about the selected case. Once the eGSMA was selected as the case candidate, further initial data gathering was conducted before the manager of the eGSMA was contacted to request the Ministry's availability for participation in this study. The initial data collected comprised a description of the e-government system, a detailed description of its official website, the services provided to external and internal users, the in-house developed application system and the management of the e-government system unit. The data was collected by browsing and downloading available documents from the eGSMA website.

This initial data collection concluded that the e-government system met the criteria for the selection of the case. In addition to the fact that the e-government system achieved high rank and won the award for several consecutive years, the unit that manages the e-government system also manages in-house its IT infrastructure, especially the application system software and ICT-skilled human resources.

The top manager consented to participate in the research. Once the selected case was considered appropriate, the researcher sent a letter to the top manager of the unit that manages the eGSMA, requesting their availability to participate in this study. The request received a positive response indicating their availability and willingness to participate in the study, which has been an invaluable support for this research. For further communication about the study, the top manager assigned an operational manager as the contact person. The contact person is an operational manager and senior staff member of the case, as the person has been in his career at the unit since the early 1980s. A copy of the response letter and its English translation are provided in Appendix A.1.

Case description. The home page appearance of the system website viewed in October 2010, the address of which is http://www.deptan.go.id, is depicted in Figure 6.2. The eGSMA is mostly managed by the Ministry's Information Systems Development unit, which is a sub-unit of the Centre of Agriculture Data and Information (PUSDATIN). The unit occupies the third echelon with respect to the Ministry's organisation and is one of three subunits of PUSDATIN; see Figure 6.3. However, during data collection the managers and the staff identified their unit as PUSDATIN; therefore, for the rest of the presentation the Information Systems Development unit will be called PUSDATIN.

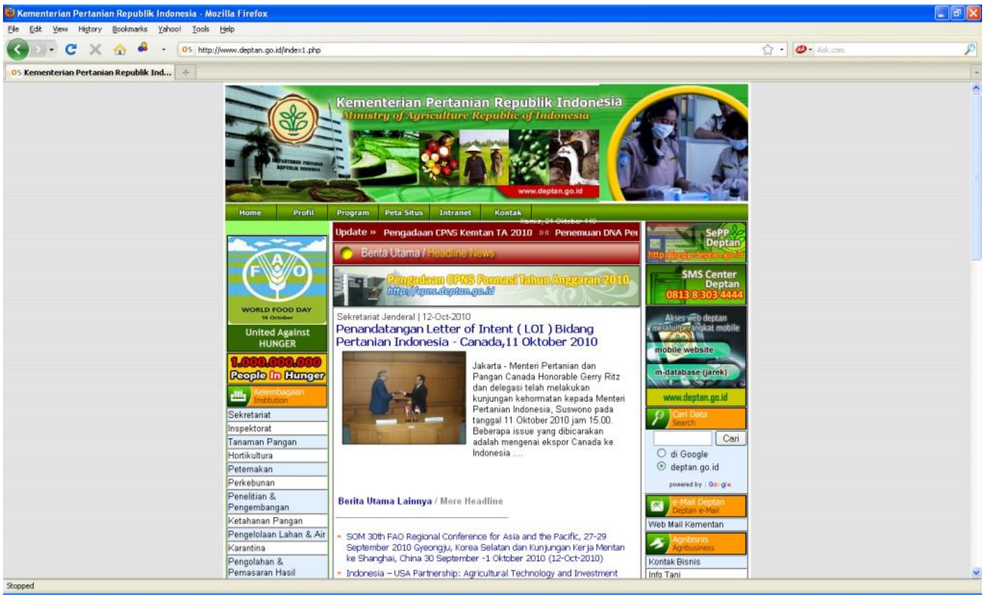
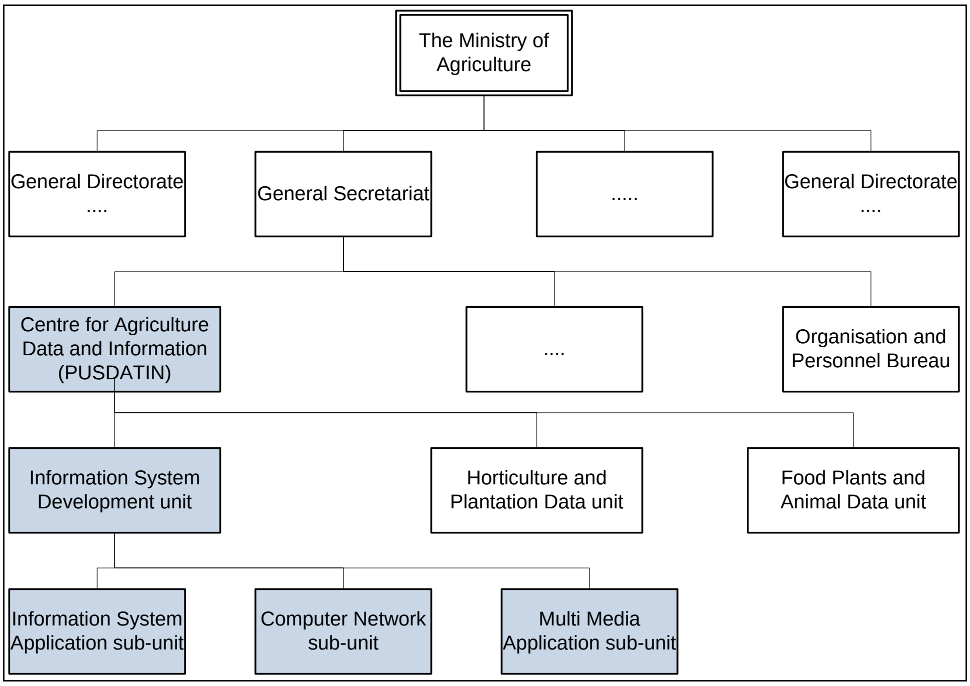


Figure 6.2 *The home page of the Ministry of Agriculture of the Republic of Indonesia website*

In the 1980s the eGSMA was established along with the PUSDATIN and in 1996 the eGSMA published its first website.



Source: *Processed from the eGSMA's website*

Figure 6.3 *Organisational chart of the Information Systems Development unit (PUSDATIN) within the Ministry of Agriculture*

6.2.3 Success

The eGSMA, as one of the Indonesian e-government systems, represents e-government systems in developing countries. According to the UN's e-government survey 2010 and the 2011 Waseda University World e-Government Ranking, the Indonesian e-government systems index level is within those of most developing countries. The eGSMA to some extent can be considered one of the e-government systems in developing countries that has achieved success. Therefore, other e-government systems from other developing countries or from other government organisations in Indonesia that are experiencing an inability to achieve success can learn from understanding the relationships between the success factors.

Concluded from the data collected at PUSDATIN, the eGSMA can satisfy the working definition of e-government success. The system satisfies at least the enhanced level of the UN's website index; has delivered e-government services for more than four years; and has an increasing number of application systems, types of services and users. Since its establishment, the eGSMA has been able to deliver services to its users, both internal and external, and has received several awards recognising its achievements. The awards and achievements are presented in Table 6.1. In addition, PUSDATIN has been able to maintain and upgrade its old eGSMA so that the system operates continuously and delivers services as intended. Even some new application systems have been developed and are being utilised to support new activities of the Ministry.

These achievements show that, in general, the eGSMA has been able to operate smoothly and maintain a particularly high level of service provision for a quite long period of time. Therefore it can be concluded that the eGSMA has been sustainably managed, and accordingly it satisfies the requirement as the selected case for the case study.

Table 6.1 The list of eGSMA achievements

No	Year	Award	Providers
1	1997	Second rank for Best website of government department	*Info Komputer* magazine
2	2000	Third position of Best web design	Bubu Award
3	2004	Third position of eGovernment Award, national level, Department category	*Warta Ekonomi* magazine
4	2005	The Runner-Up of eGovernment Award, national level, department category	*Warta Ekonomi* magazine
5	2006	The Finalist of eGovernment Award, national level, department category	*Warta Ekonomi* magazine
6	2008	The Winner of eGovernment Award, national level, department category	*Warta Ekonomi* magazine
7	2009	The Winner of eGovernment Award, national level, department category	*Warta Ekonomi* magazine

***Source**: Translated from the table of the awards received by PUSDATIN provided by the Multi-media Application sub-unit manager.*

6.2.4 Service provision

The eGSMA has been delivering services to both internal and external customers and a subset of current services it provides is presented in Table 6.2. In short, the services include support to the Ministry's internal activities, support to collaborative

Table 6.2 *Current services provided by the eGSMA*

No	Type of services	Application system name	
1	Support the Ministry in managing its human resources.	SIMPEG	Human Resources Application System
2	Support the Ministry in monitoring and evaluating the use of budget by its units and the finance performance of its units.	SIMONEV	Monitoring and Evaluation Application System
3	Support the Ministry in managing government-owned goods and equipment utilised by the Ministry.	SIMKAP / SABMN	State-owned Goods and Equipment Management Application System
3	Support the Ministry in easing and speeding up the flow of reports from all of its units throughout Indonesia.	e-Forms	Reporting Application System
4	Support the Ministry in monitoring and managing the Ministry's national program of rural agribusiness development.	SIM PUAP	Rural Agribusiness Development Application System
5	Provide land spatial data and information for agriculture-related stakeholders.	GIS	Land Use Spatial Data Information System
6	Provide downloadable information on agriculture-related government regulations.	BD Perundang-undangan	Government Regulation DBMS
7	Provide information on agriculture-related national data (commodity production, export and import of agriculture product).	BDSP and BD Eksim	Agriculture Statistics, and Export and Import Agricultural Product Statistic DBMS
8	Support the Ministry in providing a transparent and seamless process for its new civil servant recruitment.	SIM CPNS	Civil Servant Candidate Recruitment Application System
9	Support the Ministry in managing the Plants Variety Protection registration process.	SIM PVT	PVT Registration Application System
10	Support the Ministry in internal and external communication, collaboration and electronic-based transactions.	Other application systems	
11	Provide a variety of general information through a website.	The Ministry of Agriculture Website	

Source: *Processed from the data obtained from the interviews, document and website of the eGSMA* and coordinative processes with other ministries as part of a broader government obligation delivery, support to collaborative activities between the Ministry and its stakeholders such as farmers and agricultural companies, and provision of agriculture-related information services to the broader community. The data utilised to compile Table 6.2 was obtained from the interviews, documents and the website.

6.2.5 Infrastructure and application systems

The IT infrastructure of the eGSMA consists of its hardware, software, computer network and database management systems. The eGSMA information technology infrastructure data was obtained from the senior manager and/or the copy of PUSDATIN's document submitted to the Indonesian eGovernment Award 2009 organiser. The hardware operated by the eGSMA includes personal computers, laptops, printers and multi-media as well as various presentation tools which are distributed to all Ministry units. The software consists of various types, platforms and purposes including special-purpose application systems, general office application systems, multi-media systems and operating systems. The platforms of the software are Windows® and Linux. This software was acquired through purchasing, outsourcing and in-house development. Most of the applications systems developed and run by PUSDATIN are database management software. A local area computer network was developed in 1994 connecting all buildings and all levels at the central office of the Ministry, and the network and all associated components were upgraded in 2008. Currently, the network connects about 2,000 units of personal computers and computer peripherals all over the central office of the Ministry, using fibre-optic cable, to a set of consolidated servers in the PUSDATIN office building. There are two types of server operating 24 hours a day and seven days a week: Windows-based and Linux-based. In addition to hosting various application systems, the servers also function as the Ministry web-server. This computer network is fully managed by PUSDATIN.

To deliver the eGSMA services, PUSDATIN has been implementing and maintaining various special-purpose application systems, most of which were developed fully in-house, although some application systems developed by other ministries are being operated in the Ministry of Agriculture. These applications are part of a collaboration effort among all ministries of the Indonesian government, led by the Ministry of Finance. The first application of the eGSMA developed by PUSDATIN was SIMPEG, which was released in 1992, while the last was

SIMPUAP, which was released in 2008. Referring to the application system life cycle, those application systems are currently at the maintenance stage.

Currently, PUSDATIN focuses its improvement effort on the eGSMA website. This is intended to allow all of the applications to be run through the website, or at least to enable the users to access the output, and to facilitate the application updates as well as data upload through the eGSMA website. The implementation of various application systems, irrespective of whether the application is client-server based or internet based, has provided a range of highly positive benefits to the eGSMA stakeholders. For example, SIMPEG has provided an internal transparent process for staff promotion and improved staff management in the Ministry. SIMCPNS has also provided a very transparent process for new civil servant recruitment; in the past, this type of government process was suspected to be unfair and was claimed to be a source of corruption. SIMPUAP has been able to ease the monitoring process of a huge volume of tasks on the Ministry program. Table 6.3 is a list of application systems of the eGSMA built from the interviews, archival documents, documents provided by PUSDATIN staff or managers at the researcher's request, and the eGSMA website. Within the table, the complexity of specific application system data is presented based on the personal judgement of the interviewee who is responsible for managing that specific application. Table 6.3 indicates that the application systems were developed in an incremental manner, although there is no specific time interval for their development and release. The implementation of the application system was started with one or a small number of applications and followed by others in the ensuing years.

Table 6.3 *The software application systems of the eGSMA*

No	Application system	Users and software application description	
1	Name: SIMKAP/SABMN Released: > 5 years SLOC: not known Very complex considering the database size.	The data is input by the lowest level organisational units of the Ministry throughout Indonesia, and the information is used by higher-level managers of various units and the Equipment Bureau of the Ministry.	It was developed to manage state-owned goods and equipment within the Ministry. Currently it is used to tap and manage information for Ministry interests from the mandatory software from the Ministry of Finance. This client-server software evolved from DOS to Windows.
2	Name: SIMPEG Released: 1992 SLOC: not known	The users are all the Ministry's units at all levels that manage personnel throughout Indonesia.	This is client-server software used for an internal purpose. The software evolved from DOS to Windows to web-based.
3	Name: SIMONEV Released: 1995 SLOC: < 10000 Medium complexity. The complexity increases over time.	The users are the 2,300 Ministry units throughout Indonesia and the Ministry management at the central office.	It was developed to monitor and evaluate budget and finance performance of all Ministry units. Currently, it is utilised to tap and manage information for Ministry interests from the mandatory software from the Ministry of Finance. The software evolved from DOS to Windows.
4	Name: e-FORM Released: -- SLOC: not so complex	The users are all of the Ministry's units throughout Indonesia.	The software is relatively simple and used to smooth the organisational reporting process based on specific sectors such as plantation sector or food plants sector. The software evolved from DOS-based to Windows-based to web based.
5	Name: GIS Released: 2004 SLOC: a lot of modules Very complex due to type of data.	The users are various stakeholders (internal and external) of the Ministry who have interest in spatial data, especially land use.	The software is web-based and is published on the Ministry's website and can be accessed openly.

6	Name: SIMPUAP Released: 2008 SLOC: << 50000 Complex due to the large number of users.	The external users are the supervisors of a group of farmers throughout Indonesia. They input the data to the application system in a stand-alone software, then upload through the internet. The internal users are the PUAP management.	This is a web-based application system. The information can be accessed using this software at various levels of organisation.
7	Name: SIMCPNS Released: 2007 SLOC: about 10000	The users are civil service candidates all over Indonesia and the management of new civil servant recruitment within the Ministry.	This is a web-based application system. The application system is used only once a year for about three months.
8	Name: The Ministry of Agriculture website Released: 1996 Complex.	The users are general users.	This is the official website of the Ministry which houses all application systems of the eGSMA.
9	Name: BDSP Released: SLOC: not known. Not very complex.	The users are general users who need Indonesian agriculture data.	This is a web-based application system. The application evolved from a Windows-based to a web-based one.
10	Name: BDEksim	The users are general users who need export and import data on Indonesian agricultural commodities.	This is a web-based application system. The application evolved from a Windows-based to a web-based one.
11	Name: BDAbsensi	The users are the Personnel Management unit of the Ministry.	–
12	Name: BDPerundangan	The users are general users who need law and regulation documents related to Indonesian agriculture.	This is a web-based application system.
13	Name: SMS Centre	The users are general users.	This is a mobile-based application system.

Source: *Processed from the data obtained from the interviews, document and website of the eGSMA*

6.2.6 Software maintenance

Procedure for handling maintenance requests. The software application systems maintenance is carried out in-house mainly by the developers because most of the application systems are developed in-house by PUSDATIN staff. (For reasons of convenience and for consistency with previous chapters, software application systems will be stated as software; therefore software application system maintenance will be stated as software maintenance. For the purpose of managing MRs, PUSDATIN has recently developed a standard operating procedure (SOP) regarding SM. This SOP describes the steps that should be undertaken in response to an MR from users. The procedure includes type of activities; the staff who should conduct the activities; documents involved; input and output for each step; and time allocated for each step. As a general rule, once any request to enhance software is received it will be communicated to the "owner" of the application; for example, the owner of SIMPEG is the Personnel Bureau of the Ministry. Once the owner agrees, then the enhancement is carried out by the associated staff at PUSDATIN. However, the senior manager realised that the implementation of the SOP is not easy. The normal practice is that in the first instance the requester sends an official letter to request maintenance; but, on the second and further occasions, the "owner's" staff contact the PUSDATIN staff directly without any formal letter or other documentation. This is considered by PUSDATIN staff to be less complicated and makes the process easier.

There is no pre-specified priority for undertaking MRs. The requests will be accomplished on a first-come-first-served basis. However, the senior manager explained that this rule does not apply if the software is attacked by a computer virus that stops the application system from working, or the requester is a VIP at the Ministry. In fulfilling an MR, most problems that affect maintenance time are non-technical. In terms of technical matters, if the maintainer currently does not have skill related to the MR, the staff member will communicate with other staff or contact his/her colleagues outside PUSDATIN.

Current maintenance problems. The maintenance problem currently faced by PUSDATIN is in regard to the functional stability of other software developed by the Ministry of Finance. In this case, the Ministry of Agriculture has to operate the software by providing input data which will go to the Ministry of Finance. Actually, PUSDATIN has developed software with the same purpose as the one from the

Ministry of Finance, and the software has been operating in the Ministry of Agriculture for several years internally. In order to avoid double data entry, PUSDATIN modified its own software such that it is able to access the data from the Ministry of Finance software database. Because the Ministry of Finance software is not yet stable, PUSDATIN has to modify the internal software from time to time, normally every semester.

The maintainer. During the early times of PUSDATIN, SM was a team responsibility but because it is difficult for the team to sit together to discuss and perform maintenance, for the last five years it has become the responsibility of individual staff. The staff who maintain the software are those who developed or were involved in the development of the application system and understand the business process utilising that system. The staff who are responsible for a particular application system are mainly also assigned to support users in relation to the application system.

Software maintenance maturity. Overall, referring to the proposed Software Maintenance Maturity Model of April et al. (2005), the SM management of the eGSMA is still at level 1. The MRs are mostly managed in an informal manner, and the relationships between maintainer and the requester are built personally. A standardised maintenance activities record has not been established. Maintenance services are based on individual initiatives. The recent effort to build formal procedures has moved the software management slightly towards a higher level, but much management effort needs to be made.

6.2.7 Information systems staff management

Staff categories. The staff at PUSDATIN are categorised into two different organisational positions: functional and structural. The functional staff have a functional position and are assigned a specific type of job based on their particular competence. The staff's jobs do not include day-to-day managerial and administrative activities. The improvement of functional staff levels depends on staff achievements in specific knowledge and skills relating to their jobs, such as software development and maintenance; relevant training attended and completed; performance in user support and training; publication of job-related articles and other relevant criteria. Functional staff may improve their rank in two years instead of the normal four years, and theoretically they can achieve the highest level more easily than structural staff. On the other hand, structural staff have to deal with day-to-day managerial and administrative jobs, which require more generic knowledge and skill. To improve by one level, structural staff need at least four years but they may rise automatically as long as they do not display morally unacceptable behaviour or an extremely poor achievement level. However, structural staff cannot go beyond a particular level, which is based on their education and managerial position.

Functional staff jobs. In PUSDATIN, each functional staff member is assigned to one application system as their main responsibility. Their main job is to develop and maintain the software assigned to them and to coordinate user support and training, as well as other activities related to their assigned software. Nonetheless, a type of software can be assigned to more than one staff member and one staff member can be assigned to more than one type of software. In addition to the staff's main responsibility for dealing with specific software, staff can also be assigned a job as a support member for other software.

Staff sufficiency. The number of staff with respect to the PUSDATIN workload is regarded by the senior manager as sufficient from the perspective of the senior manager's experience and observation, although there has not been any study of staff sufficiency. The staff are classified as senior and junior. Most of the senior staff do not have a formal background in information technology. They began their career as high school graduates, but then the Ministry sent them to Gadjah Mada University for a six-month intensive computer programming course. In contrast, the junior staff mostly have a degree in information technology. In many cases, new staff have more current knowledge and skills in IT.

6.2.8 Summary about the case

Some specific conclusions can be drawn from the case study findings in regard to the eGSMA itself:

- the eGSMA is a successful e-government system in the sense that it has been able to ensure the required software availability level over several consecutive years
- all types of software to deliver e-government services are relatively small in size, about 10,000 SLOC or less
- on average, the software performs relatively simple and specific functions

- the software maintainers are also the developers of the software they maintain
- software maintenance requests level is decreasing over time
- maturity level of the software maintenance management is still at level 1
- the number of IS staff is sufficient.

6.3 Case study data collection

6.3.1 Introduction

This section describes the questionnaire design, interview process and collected data. The questionnaire design subsection explains the types of questionnaires used for different interviewees and the different stages of the interviews. The interview process subsection presents the process of interviewing from interview planning up to the interview activities. The collected data subsection provides description about the types of data, data validity of possible bias, and the possible threat to the model from the data validity.

6.3.2 Questionnaire design

The questionnaires. Data was collected through open-ended semi-structured interviews and from available documentation. For the interviews, three different questionnaires were prepared and the conceptual causal loop diagram and the initial model (SD model version 1.1) were used to prepare them. One questionnaire addressed a broad overview of the e-government system and was used to interview the senior manager (see Appendix A.4). Another questionnaire, lengthier than the first, was used to obtain data from the operational managers (see Appendix A.5). The second questionnaire covered all the elements and relationships within the initial models. The third questionnaire was used to interview senior staff and focused on the application system maintenance activity, motivational factors and competence (see Appendix A.6). All data collection processes were conducted in the Indonesian language because all interviewees, document providers and interviewers are native Indonesian speakers.

6.3.3 Interviews

Interviewee selection planning. The preliminary data obtained from the eGSMA website indicated that the e-government system is managed by one senior manager, three operational managers and nine application system management staff. Based on this preliminary data, it was planned to request all 13 for their availability and participation for the data collection. The senior manager was expected to provide an overall description of the e-government system and the implemented policies in relation to the focus of this study. He was also expected to provide answers for some additional research questions specially designed for collecting data from the senior manager. The operational managers were expected to provide data regarding the success factors and their relationships being investigated by this study, based on their experience in managing the operation of the e-government system. The senior staff involved in the e-government system operation were expected to give data based on their experience and views in running the e-government system, especially in relation to the SM and motivational-related factors. It must be noted that the document downloaded from the website of the selected case did not contain the names of the staff, but it did mention the names of the senior manager and the operational managers.

The actual selection of the interviewees. Despite planning to interview nine staff, the actual selection of senior staff was based on suggestions and permission granted by the senior manager and the contact person, because they had sound knowledge of the staff performance. For this, the senior manager and the contact person were informed that the staff who would be interviewed should currently maintain an application system software.

Prior to the commencement of the data collection, the researcher met with the contact person to discuss the selection of senior staff for interview, the arrangement of the interview process and time allocation. The contact person also introduced the interviewer to the senior manager, other operational managers and the senior staff who would be interviewed.

The interviewees. The actual interviewees available to participate in the research were the senior manager, two operational managers and ten staff who manage software. However, four staff cannot be considered senior in terms of their length of time in SM jobs. Despite their unsuitability, these staff provided some valuable data for further understanding the SM process. One operational manager was not available to participate because of their limited time availability. Another operational manager also considered that he/she was not an appropriate person for the interview because his/her jobs and his/her staff do not deal with software.

A summary describing some characteristics of the interviewees is provided by Table 6.4.

Table 6.4 *Characteristics of the interviewees*

Position	∑	**Education level**	∑	**Year Experience**	∑	**Job**	∑
Senior Manager	1	Masters	6	More than 5 years	10	Manajerial	3
Operational Manager	2	Bachelor	6	Less than 5 years	2	Software programming	10
Staff	10	High School	1	Not stated	1		

∑: the number of interviewees

Interview preparation. Prior to the beginning of the data collection, all documents for the data collection were handed to the contact person: the information about the study, Appendix A.2; the letter of consent form, Appendix A.3 and all three versions of the questionnaire (Appendix A.4-6). It was expected that the contact person would have general understanding of the overall data collection process and context.

Before an interview began, the interviewee was provided with general information about the research and the voluntary and confidential nature of the interview, and was asked to sign the consent letter about their participation in the interview and the use of a recording device during the interview. The interviewee was also informed that they could refuse to answer any question and could withdraw their participation at any time without any consequence to them. In addition, a concise overview of the study, some technical explanations and definitions related to SM were provided by the interviewer before the interview proper began (Appendix A.7). The interviews were conducted based on each interviewee’s time availability and were conducted within the government unit office.

Interviews with the senior manager. The interviews with the senior manager were undertaken three times on three different days by the researcher. The first interview was based on the prepared questionnaire, which was given to the manager one day prior to the interview; the second was based on the prepared CLD; and the third referred to the initial model. The CLD and the initial model were given to the senior manager several days before the interview. The questionnaire and CLD are reproduced in Appendix A.4 and A.8 respectively.

The first interview with the senior manager was undertaken in the manager's office and took about two hours. In the second interview, the causal loop diagram was presented to him to elicit his opinion of the model in term of confirming or disconfirming model elements, relationships and the overall model based on his experience in managing the e-government system. The model was presented and explained in a gradual manner by the interviewer. The model presentation began with a subset of the model that contains four loops, then continued with an additional loop that contains other elements and relationships. For each gradually added loop, the senior manager was asked to confirm or disconfirm the model based on his experience in managing the e-government system unit. The third interview used the initial model as a referent. In this third interview, the initial model was explained to the manager in a broad manner which covered the elements of the model, the flows and the reason for such model construction. The explanation did not include the detail of the initial values setting and mathematical formulations. The senior manager was then asked to give comments on the model, and verify the model elements and flows which were related mainly to the management of the e-government system under the manager's responsibility. The manager was asked if the model reflects general practice in the unit. Especially, referring to the specific sector which models the maintenance competence development of the staff, the manager was requested to evaluate the staff based on their maintenance competence level and category.

Interviews with operational managers. The interviews with the operational managers took about 2.5 hours each and were based simply on the questionnaire. The interviews were also undertaken by the researcher. During the interviews, some questions needed further clarification and elaboration through question probing, as the meaning of the questions could not be fully understood by the interviewee or there was a disagreement between the interviewee and the interviewer in understanding some terminologies. In regard to unavailability of one operational manager, the data associated with the questionnaire for this manager was obtained from other operational managers and the senior staff, especially a senior staff member responsible for coordinating the functional staff.

Interviews with senior staff. In spite of the planning that the interviews with the software staff would be conducted only once, the actual interviews were undertaken in two stages by the researcher. The first stage was based on the prepared questionnaire and the second stage used a specifically designed questionnaire related to the prepared questionnaire and the initial model. The second stage was conducted because the documents about SM activities from which the quantitative data for the model were to be derived were not available. Also, the interviewees could not provide a quantity or describe a relationship pattern as required by some specific questions. The second-stage questionnaire was devised based on the initial data obtained from the literature review used in the initial model. At the second stage, interviews were carried out with only some of those who were interviewed at the first stage, based on their level of seniority.

At the first stage, after two interviews the questionnaire was modified slightly to include some necessary questions. The modified questionnaire was then used to interview the remaining interviewees and each interview took around one hour. During the interview, some probing questions were used to clarify the meaning and understanding of some terminologies.

At the second stage, the interviewees were presented with a set of data, a table and a set of relationship graphs which were derived from previous research, and were asked to describe the same issue from their own experiences, views or perspectives. Before the interviewee gave their responses, the interviewer explained the context and the meaning of the presented data, table and relationship graphs, then asked the interviewee to describe a number or draw their own graph on prepared paper. In some interviews, as the interviewee could not describe exactly or felt unsure, the interviewer helped them draw the graphs and asked whether they agreed or not. Once they agreed, they were asked to sign the paper containing those graphs. This kind of questionnaire and an example of the associated responses is presented in Appendix C.1 and C.2.

6.3.4 Case study data

Various types of data were obtained from the case study that include qualitative and quantitative data. Some qualitative and quantitative types of data were obtained from the interview process. Data was also obtained from hard and soft documents supplied by PUSDATIN staff, as well as from observing the software run by the interviewees. The data was also collected in various forms. Verbal, tabular and graphical responses from the interviews, archival documents, software samples and observational data based on on-screen demonstrations were collected and processed.

This nature of data collected and method adopted by this study does not enable this research to perform data overview as normally conducted in a study that implements statistical methods before running a statistical analysis. However, a sufficient overview for relevant pieces of collected data is provided during SD validation process; then the data is used to validate or refine the initial SD model (e.g. Georgantzas 2003).

There is every likelihood that the responses from interviewees contain bias. The sources of possible bias are that many of the questions are about evaluating the interviewees' own performance (absences per year, personal activities during working hours); the responses are based on the interviewees' memory or the ability to recall what happened in the past; there was possible misunderstanding in interpreting terminologies although some technical explanation was provided (for example, the notion of software error/fault: it is possible that an interface design that does not match with user requirements, although it is working, may be considered by the interviewee as a software fault). For these possible biases it was necessary to reduce their possible effects. The circumstances of having a number of interviewees with similar characteristics, conducting the interviews in an isolated room and observing directly the software and the SM performed on the software were expected to lessen the negative impact of these biases. The use of an isolated room is expected to allow the interviewees to express their own views and experiences. The use of graphical questionnaires was also expected to help the interviewees to describe a particular pattern with better approximation and to lessen misunderstanding.

6.4 Model validation

6.4.1 Introduction

Following the research design presented in Chapter 4, subsection 4.3.3, especially Figure 4.2, the subset of which is also described in section 6.1, once case study data is obtained, the next step is to use the data to validate the model. This section presents CLD refinement and simulation model validation of the SD model. Data collected from the case study will be used to refine the CLD and validate the model presented in Chapter 5.

The CLD refinement will be based mainly on the interview with the senior manager, corroborated by understanding the data gathered from other interviewees, and indicated by the validated simulation model. This refinement is implemented to the CLD previously described in Figure 5.6. Presentation of this subsection will begin with this CLD refinement process.

For the model, the validation follows the process described in Chapter 4, section 4.5. As described in Figure 4.5, it begins with verifying the structure and parameters of each sector of the model and is followed by testing the model resulting from this verification. The verification will be based on analysing and understanding the interview data; and this process is organised according to the model sectors. During this verification process, it will be indicated whether or not the success factors and their relationships as modelled in Chapter 5 are confirmed by the data collected from the case study. Furthermore, the data obtained from the case will be accommodated to replace the initial values or relationship structures that were previously used in the model for simulation instantiation. Practically, this process will verify the structure and parameters of the model described in:

- Figure 5.11 – Maintenance process sector,
- Figure 5.16 – Motivational factors sector,
- Figure 5.23 – Staff competence sector,
- Figure 5.29 – Staff development sector,
- Figure 5.31 – Staff allocation sector

Eventually, this verification will conclude with verified sectors, which in turn forms a verified simulation model. Referring to the overall research design, this verified model is the eGSF SD model version 2.0.

Once the model version 2.0 is obtained, the overall research design indicates that further validation steps need to be carried out. Adhering to the steps described in Figure 4.5, these consist of dimensional consistency, boundary adequacy, extreme condition, behaviour sensitivity, behaviour reproduction and behaviour anomaly tests, respectively. Unlike the structure and parameters verification which mainly based on the case study data, this set of tests is performed using specific simulation modelling software. These steps eventually result in a validated SD model or the eGSF SD model version 2.1.

In the following subsections which presents the structure and parameters verification, to identify the source of the data, the interviewee is coded according to their sequential number in the interviewee table presented in Appendix B.1. The number is enclosed within parentheses.

6.4.2 Causal loop diagram

In accordance with the overall research design, the subset of which is described in Figure 6.1, the case study data will be used, firstly, to refine the CLD developed in Chapter 5, subsection 5.3.3.

The findings indicate that, in general, the CLD is confirmed, except that there are some success factors and relationships that need to be added or deleted. First, the senior manager did not confirm the *experienced staff turnover* factor as this specific factor relating to human resources management does not exist at PUSDATIN. The senior manager stated that PUSDATIN has never experienced any staff turnover. Further, with respect to SM, the senior manager classifies IS staff into less competent and competent categories. Second, even though the senior manager confirmed the existence of the e-champion factor, there is no indication from his interview data or from other interviews that the e-champions are a separate group of staff. It can be inferred from the interviews that the e-champions are the staff who increase effort if the maintenance performance is less than expected. In other words, the e-champion factor has been accommodated in the performance-effort relationship. Additionally, the senior manager and other interviewees mentioned that they communicated with other staff or colleagues, as well as with users, in conducting SM. The IS staff interviewed also stated that an increasing level of experience gained from doing maintenance make their maintenance performance better. Last, the senior manager acknowledged that a task from VIPs of the Ministry should be performed first and currently conducted maintenance tasks put aside, and staff are also assigned other duties, which is quite significant. The existence of these types of tasks was confirmed by other interviewees.

To accommodate the findings, the CLD needs to be adjusted accordingly. First, the *experienced staff turnover* factor is deleted, and, consequently, the *fraction of inexperienced and experienced staff* and the *total staff* are only influenced by the incoming *new staff* factor. To accommodate the classification suggested by the senior manager, the fraction is now named as the *fraction of less competent and competent staff.* Second, the e-champion related factors are also eliminated: *goals of e-champion, need for achievement of e-champion* and *e-champion effort*. Also, *communication with experienced staff and users* factor is added as positively affecting the *delivery level,* and referring to the simulation model, this *delivery level* is linked to the *software availability level* with positive polarity. The *learning and experience level* factor is now linked, with negative polarity, to the *recurrent maintenance.* And finally, a new factor called *VIP request and other duties* is added to the diagram, influencing the *total actual effort* with negative polarity. As the *VIP request and other duties* increases, the effort dedicated to the application system maintenance will decrease. It is assumed that the *VIP request* tasks do not require all staff to devote their effort to complete these tasks. One of the operational managers said that normally the tasks are not technically demanding and the senior manager will arrange the staff such that not all staff are allocated to the *VIP request* tasks. The final CLD is presented in Figure 6.4.

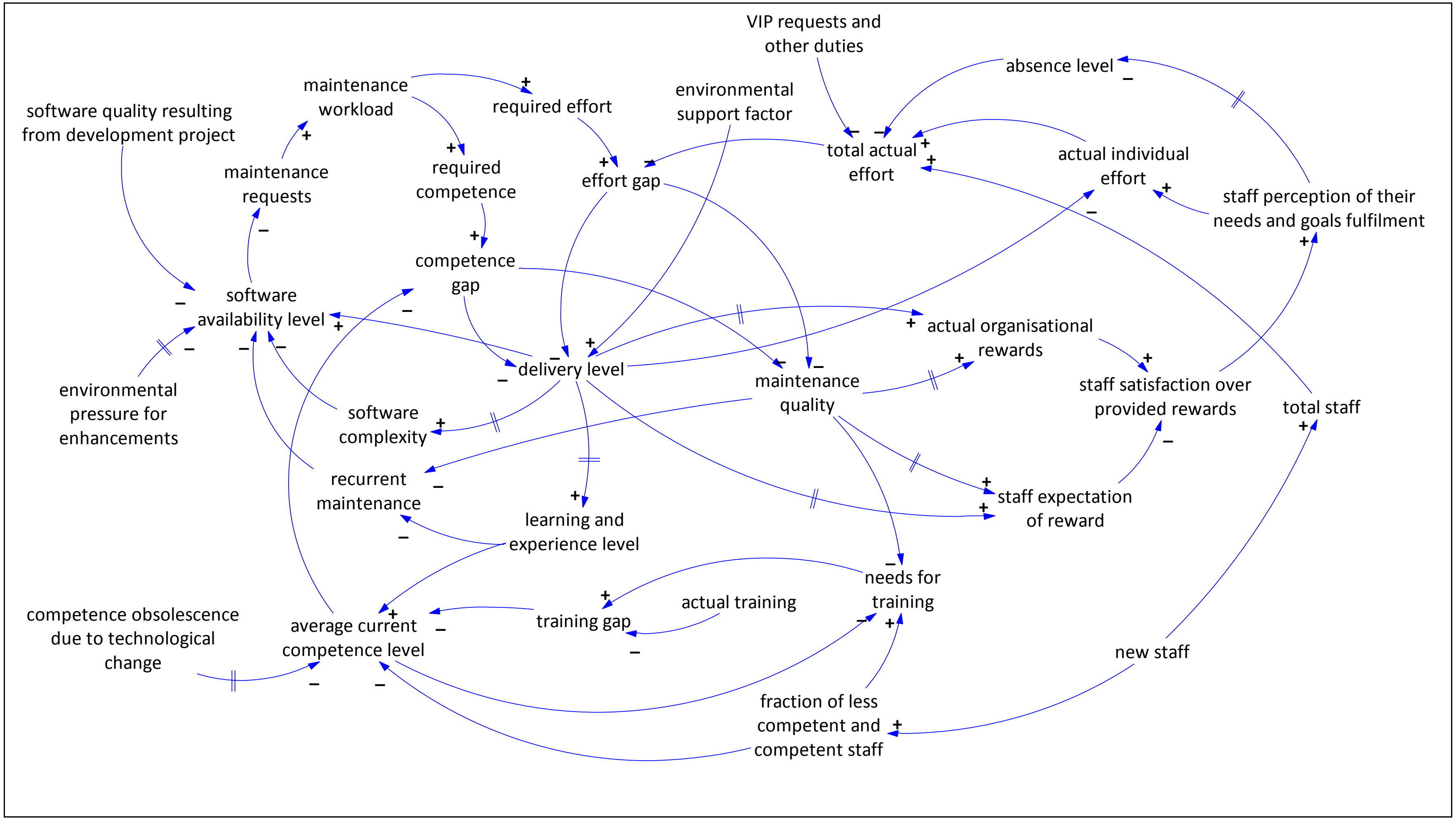


***Figure 6.4** The causal loop diagram of the eGSF System Dynamics model version 2.0*

6.4.3 Structure and parameter verification – Maintenance process sector

In this subsection, the structure and parameters of the model in the Maintenance process sector, Figure 5.11, will be verified against the case study data in accordance with validation process presented in section 4.5. On completion of this verification process, a verified Maintenance process sector will be presented at the end of this subsection.

Incoming MRs. Section 6.2.6 described that there are incoming maintenance requests which are treated on a first-come-first-served basis and are put in "batch of request" or stock using a first-come-first-served rule. The number of MRs as well as their arrival time intervals cannot be predicted in advance as they vary over time. Once received, each request is then classified as a CM or an EM task. This confirms the structure of the model that represents incoming MRs.

System size or complexity and reported error. No interviewees could provide the exact size of the application system for which they are responsible, especially the SLOC (see Table 6.3), although they develop and maintain the system. A senior staff member indicated that all application systems managed by PUSDATIN are less than 100,000 SLOC (4). The complexity of the application systems was described as varying from not-so-complex (2)(4) to very complex (6)(10). Some interviewees considered that their applications are not complex because they both develop and maintain the applications, therefore the applications size does not affect maintenance. Other interviewees considered that their application systems are complex because the systems involve a large number of modules and database tables or dynamic data. One interviewee observed that PUSDATIN has never developed software so complicated that it cannot be managed by one staff member.

From this description, the size and complexity of the software cannot be concluded with high-level approximation. As a simple estimate, this study assumes that the average size of the software is 10,000 SLOC. From Table 6.3 it is known that much of the software was developed before 2005 (historical time horizon begins), and at least three types of software were released after 2005. Because there are 13 different types of software managed internally, 10 of them released prior to 2005, then the total size of the software of the eGSMA maintained by PUSDATIN at the beginning of the simulation is 100,000 SLOC, followed by three releases occurring once a year with 10,000 SLOC each. These values are used to substitute the initial value for the *System_size* stock and to construct a graphical relationship describing the software release which is used in the *Releases_factor,* see Figure 6.5.

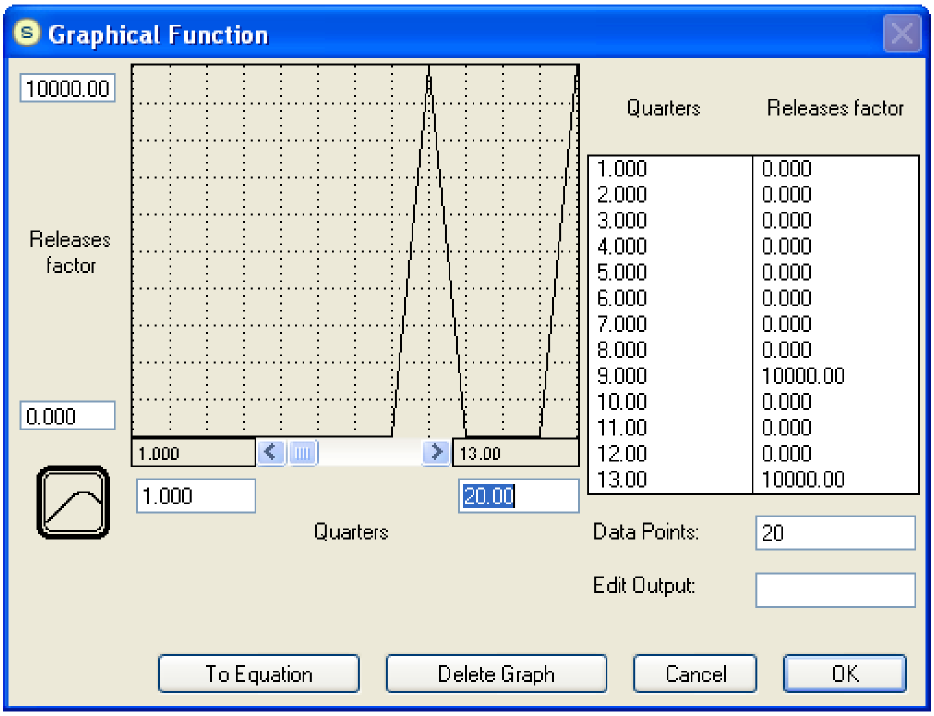


Figure 6.5 *The eGSMA software releases over time*

The interviewees agreed that the software so large in size is more prone to fault; however, they could not describe exactly the association between size and faults. They also agreed with the graph provided by the interviewer which described the relationship between system size and the number of faults (see Appendix C.2.1), but because they considered the software of the eGSMA to be small, only the small left-side part of the graph was considered appropriate. The relationship between the system size (in SLOC) and the number of faults, as well as structural relationships between the number of faults and new MRs provided in section 5.3.5, will therefore be kept in place.

System size and maintenance performance. The interviewees could not describe the number of lines of code that were deleted, modified, or added during maintenance. It was estimated around 5 to 10 SLOC, but it could be more for EM

(2). From this case study data, the system increase is set to *15* SLOC per maintenance task. This value is used to input the *System_increase_factor*.

Some of the interviewees agreed that the more complex the software, the more time is needed to complete a maintenance task; this is because the time required for analysis is increased (5)(8)(7)(10). However, some interviewees who stated that maintenance is not affected by the size of the software, it only very slightly influences the completion of maintenance tasks (4), and it is staff competence that determines the completion of the maintenance task (2). Further, after a graph representing the relationship between system size and maintenance performance that was used for the simulation instantiation, as depicted by Figure 5.22 in Chapter 5, was shown to the interviewees, they did not confirm the graph. Therefore, based on this data, the *Impact_of_system_size _on_productivity* is eliminated from the initial SD simulation model.

Time-series behaviour of MRs. It was also revealed by the interviewees that, once a type of software enters the maintenance stage, the CM number is higher than the EM. However, after a period of time the number of CM is decreasing towards zero and the EM out-numbers the CM. But, unlike the corrective, the number of enhancive MRs never reaches zero. For example, some types of software, such as SIMPEG, until recently still required EM. The enhancive MRs are mainly due to policy or regulation change, technology development and/or users' need, as indicated by previous literature. Appendix C.3 and C.4 give examples of the most current enhancive MR for SIMPEG (12) and for SIMPUAP (13).

In general, the MR level is low in the first term, high in the second term, then decreases and leads to zero for the rest of the application system life (8)(7), Appendix C2.2. In addition, the corrective MR level increases much higher than the average level soon after an improvement to the software is released (5).

Although the data from the case study does not provide a specific and detailed pattern of the time-series MR number except a very broad pattern description, the general pattern of the MRs is in agreement with that observed and reported by Burch and Kung (1997). Therefore, the time-series pattern of the MRs depicted in Figure 5.13 will be kept in place, but with a minor modification. The reason for this modification is to accommodate the fact that many of the eGSMA software have been released prior to 2005, therefore they have been in operation more than 4 quarters; there were three new releases in the 9th , 13th, and 17th quarters and the enhancement requests have never been zero. Based on the researcher's own estimation, the percentage of MRs at the first and second quarters of Figure 5.13 is reduced by 25% and this percentage is added to the 10th, 14th, and 18th quarter. This modification results in Figure 6.6.

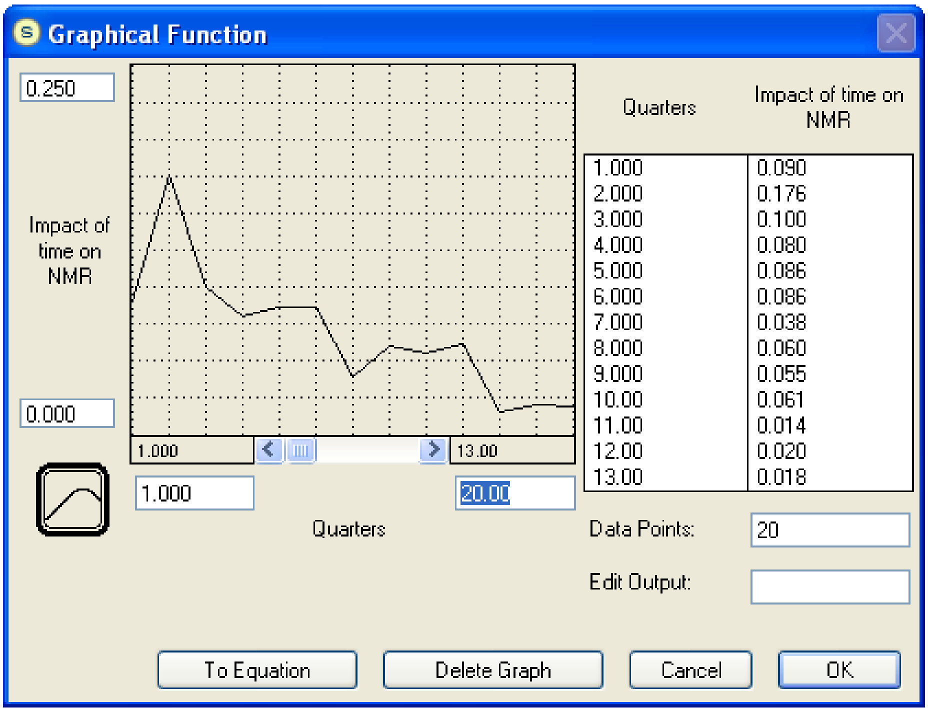


Figure 6.6 *The modified time series of the new MRs pattern*

Classification of MRs. From the interviews with the staff using both the questionnaire and the tabulated previous research description (see Appendix C.2.3), it can be concluded that the percentage of the CM is less than the EM. On average, the corrective MRs account for around 33% while the enhancive MRs account for about 67% of the maintenance, as shown in Table 6.5. This is, in general, in agreement with those found in many previous studies.

Table 6.5 *Percentage of corrective and enhancive MRs*

Interviewee #	Corrective	Enhancive
2	40%	60%
4	30%	70%
5	20%	80%
6	<40%	>60%
7	30%	70%
8	40%	60%

In general, CM is given higher priority than EM (1) (2). This is because if both type of maintenance happen at the same time, a software fault that requires CM causes disturbance to the software operation, while the enhancive MRs can still wait. This case study data is in accordance with that observed by previous studies. In the model, this is represented by the setting of the *Correction_request_ rate = Maintenance_request*Corrective_percentage* and the *Enhancement_request_rate = Maintenance_request*(1–Corrective_percentage)*. Later, in the Staff allocation subsection, this higher priority is reflected by a higher staff allocation for the CM.

As the specific and detailed pattern of the time series of the percentage of the MRs number is also not available from the case study, the *Corrective_percentage* is set in such a way that the average percentage of the CM is more or less 33%, and general time series behaviour of the CM proportion described by interviewees, presented in the time-series behaviour of MRs section above, is adopted. It can be noted that there are some spikes in the percentage of the CM after each release. This setting is presented in Figure 6.7 and used to input the *Corrective_percentage*.

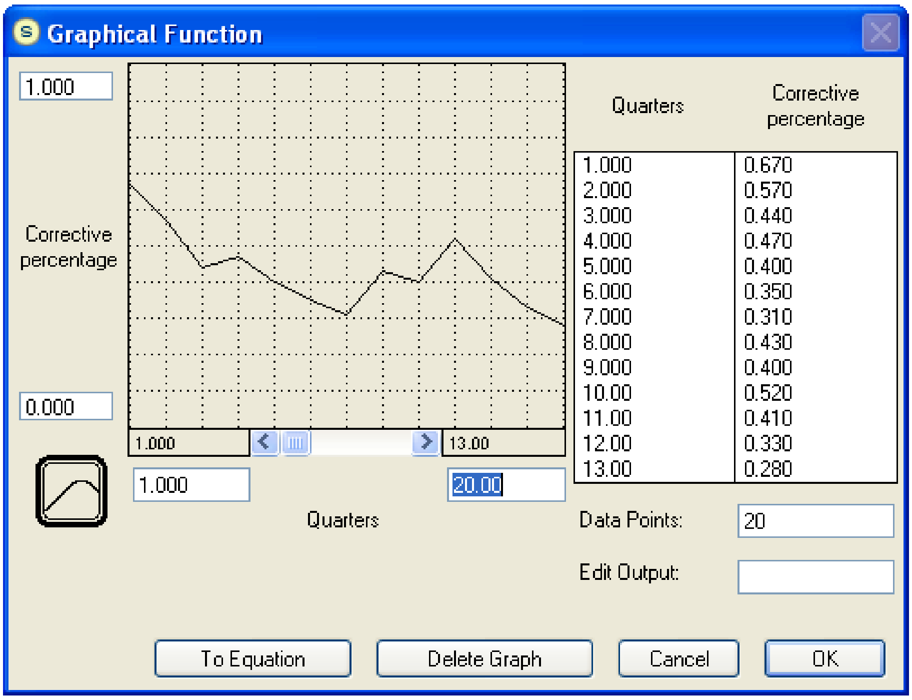


Figure 6.7 *The refined quarterly percentage of corrective MRs*

Maintenance process steps. The general maintenance activities, which can be classified into three iterative steps – analysis, modification development and delivery, as shown by previous literature (see Figure 3.2) – were confirmed by interviewees. They also confirmed the diagrams describing the steps of the maintenance in practice, which describes CM as one step and EM as three steps (see Appendix C.2.5), considering the time and resources allocated for SM activities. Therefore, the structure of the maintenance process, the flow and steps in the initial SD simulation, is kept in place.

Corrective maintenance steps. The case study data has shown that a CM task requires only one or two weeks (5)(4)(7), or even, in some cases, may take less than one day (9). This confirms that CM can be considered as consisting of only one step.

Based on this data, it can be assumed quite reasonably that the average time taken to complete and deliver a CM task by one staff member is one week or five working days. This will be used as a reference to set the *CM_delivery_rate* value. Especially, the value of the *CM_factor* is set in such a way that, on average, there are 13 CM tasks that can be completed and delivered per quarter. This becomes the initial value of *CM_productivity_variation.* Because this variation is a function of staff *Productivity_variation,* which initially depends on the initial value of the staff *Average_competence_level* (equals to 2.6475), and the *CM_factor*, therefore the *CM_factor* is set to 4.9.

Enhancive maintenance steps. All interviewees revealed that all EM tasks require significantly more time than the CM ones, although this also depends on the complexity of the request and the staff competence (2). On average, the time needed for EM is around two months but it certainly varies. This time length is mostly for communication with various associated stakeholders or for the analysis step. This confirms the findings of previous studies in terms of the comparison of the time taken for performing CM and EM maintenance tasks.

By using this finding, the value of the EM tasks rate is set in such a way that the rate is *1.5* tasks per quarter, or the average time taken to complete and deliver one EM task by one staff is about *43* working days (two months). So, the initial value for *EM_productivity_variation* is *1.5* tasks per quarter. Because this variation is a function of staff *Productivity_variation,* which initially depends on the initial value of the staff *Average_competence_level* (equals to 2.6475), and the *EM_factor*, therefore the *EM_factor* is set to 0.567.

Further, the value of the staff *EM_productivity_variation* along with the *EM_analysis_factor, EM_development_factor,* and *EM_delivery_factor* values will be used as input for the rates of enhancement analysis, development and delivery. To set each of these multipliers it is assumed that, following SM literature, the enhancement analysis, development and delivery take 50%, 30% and 20% respectively of the enhancement duration. In other words, one EM task takes 21.5 working days for the analysis, 12.9 working days for the modification development and 8.6 working days for the delivery. For each EM step, these working days are equivalent to: *EM_analysis_rate*=3.02, *EM_development_rate*=5.04 and *EM_delivery_rate*=7.56 completed tasks per quarter respectively. Because these rates depend on the staff *EM_productivity_variation,* to get the values associated with each rate the *EM_analysis_factor, EM_development_factor* and *EM_delivery_factor* are set to 2.02, 3.34 and 5.04 respectively. For example, 3.02 for the *EM_analysis_rate* is more or less 1.5*2.02.

The stock's name for each step of the EM is *EM_analysed, EM_developed* and *EM_delivered.*

Recurrent faults. An interviewee (2) mentioned that it is estimated that 10% of MRs cannot be fulfilled, but another interviewee (4) said that all the maintenance tasks assigned to him can be solved. There are also repeated maintenance requests, which are around 5% (2) to 10% (8) of the total MRs. In some cases, the same requests were addressed by more than two users at different times. This is because PUSDATIN is unable to reach the large number of users who are spread throughout Indonesia when a maintenance task is complete (5).

No interviewees could specify whether a request was actually a result of previously completed maintenance or not. In this case, this study assumes that those MRs that cannot be fulfilled are recurrent faults, and therefore the percentage of recurrent faults is assumed to be 10% of the total completed maintenance tasks. Based on this assumption, on average the rate of recurrent faults is 0.5% per quarter (note that the simulation time is 20 quarters). This value is used to set the initial value of the graphical relationships *Recurrent_faults_percentage.*

The interviewees (4)(5)(6)(7)(8), however, confirmed that as they gained more experience through self-learning and training, the level of recurrent faults, which represent SM quality, decreased. However, the decrease does not lead towards zero, although they could not provide the figure. Figure 6.8 is the *Recurrent_fault_percentage* which depends on the competence level gained from self-learning and training. The figure is reproduced from the graph pattern created by an interviewee (see Appendix C.2.4), but the percentage is set according to the above data.

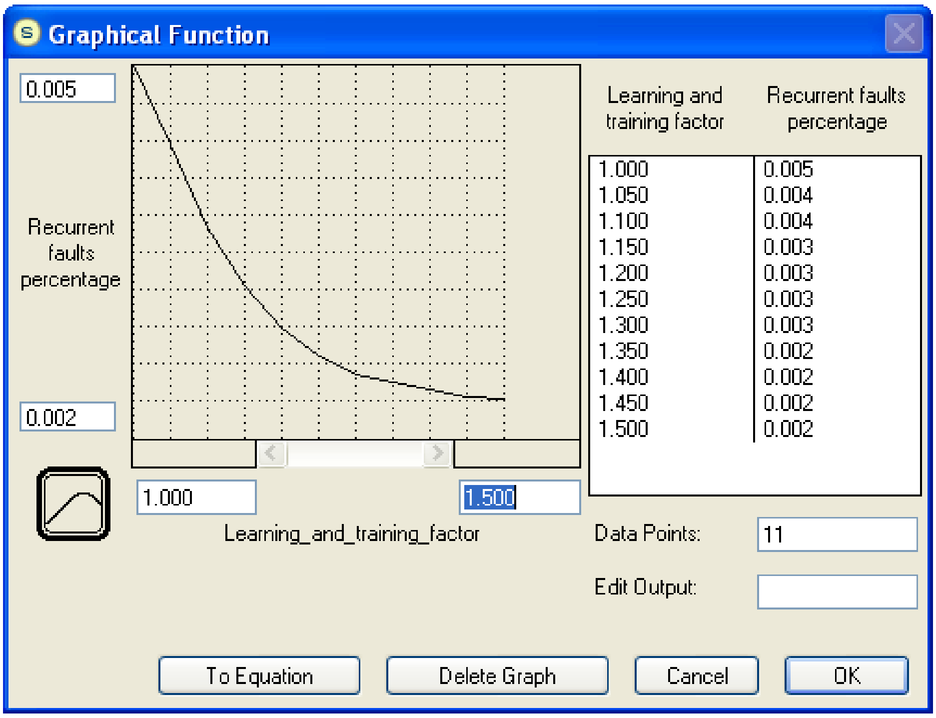


Figure 6.8 *The relationships between the learning and training and recurrent fault percentage*

Furthermore, it is assumed that all recurrent faults are transformed into MRs. Accordingly, the *Recurrent_fault_back_variation* is equal to the rate of the recurrent faults. In turn, this requires the *Recurrent_fault_semaphore_rate* to be set to 1.

The aforementioned descriptions in this section refine and validate the structure and parameter of the SD simulation such that the refined structure and parameters of the relationships represent the important factors and relationships of the SM process of the eGSMA. This is represented by the Maintenance process sector in Figure 6.9.

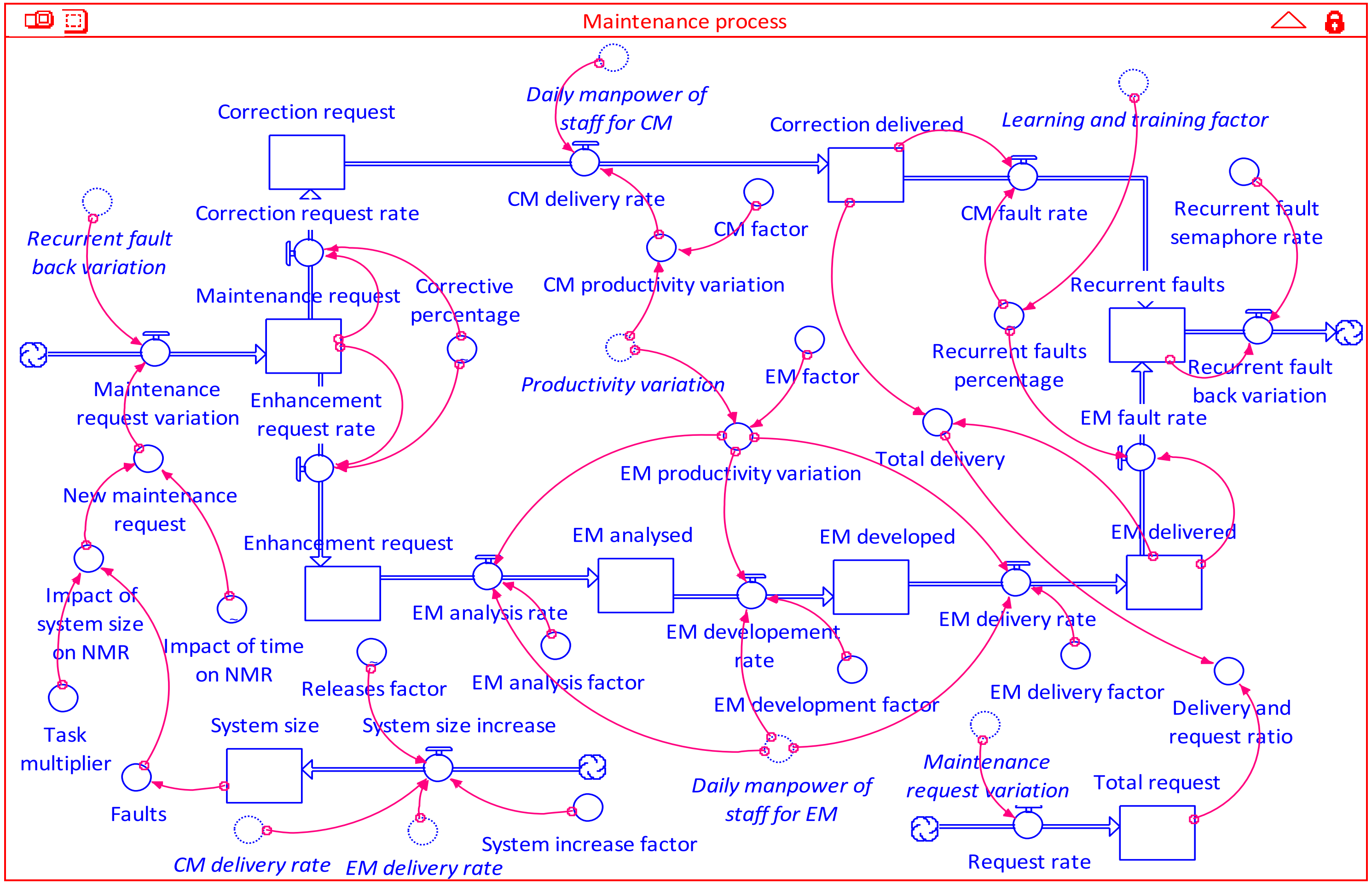


Figure 6.9 ***The Maintenance process sector of the eGSF SD simulation version 2.0 of the eGSMA***

6.4.4 Structure and parameter verification – Motivational factor sector

This subsection presents verification process of the structure and parameters of the model in the Motivational factor sector, Figure 5.16, against the case study data. This verification is undertaken in accordance with validation process presented in section 4.5. On completion of this verification process, a verified Motivational factor process sector will be presented at the end of this subsection.

Organisational rewards. Regarding organisational rewards, the senior manager explained that the financial reward for civil servants in PUSDATIN is based on the standard stipulated by government regulation, so PUSDATIN cannot go beyond that. However, the senior manager can manage the PUSDATIN budget such that organisational rewards can be provided to the staff. More honoraria are provided to those who perform better. Training based on staff interest is also part of rewards, especially overseas conferences or training. In addition to these, PUSDATIN has been trying to assign jobs based on staff competence and interest. For example, a tutor position for user training will be given to highly motivated staff (2) and this job would eventually result in additional income. As previously indicated, the organisational rewards are additional to the staff monthly salary.

The following graph, Figure 6.10, is a reconstruction of the original graph drawn by the senior manager that describes the relationship between the staff maintenance performance (abscissa) and provided organisational rewards (ordinate) (see Appendix C.2.7). The maintenance performance represents the number of maintenance jobs that can be completed and delivered successfully by a staff member per quarter; while the ordinate is a scaled value of the total rewards provided by PUSDATIN to its staff per quarter. The senior manager explained that the rewards provided to the staff who perform better will ultimately improve staff performance (reinforcing feedback); but the rewards eventually level off at a particular level. In addition to the performance in SM, the staff are also evaluated based on their ability to communicate to users the application under their responsibility, scientific papers written and the ability to organise a team.

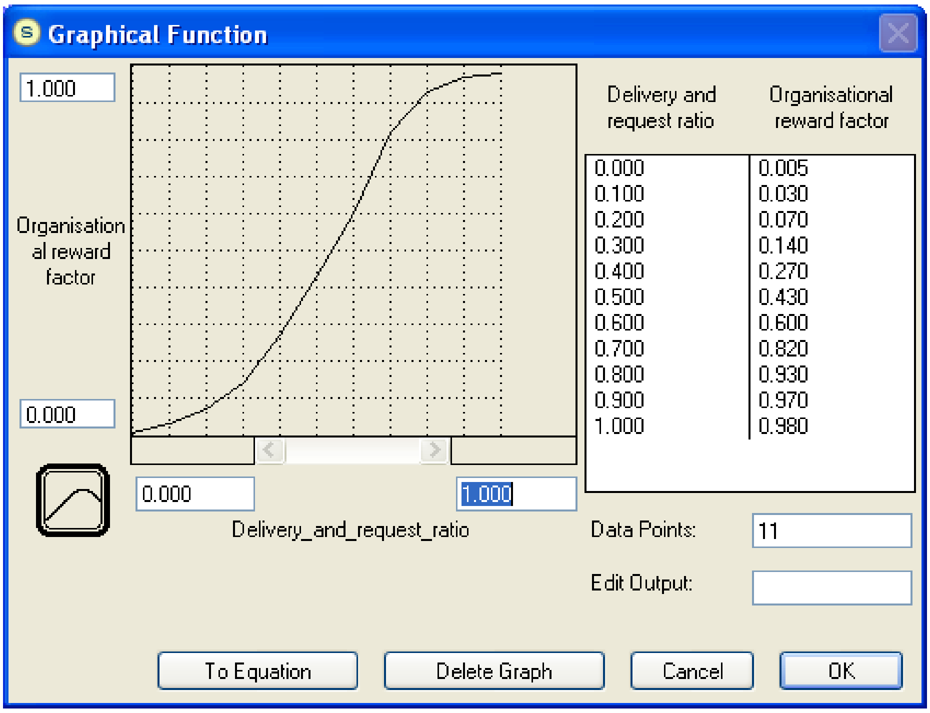


Figure 6.10 *The case study relationship between staff performance and organisational rewards*

The graph depicted in Figure 6.10 is used to set the *Organisational_reward_factor.* The rate of reward provided by the organisation is modelled by the *Reward_variation* which is formulated as *Organisational_reward_factor / Reward_adjustment_time.* It is assumed that the performance in a particular quarter will get the rewards within a stipulated amount at the same time as the staff member achieves a particular level of performance. Therefore, the *rwrd adj time* is set to *1*.

Rewards received from the organisation. In viewing the organisational rewards, the interviewees had diverse opinions. One interviewee wondered whether he had received organisational rewards or not (10), while another interviewee considered he had received an extraordinary reward regarding his contribution to the unit (4). However, all respondents said that they are proud if the applications for which they are responsible are really utilised by users and provide benefits to them. Some organisational activities such as training for staff and user training are also viewed as rewards, because the training gives them credit points which can eventually be used to improve their employment level. The user training, especially to regional areas, gives staff a chance to travel (7) which in turn provides additional income to them (6)(7). As user training was mostly assigned to the staff with better performance, the user training is considered as a reward (1)(4)(6)(7). Being known by high-level managers from other units or organisations is also considered as a reward (10). Scholarships to pursue a higher academic degree are another invaluable reward (4). Also, most of the interviewees mentioned explicitly that working in the IT area that they are keen on is in itself a reward.

The responses from the interviewees indicate that the SM staff has received rewards from the organisation and they acknowledged that the rewards consist of intrinsic and extrinsic ones.

The expected rewards. Some of the interviewees did not give their view regarding the expected rewards (4)(5). Other interviewees were aware of the consequences of being civil servants, so they do not expect more financial reward because it has been stipulated by government regulation (7)(10). However, they still expect that their application is useful and provides benefits to the users.

In response to the interview using a preset graph, an interviewee (7) described his/her reward expectation with respect to their performance as described by the following reconstructed diagram, Figure 6.11. The original diagram drawn by the interviewee is shown in Appendix C.2.8. Within the graph, the staff performance which is described as the abscissa represents the number of maintenance tasks that can be completed successfully by a staff member per quarter, while the ordinate is a scaled value of the total rewards expected by the staff member per quarter. In terms of the initial reward, an interviewee expected from PUSDATIN that when the staff performance is zero then the rewards given should also be zero, because he currently observed that in some cases although the staff performance was zero they still received certain rewards (8).

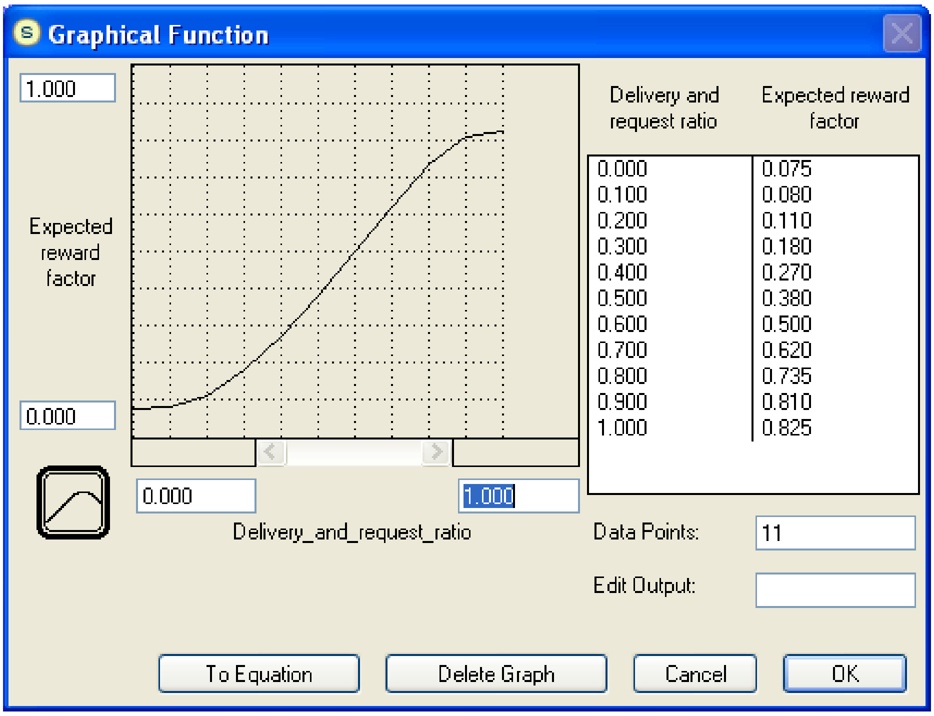


Figure 6.11 *The case study relationship between staff performance and staff expectation on reward*

This diagram is used to set the *Expected_reward_factor*. The rate of reward expected from the organisation is modelled by the *Reward_expectation _variation* which is formulated as *Expected_reward_factor / Reward_expectation _adjusment_time.* It is assumed that the staff expect organisational rewards to be given to the staff at the expected level based on performance in the quarter when they perform. Therefore, the *Reward_expectation_adjusment_time* is set to 1.

Comparison between the received and the expected rewards. Among the interviewees, there were staff who consider the rewards are less than expected (10), about the same as expected (7)(6)(8) and much more than expected (4). As in the initial SD simulation, the difference between the accumulation of the provided organisational rewards, which is kept in the *Reward,* and the accumulation of the expected rewards, which is kept in the *Expectation_of_Reward,* is calculated in the *Reward_performance_difference*.

Impact of the rewards on working hours. In responding to the question about the impact of the rewards on working hours, all the interviewees felt that the level of rewards received does not reduce their base maintenance working hours. While some of them acknowledged that the received rewards can motivate them to work better (4)(6)(7), other interviewees explained that the rewards do not influence their working hours (5)(10)(2) because they already receive a fixed income. All of the interviewees, who were asked to draw a graph describing the relationship between the rewards performance difference and efforts (working hours), agreed with the graph drawn by (7) as depicted in Appendix C.2.9. This, therefore, replaces the initial relationship graph shown by Figure 5.20.

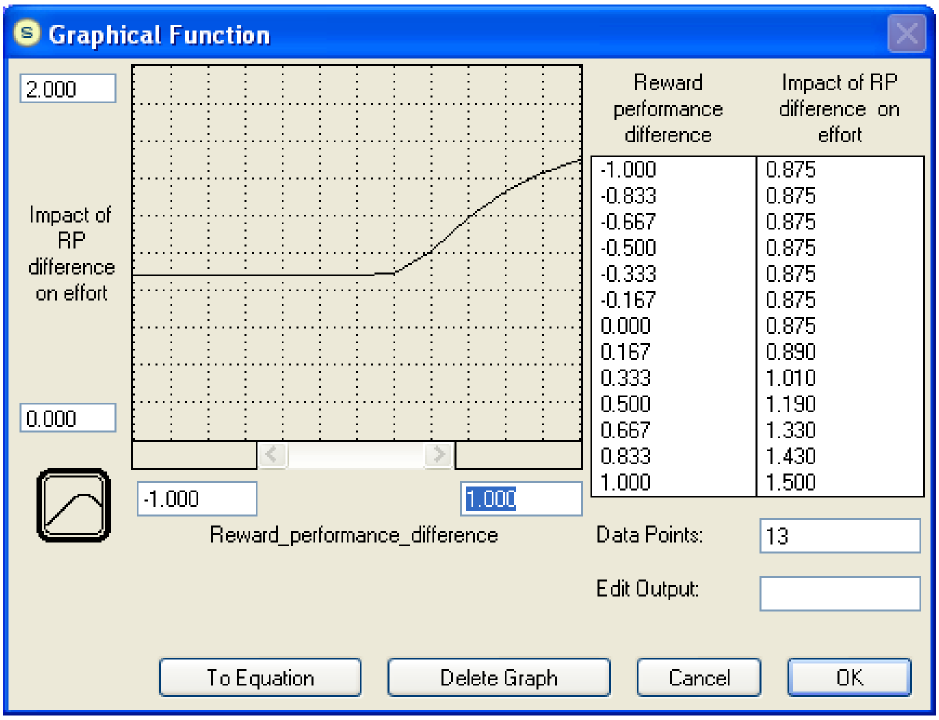


Figure 6.12 *The case study relationship between reward performance difference and effort*

The reconstructed graph is presented in Figure 6.12, which describes the relationship between the rewards and efforts in the eGSMA. The ordinate of the graph represents the difference between the rewards provided by the organisation and the rewards expected by the staff; both refer to the maintenance performance, while the abscissa represents the effort exerted by the staff as an impact of the difference. The interpretation of the abscissa is the same as that in the initial model. The graph indicates that if the difference (ordinate) is positive, then the effort can be improved up to 50% (7) or 30% (4) of the normal working hours; however, if the difference is negative, then the effort is not reduced. If the difference does not exist, the staff will exert according to their normal working hours, that is, 0.875 or 87.5% of 8 official working hours a day. This equals to *Base_effort* value. Figure 6.12 indicates that the maximum effort exerted by the staff is 1.5 or 150% of 8 working hours a day. One interviewee indicated that he reduces slightly the working hours if the reward is less than the performance. The relationship graph is used to set the *Impact_of_RP_ difference_on_Effort.*

Impact of effort and performance difference on working hours. All interviewees mentioned that they bring their job home if they cannot finish it during office hours. In addition, there are many occasions when they are required to work overtime by PUSDATIN. In this case, they are not fussy because they know the consequences of being a civil servant (2). For the overtime work required by PUSDATIN, the staff will be provided with all the necessary resources and honoraria (2), so they will add to their working hours if they consider their performance will miss the target time. On the other hand, all interviewees stated that if they can finish sooner than the scheduled time, then they do not reduce their effort and will report to their manager as the job is complete. In response to a graph presented to them, the interviewees described the relationship between the effort performance difference and working hours as depicted by the graph in Appendix C.2.6.

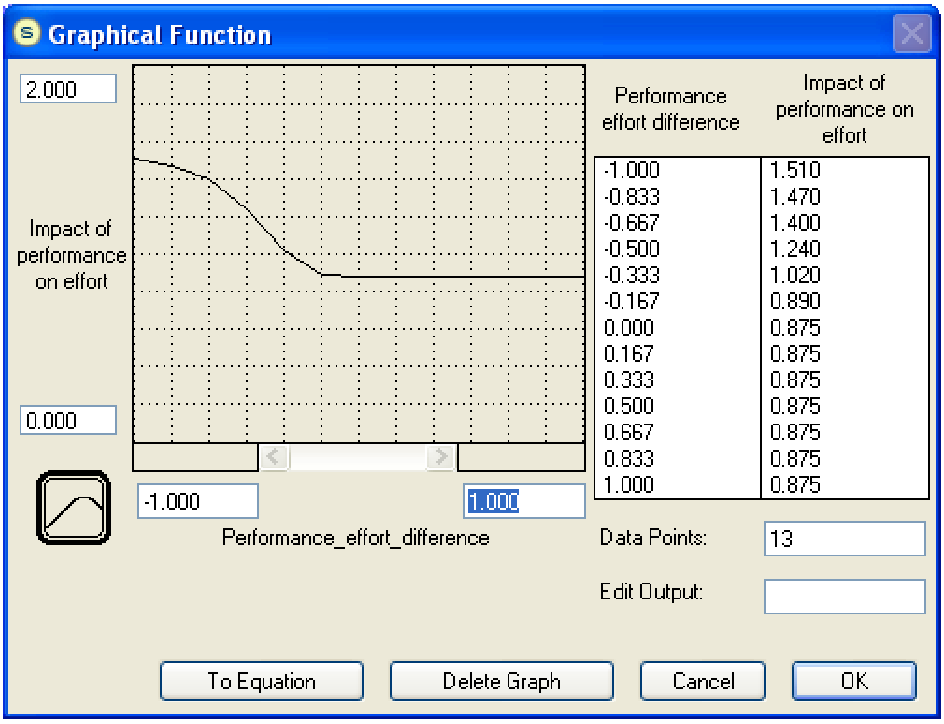


Figure 6.13 *The case study relationship between performance effort difference and effort*

Figure 6.13 is the reconstruction graph of the original graph provided by the interviewee (7) who described the relationship between performance and effort.

The ordinate of the graph represents the difference between performance and effort, while the abscissa represents the effort exerted by the staff considering the difference between performance and effort. The interpretation of the abscissa is as in the initial model. When the difference is zero, in the sense that a normal effort results in on-time completion of a maintenance task, the ordinate value is set to 0.875 or 87.5% of 8 (official) working hours a day. This equals the *Base_effort* value. This is also the value when the staff can finish a maintenance task faster than scheduled or the difference is positive. On the other hand, if the performance is estimated to miss the schedule or the difference is negative, then the staff will add their working hours up to 50% of the working hours. The graph provided in Figure 6.13 is used to set the *Impact_of_Performance_on_Effort*.

The e-champion. The senior manager mentioned the existence of a number of staff who can resolve many serious problems when the other staff cannot. However, this is not a special group of staff but they are among those categorised as competent staff. Accordingly, the e-champion does not need to be modelled separately. It has been accommodated by the *Impact_of_ Performance_on_Effort*. Thus, the *Impact_of_Echampion_on_Effort* is eliminated from the model, which is also eliminated from the CLD.

Communication with other staff and users. The staff communicated with other staff in order to solve a problem (4)(6)(8), to understand the problem (10) or to get an idea of how to solve the problem (5)(7). The interviewees said that the communication level is good and easy. The communication with other staff helps the staff to maintain the software, but there is no exact figure about the impact of this communication towards maintenance performance. Quite often the communication took place in an informal way (5). The senior manager explained that in terms of technical matters, if a staff member as the maintainer currently does not have a skill related to an MR, they will communicate to other staff or contact their colleagues outside PUSDATIN.

The interviewees felt that communication with users also affects maintenance completion time and especially that the number of users negatively influences maintenance performance (4)(6)(7). This is so because the number of users affects the number of requests; and the more requests the more difficult they are to prioritise (8). In addition, the clarity of users' requests also affects maintenance completion time (10) although they could not give the percentage of the influence. As there is no specific data from the case study, the data used in the initial simulation model is kept in place. From the description of these factors, it can also be concluded that dependence on communication increases the length of completion time or reduces productivity.

Personal activities during working hours. All the interviewees acknowledged that they did some personal activities during working hours. The time used for these activities varied from 5% (10) to 20% (2). On the other hand, they mentioned that it is quite normal that they still think about the job while being at home or even bring the job home. The average of these personal activities times is used to determine the base effort. The average is 12.5%. Therefore, the *Base_effort* is set to *0.875*.

The case study data has been presented and used to validate and refine the motivational factors sector of the initial SD simulation, therefore this sector corresponds to the significant structure and parameters of the motivational factors and their relationships of the eGSMA. The validated and refined version of this motivational factors sector is presented in Figure 6.14.

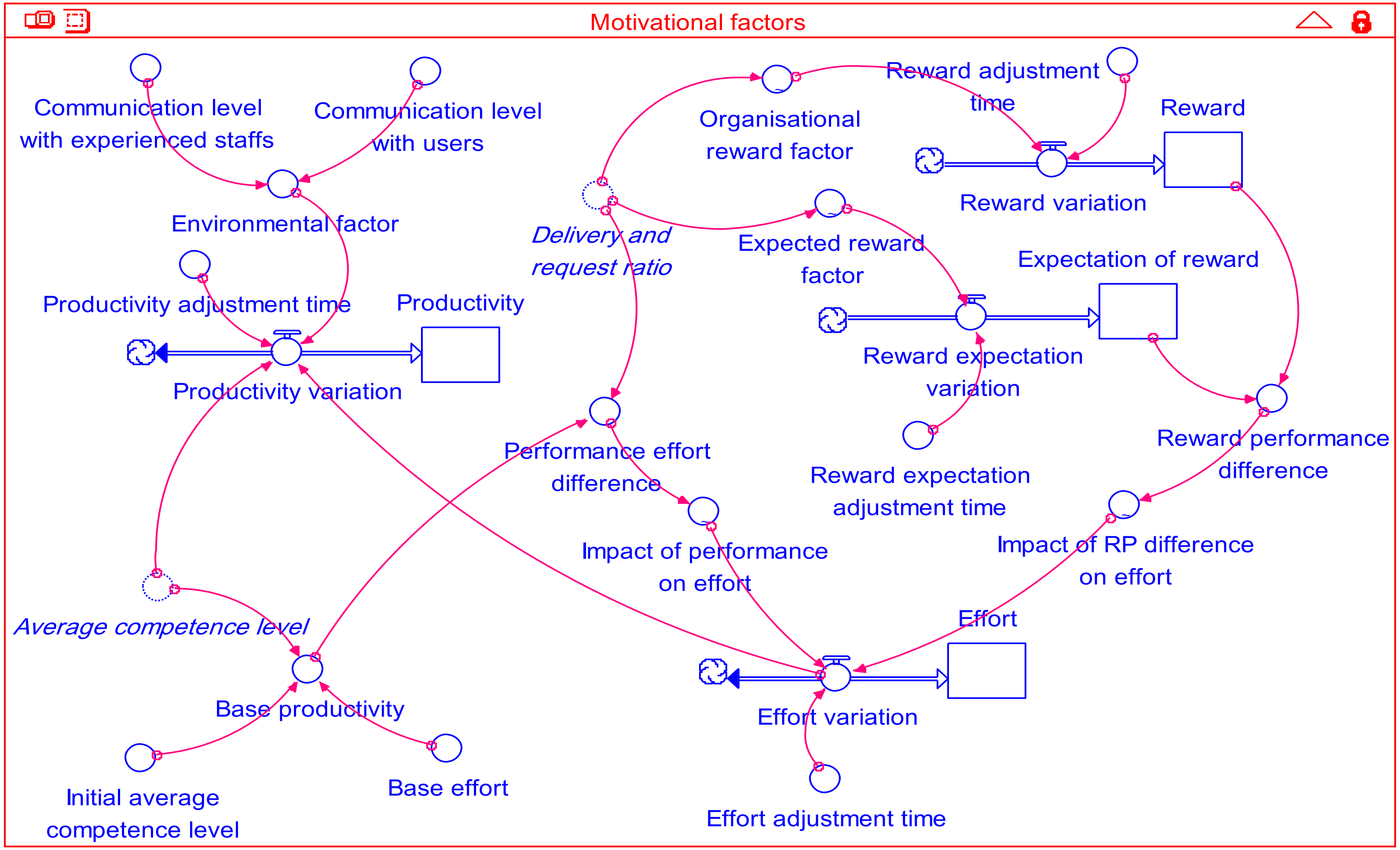


Figure 6.14 *The Motivational factor sector of the eGSF SD simulation version 2.0 of the eGSMA*

6.4.5 Structure and parameter verification – Staff competence sector

In this subsection, structure and parameters of the model in the Staff competence sector, Figure 5.23, will be verified against the case study data in accordance with validation process presented in section 4.5. On completion of this verification process, a verified Staff competence sector will be presented at the end of this subsection.

Staff competence level. In regard to the competence of the staff in SM, the senior manager stated that there are only two categories: competent and less competent staff, denoted as *Competent_staff* and *Less_competent_staff* stocks, while the competence levels are represented as *Competence_level_of_CS* and *Competence_level_of_LCS*. The senior manager expressed the comparison between these two levels of competence as “70 versus 30”. This can be interpreted as, within a specific time interval with the same number of jobs, the competent staff can finish 70% of the jobs while the less competent can complete only 30%. This data along with the average time required to complete a CM as well as an EM task is used to approximate the value of the levels of competence of competent and less competent staff. As mentioned in section 6.4.3, on average a staff member can complete 13 CM and 1.5 EM tasks in a quarter. Considering the percentages of corrective and enhancive MRs which are 33% and 67%, respectively, the average number of tasks that can be completed by a staff member in a quarter is 0.33*13 + 0.67*1.5 = 5.295 tasks. From this value, the level of competence of competent staff is 0.7*5.295 = 3.7065 tasks per quarter, while the level of competence of less competent staff is 0.3*5.295 = 1.5885 tasks per quarter. Therefore, the *Competence_level_of_CS* is set to *3.7065*, and the *Competence_level_of_LCS* is set to *1.5885*.

Impact of maintenance experience on level of competence. SM experience improves staff competence and confidence through self-learning and the improvement is realised in terms of the time required to complete maintenance tasks: the time becomes shorter (5)(6)(7)(8)(10). However, the increase in the competence eventually levels off because the maintenance problems are typical (2)(4). The improvement varies from 10% to 70% or attains 40% on average.

The graph describing the relationships between the number of maintenance tasks delivered and competence level improvement by self-learning was confirmed by the interviewees, although the maximum value may be lower, as in Figure 6.15. This figure is reconstructed from the interviewees' responses as described in Appendix C.2.10, but the maximum improvement value is taken from the above data. This relationship graph is used to set the *Learning_factor*.

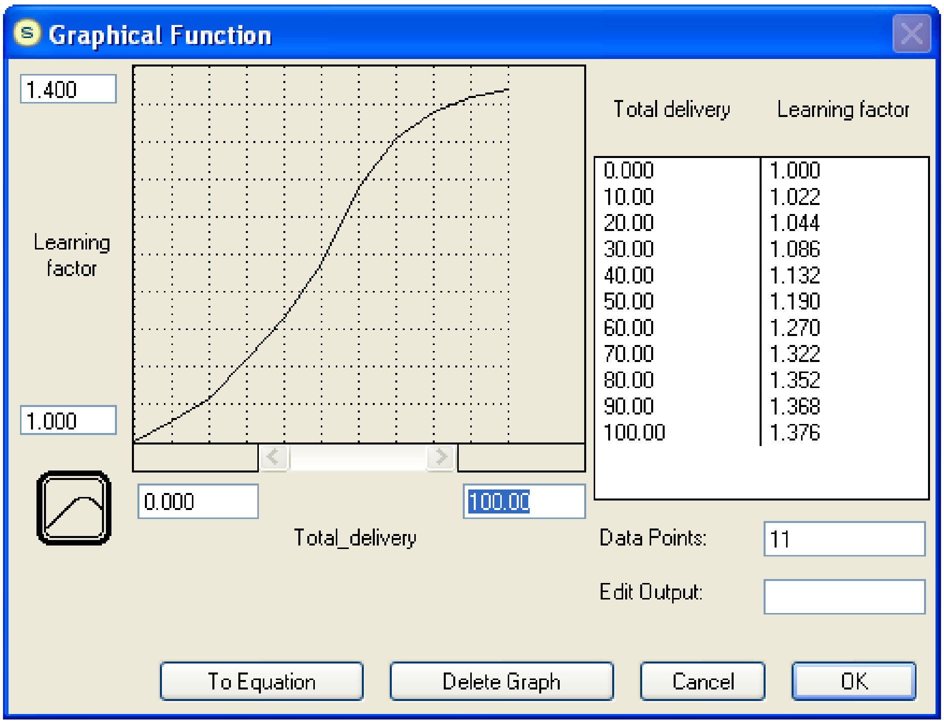


Figure 6.15 *The impact of maintenance delivery on competence level through self-learning*

Training. The managers of PUSDATIN are well aware of ever-improving IT and its impact on staff competence levels as well as the dynamic status of the eGSMA. The negative impact of this was responded to by PUSDATIN by conducting regular training for the staff in IT at least twice a year. The importance of training was also realised by the top manager of the Ministry and hence an adequate amount of budget was allocated specifically for training programs.

Responding to the negative impact of ever-changing IT constitutes 60% of the reason behind the decision to organise regular training (4). This interviewee explained that the other reasons, such as staff maintenance performance and incoming new staff, are much less significant. Staff maintenance quality performance and incoming staff could account for maximums of 30% and 10% respectively. The interviewee, who is the training coordinator within the unit, agreed with the prepared graph shown to him and depicted in Figure 6.16. The graph describes the relationship between competence level and training need. In this relationship graph, the abscissa represents the number of maintenance tasks that can be completed per person per quarter while the ordinate indicates the degree of training need.

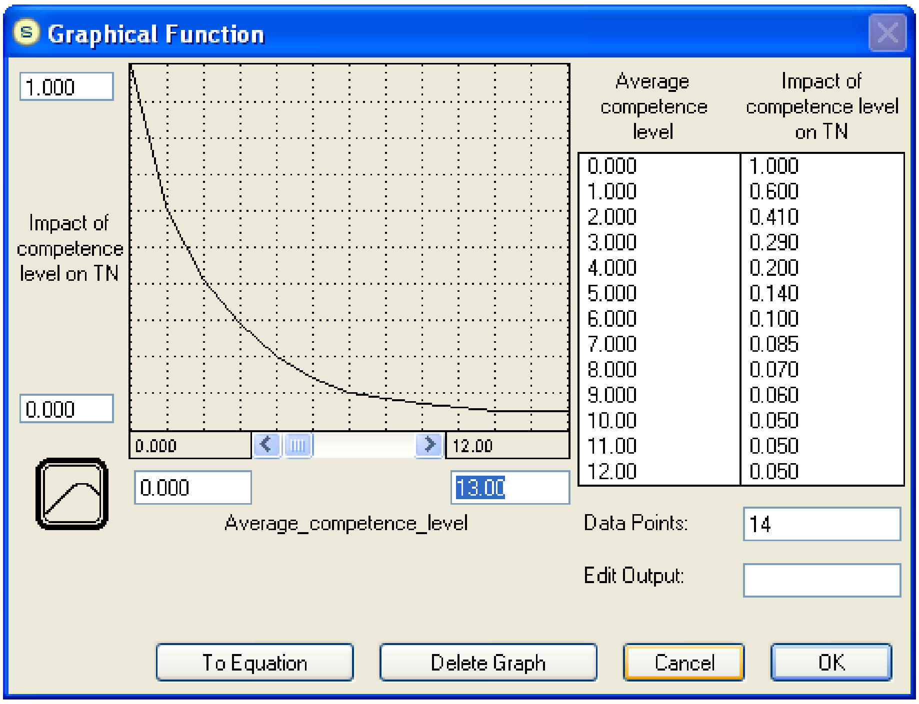


Figure 6.16 *The relationship between staff competence level and degree of training need*

This interviewee did not agree with two other graphs presented to him depicting, first, the influence of quality maintenance performance and, second, the influence of newly recruited staff on the decision to conduct training, but the interviewee did not suggest any alternative relationship forms. In this case, it is assumed that newly recruited staff are equal to less competent staff. The interviewee underlined that these two reasons are less considered in a practical sense.

Therefore, the initial SD simulation needs to be adjusted to reflect these two less considered factors. For this, the graphs of the initial model of the two factors are kept in place, but their weight in calculating the training need average is further reduced, such that the weights of the maintenance quality performance and new staff factors are set to *20%* and *5%* respectively. Consequently, this increases the weight for the competence level factor to *75%*. These reasons are used to restructure the *Training_need_varn.* Now, mathematically:

*Training_need_varn = 0.75*Impact_of_competence_level_on_TN + 0.05*Impact_of_LCS_on_TN + 0.20*Impact_of_recurrent_fault__on_TN*

The range of this variation is from 0 to 1.

The training program is organised by a senior staff member who was an interviewee. The training is delivered in collaboration with private companies. Topics for the training were offered to staff and if they were interested in a training topic they could attend the training (1)(4). On other occasions, it was the staff who requested a particular training (11) and, if PUSDATIN considered the training was valuable and related to staff jobs, then it was conducted. Sometimes, the training was carried out because of organisational needs (1). In this case, PUSDATIN assigned some staff to attend. On average, the staff attended training twice a year.

The interviewees explained that in addition to the acquired skills, training participants also received certificates, which could give them credit points. The accumulation of these credit points, along with other aspects, could be used to improve their position level, which in turn increased their income.

To date PUSDATIN has never formally evaluated the effectiveness of the training in improving staff competence level, even though the senior manager observed that an improvement effect on staff competence level did exist. All interviewees also acknowledged the impact of training on their competence level. The level of impact varied: 10% to 20% (10), 15% (11), 15% (8), 30% (9), 50% (7) and 75% (6). The interviewees stated that the degree of impact on competence level depends on the appropriateness of the training topic relative to the problems being faced by staff and the quality of the tutor. The improvement level rose to 50% or even 75% if the training contents matched with the problems being engaged by the staff. Accordingly, based on that data the average value of that impact is assumed to be 25% for each training program.

The average impact value is used as a reference to refine the graph that describes the relationship between the training gap, the *Actual_training – Training_need,* and the improvement in staff competence level. Thus, when the training gap is zero – that is, when the actual training equals the training need – then the improvement in staff competence level due to training is 0.25. The impact of training on the improvement in staff competence level is depicted in Figure 6.17, where the maximum value of the improvement is set to 0.75. This figure is used to set the *Training_factor*. The *Average_improvement_per_training* is set to 0.25 and the *Training_improvement_adjustment_time* is 1.

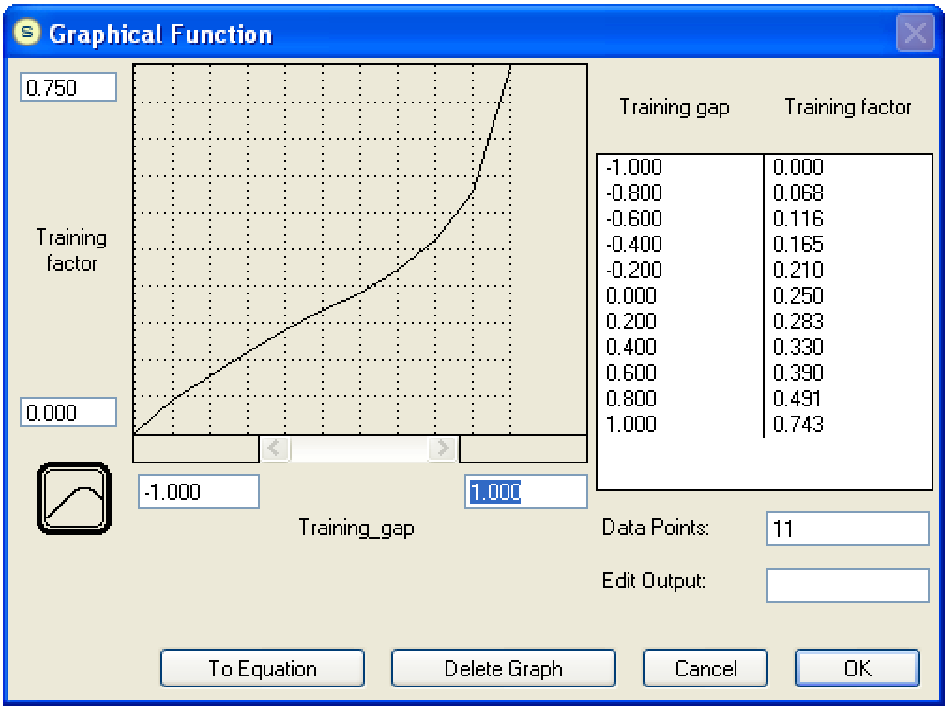


Figure 6.17 *The impact of training on improving the competence level*

The case study data, especially in relation to the staff competence factors, has been presented and used to validate and refine the initial SD simulation. Therefore, this validated structure and parameters of the staff competence factors and their relationships correspond to the significant structure and parameters of the staff competence factors and their relationships with the eGSMA. The validated and refined Staff competence factors sector is presented in Figure 6.18.

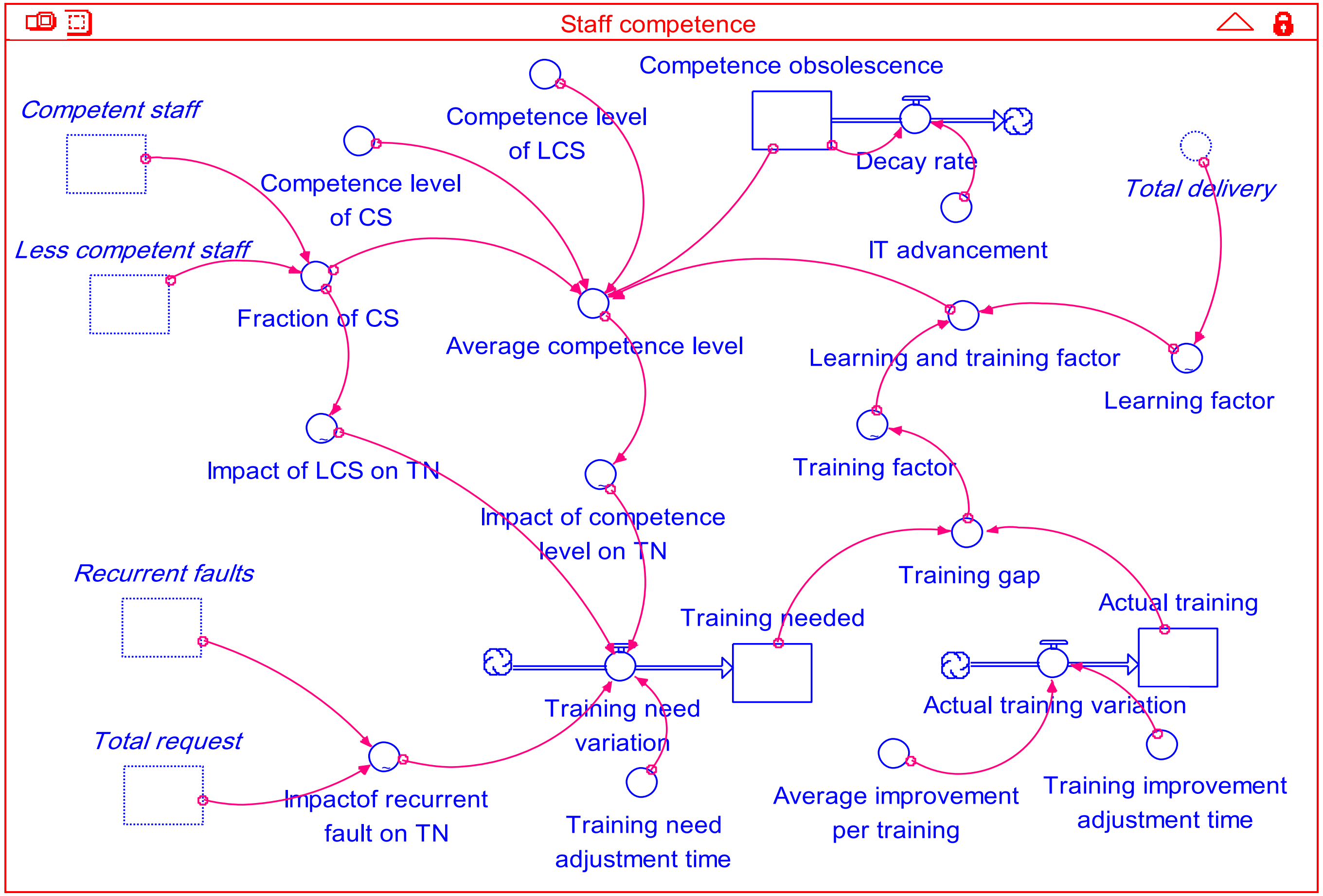


Figure 6.18 *The Staff competence sector of the eGSF SD simulation version 2.0 of the eGSMA*

6.4.6 Structure and parameter verification – Staff allocation sector

This subsection presents verification process of structure and parameters of the model in the Staff development and Staff allocation sectors, Figure 5.29 and Figure 5.31, respectively, against the case study data. This verification process is carried out in accordance with validation process presented in section 4.5. On completion of this verification process, a verified Staff allocation sector which is constructed from the verified structure and parameters of previously Staff development and Staff allocation sectors will be presented at the end of this subsection.

The case study data collected from interviewing the senior manager indicates that some parts of the initial model structure cannot be confirmed. These include IS staff category and supervision. For this reason, the initial model will be refined and simplified.

Personnel development. As mentioned previously, with regard to SM performance the senior manager categorised the staff into competent staff and less competent staff. In this case he disregarded the academic degrees held by the staff. Referring to the number of current staff, 16 persons, who were responsible for specific software operation and maintenance, the senior manager considered 11 staff to be competent and the rest less competent, Appendix B.2. Further, the senior manager said that about four years ago the number of less competent staff was eight. Because there are only two classification of IS staff, the general structure of the initial model associated with staff classification, presented in section 5.3.8, is not fully confirmed. Consequently, in accordance with this case data the model is simplified into one layer consisting of only *Less_competent_staff* and *Competent_staff* stocks, in contrast to the initial model which comprised two layers with five stocks.

Considering the fact that two staff members were recruited in 2008 and 2009, the initial value of the total number of staff is 14; the stocks initial value of the *Less_competent_staff* is set to eight and that of the *Competent_staff* is set to six. The change in the number of current *Competent_staff* compared with the number four years ago shows that there has been an improvement in the level of staff competence. The improvement is modelled by the *Improvement_rate* which is a division function of the *Less_competent_staff* and the *Improvement_* *adjustment_time*. The larger the *Improvement_adjustment_time* value, the longer it takes to improve. The formulation for this factor is:

Improvement_adjustment_time = Base_improvement_adjustment_time / Learning_and_training_factor.

In this case the *Base_improvement_adjustment_time* is set to 26 because at the end of simulation time there should be 11 competent and five less competent staff. The *Improvement_adjustment_time* is formulated as division because the *Learning_and_training_factor* value is larger than 1. This means as the *Learning_and_training_factor* increases, the *Improvement_adjustment_time* decreases, which in turn causes the *Improvement_rate* – that is, the number of staff who improve – to increase.

Staff recruitment and retirement. The senior manager explained that PUSDATIN cannot recruit new staff independently and directly but must request the Ministry through the Personnel Bureau of the Ministry. The bureau will allocate a quota for PUSDATIN by considering the whole requirements of the Ministry. The recruitment of new staff is a once-a-year process but there is no specific limitation on the number of new staff to be recruited.

For this reason, in order to avoid zero recruitment the *Recruitment_limitation* is set to one. The number of new staff recruited is formulated as:

New_staff_recruited = MIN(recruitment limitation, total staff gap)

This is then used to determine the *Recruitment_rate*.

Recruitment_rate = New_staff_recruited/Recruitment_adjustment_time

As recruitment occurs once a year, the *Recruitment_adjustment_time* is set to four quarters.

On the other hand, staff exit from PUSDATIN so far is almost non-existent other than for two staff who retired over the period of the last four or five years. In general, they stay as PUSDATIN staff until retirement time comes. For example, there are four current staff who started their careers at PUSDATIN as programmers from 1981, or about 30 years ago. This study assumes that this also applies to the other staff. This assumption also means less competent staff never exit. One interviewee mentioned that the advantage of being a civil servant is in terms of job and income security, especially during unstable national economic conditions (8). This staff employment time is used to input the exit rate, which is represented by an outflow from the *Competent_staff* stock named *Exit_rate.*

Exit_rate= *Competent_staff/exit_adjustment_time*

The *Exit_adjustment_time* is set to 120 quarters, or 30 years.

Considering the number of staff with respect to the PUSDATIN SM workload, the senior manager regarded it as sufficient based on his experience and observation, although there has not been any formal study of staff sufficiency. Consequently, the total staff needed can plausibly be assumed to be equal to the current total staff. As the number of staff responsible for SM is currently 16, so the *Total_staff_needed* is set to 16. As in the initial model, the difference between the number of current staff and the number of staff needed is modelled by *Total_staff_gap*.

Total_staff_gap = *Total_staff_needed* – *Total_staff*

Staff availability. The senior manager confirmed that some senior and competent staff supervise less competent ones but this supervision does not take up a significant portion of their working time. Therefore the senior manager stated that staff availability is not affected by their supervisory activity. Accordingly, the *Supervisors* factor is eliminated from the initial model. This elimination causes:

Total_staff_available = *Total_staff* * A*verage_availability_per_staff*

In turn, the *Average_availability_per_staff* is determined by the level of staff absence, other duties and tasks from VIP.

One interviewee (the operational manager) said that the staff absence level is low, although there is no specific data about this. Some of the staff currently have external jobs but these are performed after working hours or on the weekend (5)(6)(7). Thus, these external jobs do not affect absence level. Considering this fact, it is assumed that the average absence level is 5% of the total working days in a year, and the relationship between reward performance difference and absenteeism of the initial model is kept in place.

The average absence level will be used to set the minimum value of the graphical relationship between the *Reward_performance_difference* factor and absence level, which is represented by the *Impact_of_RP_on_absence_level,* shown in Figure 6.19. This graph replaces the relationships graph depicted in Figure 5.32.

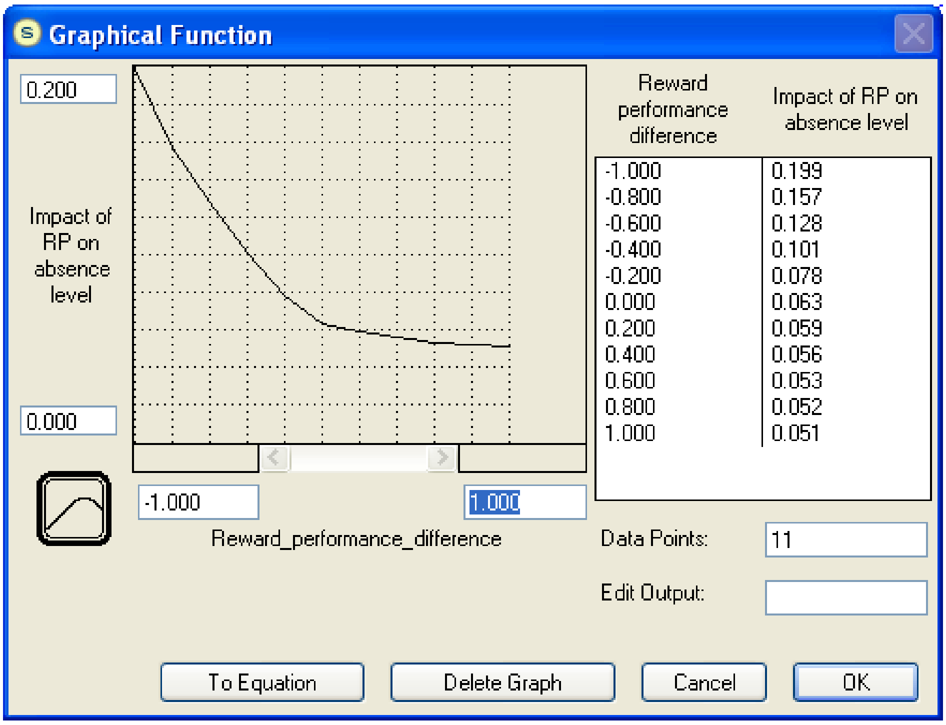


Figure 6.19 *The case study relationship between reward performance difference and absence level*

Other duties. In addition to maintaining the software, all the interviewees mentioned that they have to perform other organisational tasks. The assigned tasks can be either related or not related to the software for which the staff are responsible and can include other software development, user support and training, meeting and ceremonial activities and budget planning. According to the senior manager, the workload of these organisational tasks varies from 10% to 50% of the staff total workload. The staff who are assigned these tasks are those who have high performance levels, because these additional tasks could lead to more income (1)(2). Based on this data, the *Other_duty_level* is set to 0.3, that is, the mid value of the range of the level of other organisational tasks.

Tasks from VIPs. Another factor that affects staff availability for SM is tasks from VIPs in the Ministry which can pre-empt staff maintenance work. However, it is assumed that these pre-emptive tasks do not apply to all staff at the same time, or that only a portion of staff are given this type of task. This is modelled as *Tasks_from_VIP*. The interviewee mentioned that on average the VIP tasks are about 7.5% (2). This value is used to set up the *Tasks_from_VIP* factor.

Considering these factors, then:

Average_availability_per_staff = 1 – (Impact_of_RP_on_absence_level + Other_duty_level +Tasks_from_VIP)

Staff allocation. In terms of placing more importance on CM than EM, the senior manager said this is because a fault or error could stop the software from working, while enhancement could wait. If the importance of CM is translated into the number of staff allocated to CM tasks, then it means that if both types of maintenance task occur concurrently, some of the EM staff are reallocated to CM tasks. However, the proportion of reallocated staff is less than 50% of the current number of EM staff.

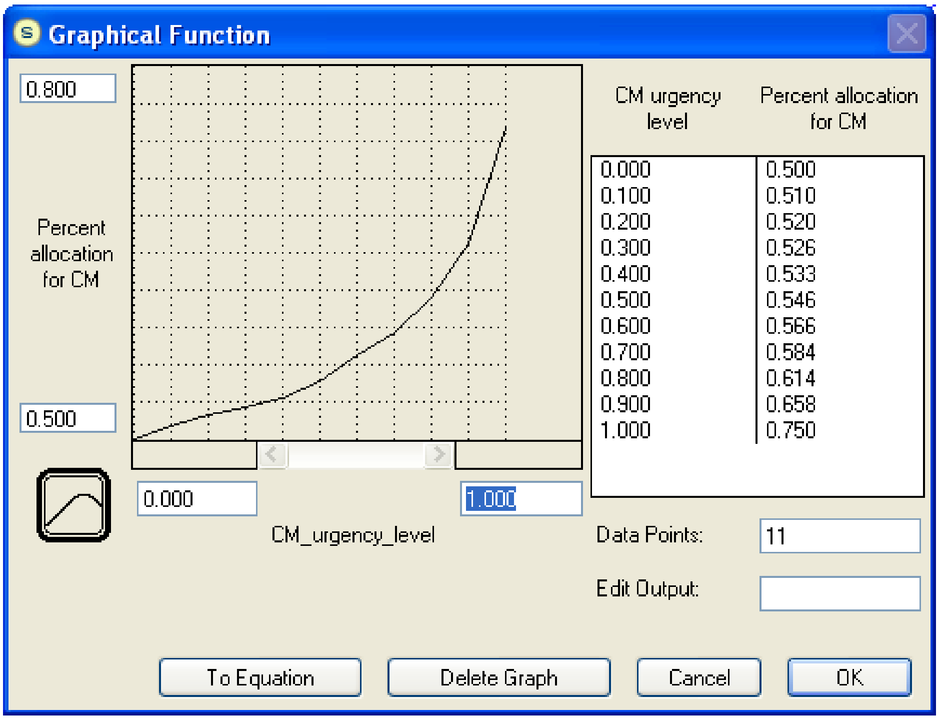


Figure 6.20 *The relationship between the urgency of CM and the percentage of staff allocated for CM*

This data is used to set the maximum value of the *Percent_allocation_for_CM* (see Figure 6.20) which is 75%.

The multiplication of the *Percent_allocation_for_CM* with the *Total_staff_available* determines the actual number of staff allocated for each type of maintenance task. Therefore, the:

Daily_manpower_of_staff_for_EM = (1 – *Percent_allocation_for_CM*) * *Total_staff_available*

and

Daily_manpower_of_staff_for_CM = *Percent_allocation_for_correction* * *Total_staff_available.*

The presentation and analysis of the case study data with regard to the staff development of the eGSMA cause a necessary modification to the structure of the sectors. To simplify the model representation, model structure associated with IS staff development and allocation is now represented in one sector named the Staff allocation sector. This model sector consists of only one layer and does not include factors related to supervision. This model sector, which corresponds to the significant structure and relationships of the success factors of the eGSMA, is presented in Figure 6.20.

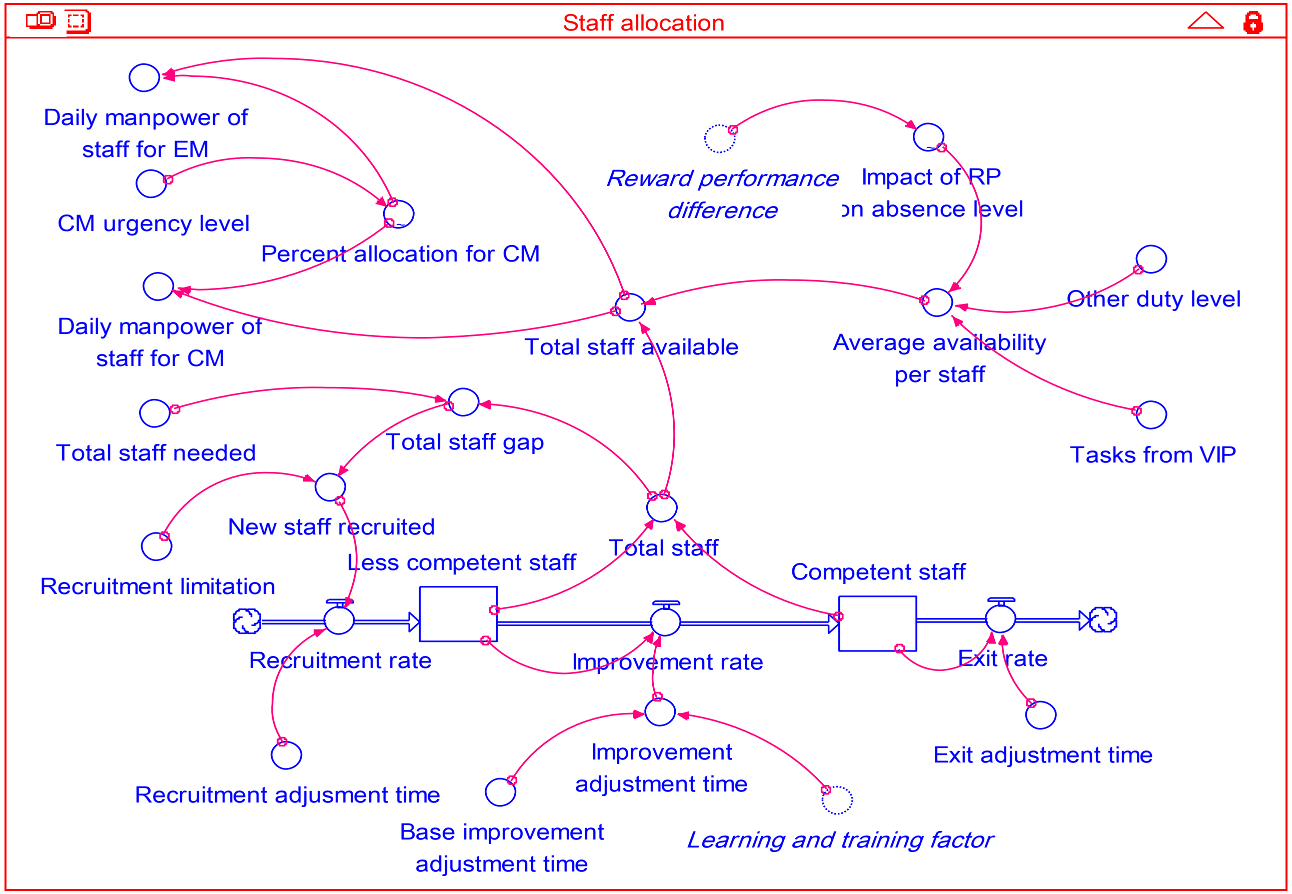


Figure 6.21 *The Staff allocation sector of the eGSF SD simulation version 2.0 of the eGSMA*

The structure and parameter verification process which has been performed in this section (section 6.4) results in the model version 2.0. Now, the model consists of four sectors:

- Maintenance process sector – Figure 6.9,
- Motivational factors sector – Figure 6.14,
- Staff competence sector – Figure 6.18, and
- Staff allocation sector – Figure 6.21.

The model of this version was built up gradually using the knowledge derived from previous studies, the researcher's assumptions and the eGSMA case study data. As the structure and parameter verification has been undertaken, the structure and parameters of the model version 2.0 correspond to the structure and parameters of the success factors system of the eGSMA; and the significant structure and parameters of the success factors of the eGSMA relevant to the study have been represented properly in the model.

As previously mentioned in Chapter 4, section 4.5, the model of version 2.0 will undergo a desk validation process in accordance with the steps presented in Figure 4.5. The following subsections present this process. Because the tests that will be conducted as part of the validation process to the model version 2.0 may involve some changes in the model parameters, the model version 2.0 is called the base case.

6.4.7 Dimensional consistency test

Now, the model version 2.0, which consists of four sectors, will undergo a dimensional consistency test. The dimensional consistency test aims at ensuring the measurement units at the left-hand side and right-hand side of an equation in the model version 2.0 are equal. It is also useful in revealing whether a parameter included in the equation is meaningful as a component of the model structure. A failure to pass this test indicates a faulty model structure (Forrester & Senge 1980: 216). For this study, the test was performed through the STELLA® facility along with the model development and refinement process. The output from STELLA for each converter, flow and stock was reviewed. The dimensions of all equations of the SD simulation have been tested and have passed. The meaning of the scaled parameters in each equation of the SD simulation has also been re-evaluated. It has been ensured during model development and refinement as well as model testing that each parameter included in an equation has been well defined and explained, and that all scaled parameters have practical meaning relevant to the problem.

An example is provided as follows. An equation that determines the rate of CM delivery named *CM_delivery_rate* is evaluated. The dimension of this rate is *tasks/quarter* which means the number of maintenance tasks that can be completed and delivered in one quarter. The rate is formulated as:

*CM_delivery_rate=Daily_manpower_of_staff_for_CM * CM_productivity_variation*

and the dimension of this equation is:

*tasks/quarter = persons * tasks/person quarter*
= task/quarter

Therefore, the dimension on the left-hand side equals the dimension on the right-hand side of the equation. This mechanism can be performed through STELLA® software. For example, a dimension evaluation of the equation that forms the *Maintenance_request_variation,* the variation of the incoming maintenance request, results in Figure 6.22; and a dimension test on the overall equations in model version 2.0 produces the output in Figure 6.23.

In sum, the dimension consistency test concludes that the dimensions of the left-hand side are consistent with those on the right-hand side of all equations in the model version 2.0.

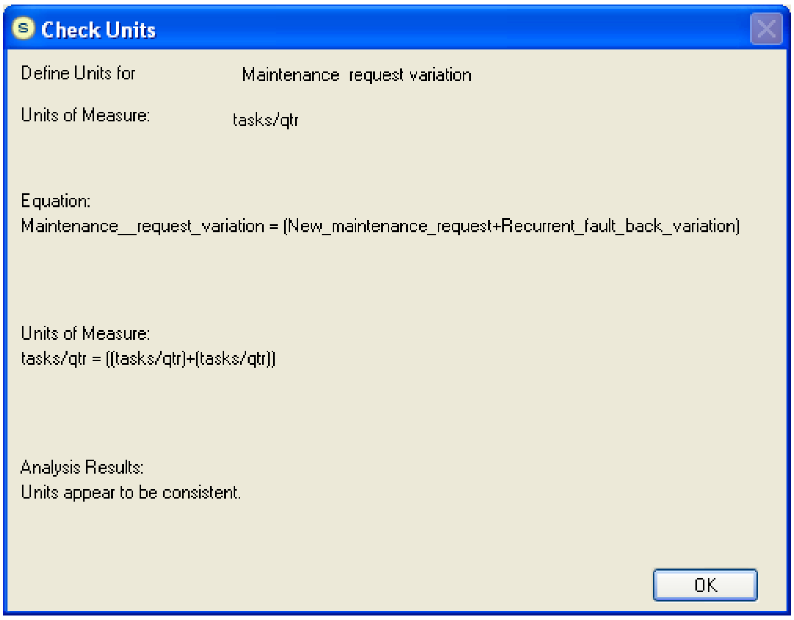


Figure 6.22 *The dimension evaluation of the* Maintenance_request_variation *using STELLA®*

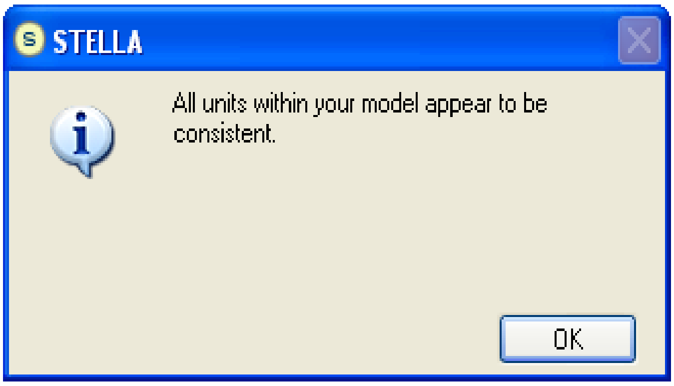


Figure 6.23 *The result of the dimension evaluation of the overall equation using STELLA®*

6.4.8 Boundary adequacy test

Once the dimensional consistency test completes, the next test over the model version 2.0 is the boundary adequacy test. The implementation of this test in this study is indicated by the sectors, exogenous and endogenous factors of the SD simulation. The SD simulation sectors disaggregate the overall system under interest according to the SD problem formulation and the purpose of the model. Consistent with the purpose of the case study SD model, the primary important exogenous and endogenous factors of the model based on the case study are presented in the following table, Table 6.6. Specifically, these factors do not include those that become the elements of the driving and governing factor, service delivery management, external factors, hardware, IS development, etc.

Table 6.6 *The endogenous and exogenous factors reflecting the boundary of the SD model*

Endogenous	Exogenous
Maintenance process sector	
Maintenance requests	Project quality and enhancement pressure (impact of time to new maintenance request)
Maintenance deliveries	Semaphore rate for recurrent faults
System size and complexity	New releases
Recurrent faults (system quality)	
Motivational factors sector	
Reward provided to staff	Staff communication level with environment
Reward expected by staff	
Staff productivity	
Staff effort	
Maintenance performance	
Staff competence sector	
Average competence level	Staff initial competence level
Learning and training	IT advancement
Level of training needed	Actual training
Staff allocation sector	
The number of competent and less competent staff	Other duties and VIP tasks
Staff availability	The level of urgency of a CM task
Staff allocation	Total staff needed
	Limitation on the number of new staff recruited
	Length of tenure
	Recruitment period

6.4.9 Extreme conditions test

This is conducted to the model version 2.0 after the boundary test has been completed. This test is performed by introducing appropriate extreme values (large negative, zero or large positive values) to the model. In implementing this test, in order to obtain meaningful results the introduced extreme values should also have an association with practical meanings of the real-world situations.

Otherwise, the test will not give any value to the enhancement of the model soundness and usefulness.

Negative extreme values. Negative extreme values are not introduced because a negative value within the context of the study does not have real practical meaning. For example, a negative value of total staff or incoming requests does not have practical meaning. It is not possible for PUSDATIN to "borrow" (which could be associated with a negative value) staff from other organisations.

Zero value. The model is subjected to the introduction of a zero value which is implemented to a number of factors. The factors and the meanings of the zero value setting are presented in Table 6.7. The expected reasonable output for each factor setting is that there is no maintenance delivery all the time. Therefore, the ratio between maintenance deliveries and requests should also be zero all the time; or, especially for the CM and EM, if the *Corrective_percentage* is set to zero or is set to 1, then the CM or EM delivery should be zero.

Table 6.7 *The extreme condition test with zero value setting*

Factor	Meaning
Impact of Time on NMR	There is no incoming maintenance request.
Effort variation	The staff does not exert any effort.
Other duty level, *Impact of RP on absence level,* *Task from VIP*	This is set to 1. All staff are not available for doing maintenance jobs because they are doing other tasks or absent all the time
Less competent staff, *Competent staff, and* *Total staff needed.*	The staff are not available and there are no new staff needed.
Competence level of CS, *Competence level of LCS*	All staff do not have any competence.
Corrective percentage or *1- Corrective percentage*	If the corrective MR is zero, then all incoming MRs are of the enhancive type, and vice versa.

This zero value test is exactly the same with the physical conservation law test, which means if there is no input, then there is no output. In addition to the ability of the model to pass this test, the introduction of zero value has been able to identify the division-by-zero-problem within some mathematical equations, such as that involving division by the *Total_request* factor. These problems have been rectified by introducing IF THEN ELSE statement. This test has also been able to mark a negative value problem, which means revealing an incorrect formulation such as the formulation of the *Average_availability_per_staff*. The problem has been overcome by re-formulating the equation as:

Average_availability_per_staff = MAX
(0, 1 – (Impact_of_RP_on_absence_level + Other_duty_level +
Tasks_from_VIP))

As the graphical outputs have shown the expected results, that there is no maintenance delivery, then the model version 2.0 is considered to have passed this test. The graphical outputs are presented in Appendix D.1.1.

Positive extreme values. The third type of extreme value test is performed by setting parameters, stocks or rates to a positive large value relative to the current value used to input the model. To perform this testing, the following are setting examples applied to the model, see Table 6.8. Graphical output when MRs is 100000 times higher is provided in Appendix D.1.2.

Table 6.8 *The extreme condition test with large positive value settings*

No	Parameter, stock or rate	Setting	Meaning of the setting and expected output
1	*Maintenance request variation*	100000, or STEP (100000,10)	The number of MRs is unfeasibly large from the very beginning or from the 10th quarter. Zero level of delivery and request ratio from the beginning or after 10th quarter
2	*Reward adjustment time*	4000	Practically there is no additional reward. Effort will decrease to the lowest level.
3	*Communication level with experienced staff*	STEP(1,10)	An MR that can only be solved with external experts assistance which are not available since 10th quarter. Zero level of delivery from 10th quarter to the end.

As seen in Appendix D.1.3, the graphical output seems as expected. The graph resulting from the first setting describes that the dynamic of the delivery and request ratio shows an extreme drop after the 10th quarter. This is so because the requests become very large.

The figure as a result of the second setting shows that as there is no reward, the exerted effort due to reward is never increased from its base effort level: 0.875 out of official working hours a day, see Appendix D.1.4. It must be noted that in this case, from the beginning of the simulation period the organisational rewards were lower than the expected reward level. However, the exerted effort does not continuously decrease because the staff never reduces the base effort level. The absence of the organisational rewards affects staff absence up to its maximum level of 0.2; and it affects the availability of the total staff to decrease to a certain level. Once the staff absence reaches its maximum level, the availability of the total staff starts to increase, caused by new staff recruitment. As the output is plausible, then SD simulation version 2.0 passes this kind of test.

The graph, as a result of the third setting, indicates that the output is plausible, see Appendix D.1.5. As the staff are unable to get a solution from the expert from the 10th quarter, then the productivity level drops and causes zero delivery. Therefore, the delivery and request ratio decreases from the 10th quarter. This decreasing pattern is because the cumulative delivery does not increase; on the other hand the cumulative requests keep rising.

6.4.10 Behaviour sensitivity test

This test is also applied to the model version 2.0 after the extreme condition test completes. Technically, the behaviour sensitivity test in this study is conducted by following the one shown in Maani and Cavana (2007, Case 5). First, several parameters and graphical relationships of the model version 2.0, for which uncertain values are introduced, are selected and listed. The model version 2.0 that has passed all previous tests and which serves as the base case is run. The values at the end of this simulation run are listed. Then each of the parameters and relationships of the base case is decreased and increased by 10%. For each decrease or increase, the simulation is run. For example, the base case of the *Other_duty_level* parameter is 0.3, and the 10% decrease and 10% increase of this value are 0.27 and 0.33 respectively. For a non-multiplier graphical relationship, these decreases and increases are applied to all values of the graph. While for a multiplier graphical relationship which contains 1 that means no effect, the settings are applied by calculating the difference between the graph values and 1, then multiplying these differences by 1.1. The resulting values are then added to 1 or subtracted from 1. The selected parameters and graphical relationships, for which their values are considered uncertain, are presented in Appendix D.2.

Second, for the purpose of this sensitivity test, four important factors are selected and their final values at the end of the simulation run are observed: *Delivery_and_request_ratio* as a measure of SM performance; *Average_competence_level* as a measure of overall staff competence level; *Productivity_variation* as a measure of motivational factors; and *Total_staff_available* as a measure of staff allocation. To perform the simulation run for each 10% decrease and 10% increase, the facilities available within STELLA® software are utilised. For example, Appendix D.3 is the interface created within STELLA® to conduct the sensitivity test especially for the Maintenance process sector. It contains slider and graphical inputs, and numerical outputs of the selected factors.

Third, after running the simulation with 10% decrease and increase the outputs are tabulated. Again, using the criteria in Maani and Cavana's case, the parameters and graphical relationships are classified as sensitive if there is 5% – 14% change, very sensitive if there is 15% – 34% change, and highly sensitive if over 35% change is observed within the selected important factors. The observed changes are presented in Appendices D.4 and D.5.

Table 6.9 *Summary of the sensitive parameters*

		Delivery and request ratio	Average competence level	Productivity variation	Total staff available	Classification*)
	10% decrease / 10% increase					
	Maintenance process					
1	Task multiplier					both
2	System increase factor					both
3	EM analysis factor					both
4	EM development factor					both
5	EM delivery factor					both
6	Recurrent fault percentage					internal
7	Recurrent fault semaphore rate					external
	Motivational factors					
8	Organisational reward factor					internal
9	Expected reward factor					external
10	Reward adjustment time					internal
11	Reward expectation adjustment Time					external
12	Impact of RP difference on effort			–6% / 6%		both
13	Impact of Performance on effort					
14	Base effort					
15	Communication level with experienced staffs					both
16	Communication level with users					both
	Staff allocation					
17	Impact of RP on absence level					
18	Other duty level				5% / –5%	internal
19	Tasks from VIP					internal
20	Base improvement adjustment time					external
21	Recruitment limitation					internal
22	Total staff needed				–10% / 9%	internal
23	CM urgency level					internal
24	Percent allocation for CM					internal
	Staff competence					
25	Competence level of CS		–10% / 10%	–10% / 10%		internal
26	Competence level of LCS					internal
27	IT advancement		14% / –12%	14% / –12%		external
28	Learning factor		–12% / 12%	–12% / 13%		external
29	Training factor					internal
30	Average improvement per training					both
31	Training improvement adjustment Time					internal
32	Impact of competence level on TN		-5% /	-5% /		internal

*) Classification column indicates whether the associate factors are under e-government system management control or not: internal – under control; external – beyond control; and both – management can control up to some levels.

The summarised results of the behaviour sensitivity are given in Table 6.9. It can be observed that none of the parameters is very or highly sensitive towards the selected parameters. This table shows that the *Delivery_and_request_ratio* is not sensitive to all parameters and all selected parameters of the sectors are not sensitive to all parameters of the Maintenance process sector. The *Total_staff_needed, Competence_level_of_CS, IT_advancement* and *Learning_factor* relatively have higher sensitivity levels compared with other sensitive parameters and the decision-makers or managers of an e-government system should pay more attention to them. On the other hand, insensitive parameters indicate that small changes to the base value do not cause significant changes to the selected important parameters. Possibly, with structural changes these insensitive parameters could cause considerable variation to the performance parameters.

6.4.11 Behaviour reproduction test

Following the behaviour sensitivity test is the behaviour reproduction test which is also performed to the model version 2.0. The purpose of this test is to evaluate the closeness of the time-series simulation run output reflecting the behaviour of the model with the reference mode of the eGSMA. However, as mentioned in the previous section, data about the reference mode or the steady-state software availability level over time of the eGSMA is not available. Any documents indicating MRs and the maintenance delivery cannot be obtained because they are unavailable. Therefore, the use of point-by-point fit to evaluate the closeness is infeasible. On the other hand, PUSDATIN has shown that it can successfully maintain its software for a relatively long time. The interviewees mentioned that the percentage of unfulfilled MRs is less than or equal to 10%. Based on this data, it is assumed that the steady-state software availability level over time of the eGSMA is about 90% or higher.

Accordingly, to pass this test the model should be able to produce a delivery and request ratio which stays at a high value, about 90%, for quite a long time, more than four years. The judgement of the ability of the model to reproduce the dynamic behaviour of the software availability of the eGSMA is mainly qualitative but is guided by the dynamic pattern produced by the model. A relatively stable high value of the ratio will affirm that all significant factors, relationships and structures as well as parameter values of the sustainable eGSMA have been well represented by the model version 2.0.

For this test, the model version 2.0 is run to produce the dynamic behaviour of the *Delivery_and_request_ratio* as the measure of the steady-state software availability level and SM performance. The dynamic behaviour of the ratio is presented in Figure 6.24.

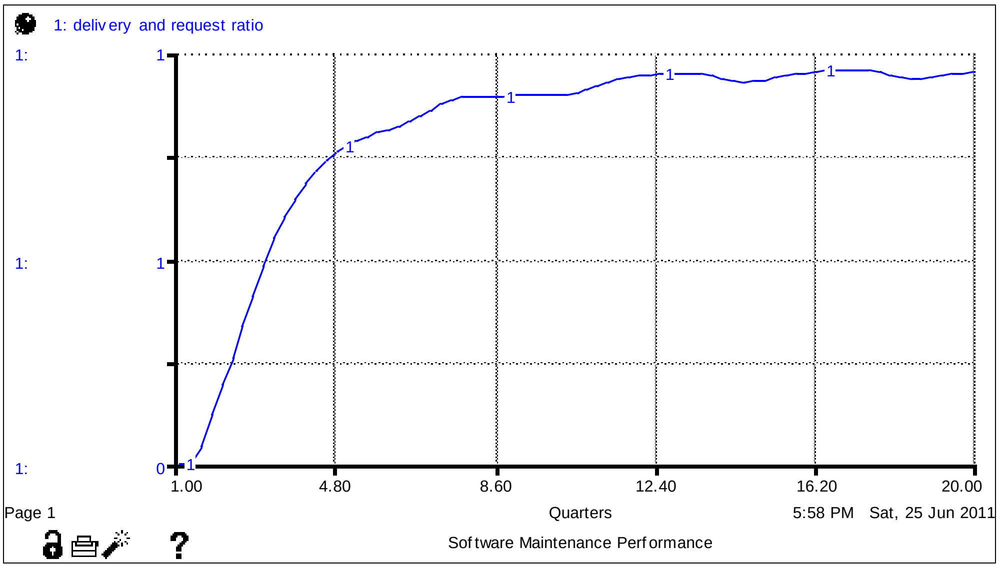


Figure 6.24 *The dynamic behaviour of the SM performance of the base case*

As can be seen in Figure 6.24, the level of the ratio over time is steadily increasing from 0 from quarter 0 to quarter 5, then is close to 1 after quarter 5 for the rest. To understand the early pattern of the ratio in Figure 6.24, which does not start directly with a high value, two aspects must be considered. First, the ratio is the number of deliveries divided by the number of requests at the same time point, say *t*. The deliveries at a particular point of time actually refer to the requests at a certain interval time prior to that point of time, or *t-a*, (*a* is a positive constant value) because maintenance takes time. Second, the formulation of the ratio is based on the cumulative value of maintenance delivery and request; the delivery always gradually increases from zero, while the request can start with any positive value. Accordingly, in general this delivery and request ratio satisfies the behaviour over time of the steady-state software availability level of the eGSMA, except for the early period of the run.

6.4.12 Behaviour anomaly test

In accordance with the validation process described in subsection 4.5.2, Figure 4.5, this is the last test performed on the model version 2.0. In the search for any behaviour anomaly of the model, the test found likely anomalous behaviour when the *Recurrent_faults_percentage* is set to a relatively large and constant value. In this case, it is set to 0.2. The *Recurrent_faults_percentage* represents the quality of maintenance; a higher value means a low-quality maintenance delivery. This reduces the usability of the completed and delivered maintenance tasks. By setting it to a relatively large and constant value, it is assumed that the quality of maintenance delivery is relatively low and does not change over time. The impact of this setting on software availability or maintenance performance is an anomalous pattern, see Figure 6.25. The strange behaviour is shown by the fact that the ratio increases but after a particular time period it is monotonically decreasing, although in this test other factors setting are not changed, including the decreasing new MRs or *New_maintenance_request*.

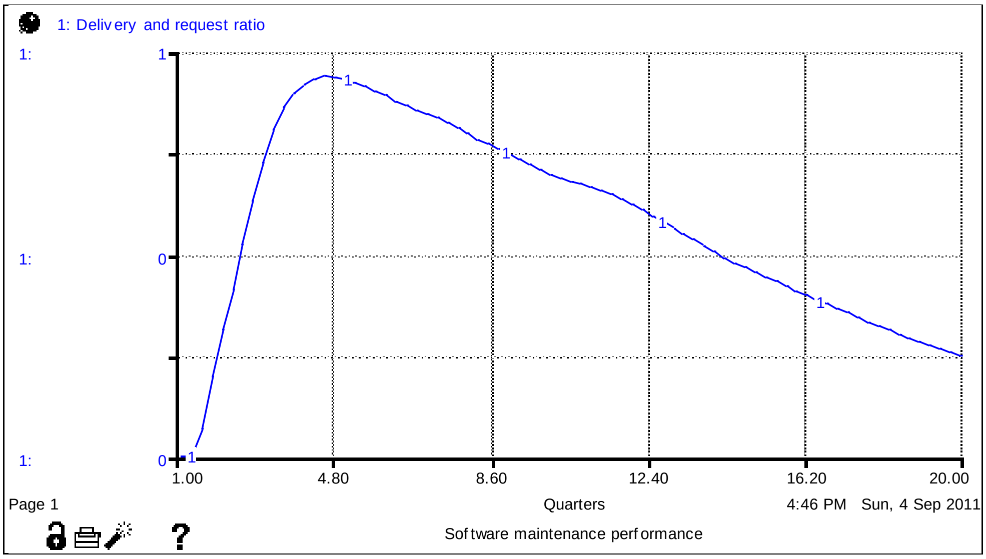


Figure 6.25 *Anomalous behaviour of software availability under a constant and large value of Recurrent_faults_percentage assumption*

Although the *Delivery_and_request_ratio* shows an anomalous pattern, by tracing all feedback-loop relationships that link the *Recurrent_faults_percentage*, the decreasing pattern of the ratio is understandable. First, this is because of the existence of negative reinforcing feedback-loop between the percentage, MRs, and the ratio. A high percentage of recurrent faults prevents the total MRs from decreasing, which eventually reduces the ratio level. Second, once the ratio is continuously decreasing, the rewards are affected in such a way that the difference of provided and expected rewards does not make the staff effort higher than the base effort level, as in Figure 6.26. Third, the staff absence level tends to increase after the 12th quarter, which in turn reduces the staff availability level – or the staff are still available but mostly dominated by new staff, see Figure 6.27. All of these cause the ratio to continuously decrease.

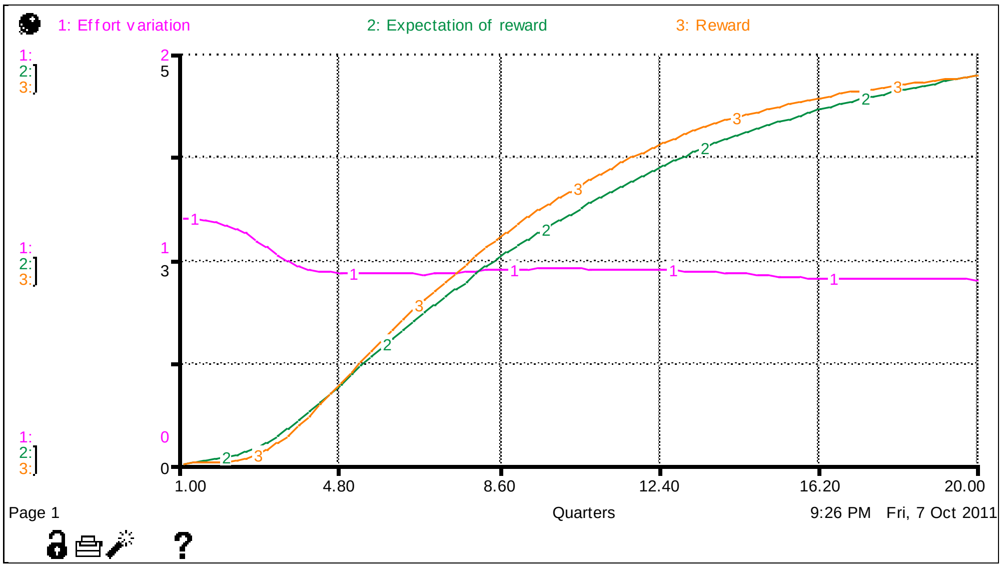


***Figure 6.26** The impact of a constant and large value of* Recurrent_faults_percentage *assumption on rewards and effort*

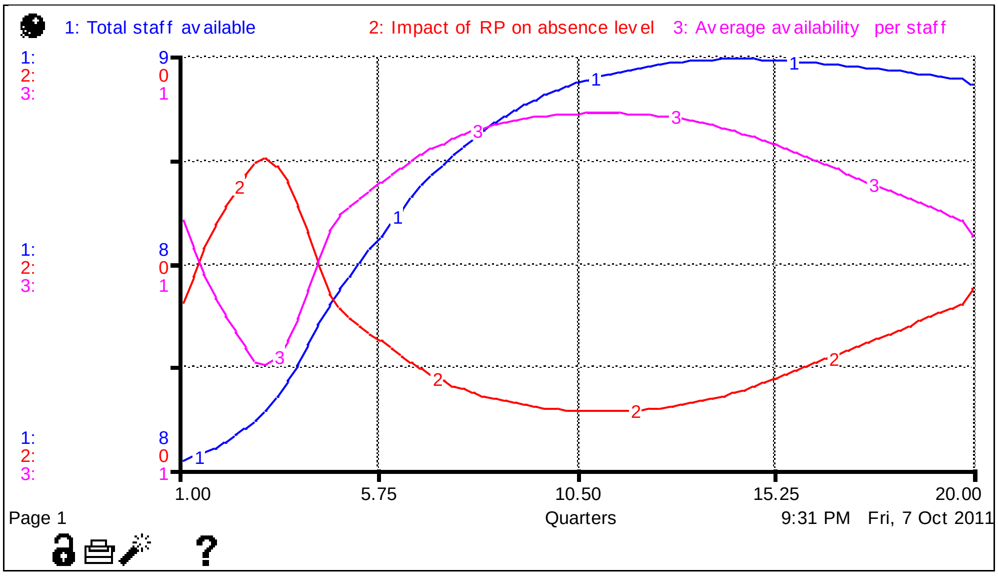


***Figure 6.27** The impact of a constant and large value of* Recurrent_faults_percentage *assumption on staff availability*

In sum, the anomalous behaviour can be tracked within the model, therefore model version 2.0 passes this test. The test also shows the importance of keeping a high quality level of the maintenance delivery factor.

6.4.13 Conclusion

The validation process has been performed to the eGSF SD model version 1.1. The CLD has been refined based on the data from a case study and the structure and parameters of the simulation model have also been validated against the case study data, resulting in eGSF SD model version 2.0. During the structure and parameters verification process, it has been indicated which structure and parameters of the initial model presented in Chapter 5, sections 5.3 are confirmed or not confirmed by the case study. The eGSF SD model version 2.0 is presented as a CLD in Figure 6.3 and a simulation model, which consists of four sectors, is presented in Figures 6.9, 6.14, 6.18 and 6.21. Therefore, the model can also be called the eGSMA SD model.

A further validation process as required by the SD method has been conducted on simulation model version 2.0 confirming its model structure and behaviour, and further refinement is not necessary. Therefore, eGSF SD model version 2.0 now becomes eGSF SD model 2.1 or validated eGSMA SD model.

It must be noted that although the initial model was developed from various research not specific to developing countries, the model is not intended to represent e-government success in developed countries also. First, the model was validated against a successful e-government system in a developing country. The criteria for the selection of the case were based on the working definition of e-government success, which specifically refers to the common level of e-government index of developing countries. The average index for developing countries is much lower than for developed ones. Second, most of the parameters of the validated model are derived from the case study which might not represent e-government systems in developed countries such as system size, competence level and turnover rate.

6.5 Summary

This chapter has presented the validation process of eGSF SD model version 1.1 and the process has resulted in eGSF SD model version 2.1 or validated eGSMA SD model. The documentation for the equations, graphical relationships, parameter values and initial values is provided in Appendix E. This process implements the research design and method presented in Chapter 4 in order to validate and refine the initial and conceptual model, eGSF SD model version 1.1 as presented in Chapter 5, which was developed based on the literature review and the researcher's mental model.

To represent an actual successful e-government system, a successful e-government system in Indonesia has been selected for the implementation of a single case study. In this study, the case is the eGSMA. Data collection has been carried out guided by eGSF SD model version 1.1 and the data has been used to confirm or disconfirm and refine eGSF SD model version 1.1. The CLD as well as the structure and parameters of the simulation model have been refined based on the case study findings.

There is some possible bias in the data as a result of the unavailability of formally documented data and reliance on the personal judgements of the interviewees. However, as the interviewees are those who have been intimately involved in e-government SM for a relatively long period, the bias can be reduced. In order to build confidence in the SD simulation model, a series of further tests has been undertaken: dimensional consistency, extreme conditions, boundary adequacy, behaviour sensitivity, behaviour reproduction and behaviour anomaly tests.

As in SD in general, the validated eGSMA SD model can now be used with high confidence to evaluate and analyse a policy or a particular situation. In Chapter 7, this validated model will be discussed by using it to analyse success factor relationships of the eGSMA, and be used to evaluate and analyse why and how a (set of) factor(s) and their relationships are important for – or give high leverage to – e-government success.

Chapter 7 Discussion

7.1 Introduction

This chapter discusses the usability of the validated simulation model and presents some interesting results from using the model. The validation process in the previous chapter has built a sufficient level of confidence in the soundness and usefulness of the model which therefore has reasonably represented a success factor system of an actual successful e-government system. Hence, the model can be used with a reasonable level of confidence to design and analyse policies for e-government system improvement (Sterman 2000**:** 103).

In this chapter, the model will be used to explain why the case study achieves success through the chain of success factor relationships and to identify possible leverage factors for its e-government success. In addition, the model will be utilised to analyse some factors identified in previous studies which have been shown to cause e-government failure. For this, a number of simulations will be run over the model to evaluate and analyse different scenarios applied to the model which accommodate those factors. The discussion and analysis of the simulation outputs could assist the e-government decision-makers or managers of other e-government system units in Indonesia or other developing countries to design policies preventing the conditions that cause e-government failure from its occurrence in the future (Maani & Cavana 2007 Ch. 5).

In order to present the discussion and analysis, this chapter is organised as follows. First, the description about the scenarios that will be simulated on the validated model is presented. Second, the scenarios are applied to the validated model, a simulation is run on each scenario and the results are analysed. Before the scenario simulations are run, a base scenario simulation (in which no parameter or structure of the model is altered) representing the validated model based on the case study is run. Third, a summary of this chapter is provided.

7.2 Scenario simulations

Within the context of SD modelling, a scenario simulation for analysing a policy is "an extremely important part of the modelling process" (Maani & Cavana 2007**:** 75). A scenario is implemented by altering the value of a (set of) parameter(s) and running a simulation on the validated model based on these altered values. The scenario could also be implemented by adjusting the structural relationships of the validated model such as by changing the graphical relationships, linking model elements which previously were not linked, or deleting or adding elements from or to the model (Maani & Cavana 2007). The dynamic behaviours resulting from the simulation run of a scenario are then observed and analysed.

In this study, a policy can mean a success factor or a structural relationship of success factors which has been indicated by the case study or previous studies to influence e-government success or failure. By running this simulation scenario on the validated model, it will be revealed what factors and their relationships lead to e-government success or failure and how they do so, otherwise explanations could be sought from the model as to why a certain level of success cannot be realised. Among the benefits of this simulation is that a policy can be exercised to help decision-makers to review the dynamic impacts of the policy without costing any resources or affecting current organisational settings (Morecroft 2007; Sterman 2000).

For this study, in order to run scenario simulations it is necessary to identify various possibilities that might occur or have already occurred. However, considering the number of model parameters and structural relationships as well as the number of different combinations between them, the number of possible situations can be enormous. Running simulation of a large number of possible situations can result in a large number of simulation outputs which will be very complicated and difficult for interpretation, hence less useful for e-government decision-makers or managers. For example, to give an idea of these numerous possibilities section 6.4.10, which describes behaviour sensitivity tests, has indicated a large number of simulation runs that have to be performed when the value of only one parameter of the model is altered and other values are kept at their original values.

So, in order to make scenario simulations meaningful and useful it is necessary to narrow down the scope of the scenario by reviewing the aim of this modelling research. As described in Chapter 5, the SD modelling aims to give insight into the importance of complex feedback relationships of the eGSFs in influencing e-government success and to show why and how these relationships lead to e-government success over time from the perspectives of SM and IS staff management. Using a different side of the same coin and referring to previous studies, scenarios will be simulated to explain why a (set of) particular parameter(s) and/or some structural relationships are important for e-government success, or otherwise lead to an e-government failure. The scenario will be evaluated mainly based on the software availability over time.

For this, a list of scenarios is devised and presented in Table 7.1. For each of these scenarios, other parameter values and relationships of the base scenario are kept in place unless stated otherwise.

Table 7.1 *A list of scenarios*

Scenario	Description
Base	The validated eGSMA SD model
Recurrent fault increase	Increase the recurrent fault percentage in the base scenario to emphasise the importance of impact analysis
Quality of developed software	Adjust dynamic level of MRs in the base scenario to reflect the role of quality software resulting from development project within SM perspective
Rewards and staff turnover dependency	Construct a structural relationship between rewards and staff turnover factors in the base scenario to describe dependency of these two factors
Rewards, absence level and work morale dependency	Constructing structural relationships between rewards, staff absence level, and recurrent faults factors in the base scenario to represent dependency among these factors
Lack of competence level and training	Lower the average competence level as a result of inability of the organisation to recruit competent staff due to low wage offered and job characteristic perceptions, and reduce the average improvement level per training and the training frequency of the base scenario to reflect the lack of funding.
Reallocate idle staff	Construct a link between maintenance requests and other duty levels in the base scenario to describe a possibility to re-allocate idle IS staff to other tasks when maintenance workload is at low levels.
External expertise	Adjust the value of environmental factors in the base scenario to reflect the impact of high dependency on the external factors.

7.3 Base scenario

7.3.1 Characteristics

A formal model representing dynamic relationships between eGSFs has sufficiently completed a validation process in Chapter 6. This model, for which the parameters and structures have been verified against a successful e-government system case study (the eGSMA), is termed the “base scenario” and it is on this base scenario that parameter values and relationship structures will be varied and simulated to explore how a possible situation might dynamically influence e-government success or lead to e-government failure. Despite the complete listing of parameters and structures of the base scenario provided in Appendix E, the following list summarises general characteristics of the base scenario. Referring to model version numbering used in the previous chapter, this base scenario refers to the model version 2.1.

- Overall initial system size is 100,000 SLOC, which is relatively small, then increases 10,000 SLOC each year from third year due to new version releases.
- Total increase of SLOC for each maintenance delivery on average is 15.
- Following the characteristics of close-developed software, the MRs of this type of software have a general decreasing pattern over time with a lump of MRs soon after new version releases.
- The total of quarterly MRs that are the total of corrective and enhancive MRs each quarter is equal to the faults estimated from the SLOC of the system.
- Average dynamic maintenance performance is around 95% which also reflects the level of dynamic software availability.
- Recurrent faults level is 0.5% and dynamically improves (goes down) due to learning and training gained by the staff.
- Staff spends minimum 0.875 and maximum 1.5 of official working hours (8 hours) per day when undertaking maintenance tasks.
- The relationship forms between the delivery and request ratio and both organisational rewards and staff expectation of reward are scaled S-type graph.

- Communication level with other experienced staff is set to 0.95 and with users is set to 0.93 and is constant over time, which means the communication goes relatively smoothly.
- Initial competence levels of competent staff and less competent staff are set to 3.7065 and 1.5885 tasks / person / quarter respectively.
- Decay factor that reduces the average competence level of the staff is approximated using Moore's Law of computer processor capability which doubles every about two years, or 0.0625 per quarter.
- Training is conducted once in a quarter for all staff, which can improve staff competence level by 25%.
- In determining the level of training needed, the management is assumed to give weight to average competence level, fraction of competent staff and recurrent faults factors as 75%, 20%, and 5% respectively.
- Initial values of the less competent staff and competent staff are eight and six persons respectively.
- Base improvement adjustment time from less competent to competent staff is 26 quarters and can be faster due to learning and training.
- Staff recruitment process is only for less competent staff, occurs once a year and is limited to one person.
- Current total staff responsible for SM is 16 and is considered sufficient.
- Staff spends time in organisation for 30 years.
- Average staff assigned to carry out other maintenance tasks and tasks from VIPs are set to 30% and 7.5%, respectively and are constant.
- Maximum absence level is 20% and decreasing based on the difference between organisational rewards and staff expectation of rewards.
- The simulation is run for 20 quarters or 5 years.

7.3.2 A simulation run

After running a simulation on the base scenario, various selected outputs have been obtained and will be analysed and discussed in this section. The discussion aims to show how one factor dynamically relates to other factors in influencing eGSMA's success. This is, therefore, to indicate the importance of not only the success factors but also the relationships between them. In addition, the discussion also tries to identify possible high leverage factors for e-government success in the eGSMA.

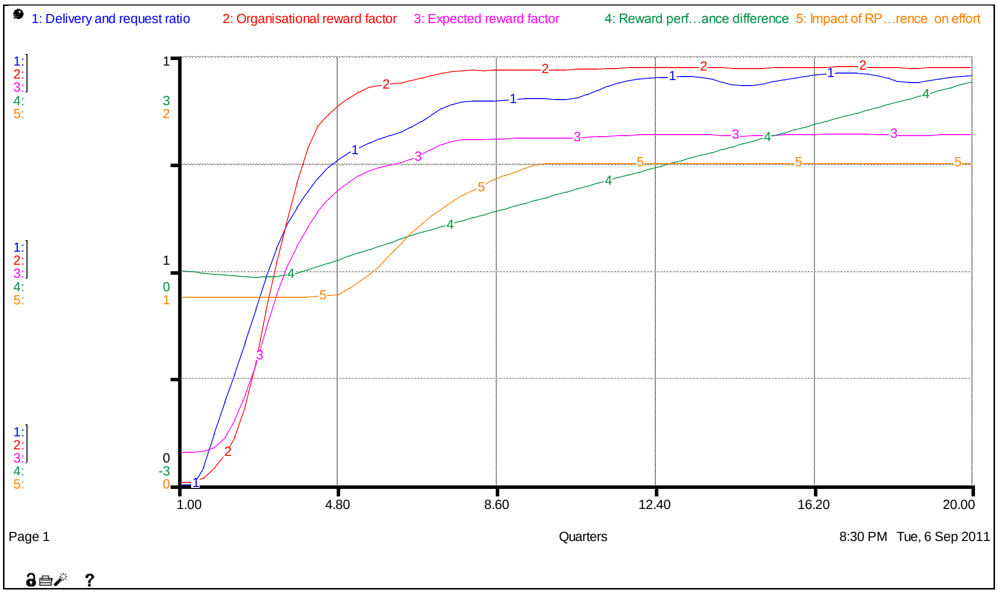


Figure 7.1 *Dynamic relationship between performance rewards difference and effort of eGSMA staff*

Performance – rewards – efforts. The dynamic influence of the SM performance on rewards and then on effort can be observed in Figure 7.1**.** Except during the early period of simulation, the levels of rewards provided are mostly higher than the expected ones, and this causes the difference between the cumulative level of these two factors to be always positive and increasing most of the time (line 4). This positive difference means the IS staff are willing to increase their exerted effort. However, the increase in exerted effort does not occur at the same time as the increase in the positive difference of rewards. This is possibly because the "orientation" of the staff on rewards is towards long-term rewards. They try to make sure that the rewards will be consistently provided. Once they realise that the provided rewards are consistently higher than their expectation, they increase their effort. As can be seen in line 5, this increase is up to a certain maximum limit, although the rewards difference is positive and increasing. Practical time limitation, work exhaustion, burnout and stress (Kim & Wright 2007; Mak & Sockel 2001; Rehman et al. 2011) mean the effort level cannot go beyond a particular maximum level. During early periods the rewards difference is negative, causing the IS staff to exert their effort at the lowest level. This dynamic level of effort emphasises the existence and importance of the relationships between SM performance, rewards and effort (Prasad & Suar 2010). In particular it underlines the critical role of rewards factors as shown by many of previous studies, for example (Hall et al. 2008; Rehman et al. 2011).

Further, this simulation output shows innovatively the change of effort levels as the difference between the provided and expected rewards changes over time. Also, possibly specific to the eGSMA, the IS staff never reduce the effort level from 0.875 (their minimum effort level) of the official working hours.

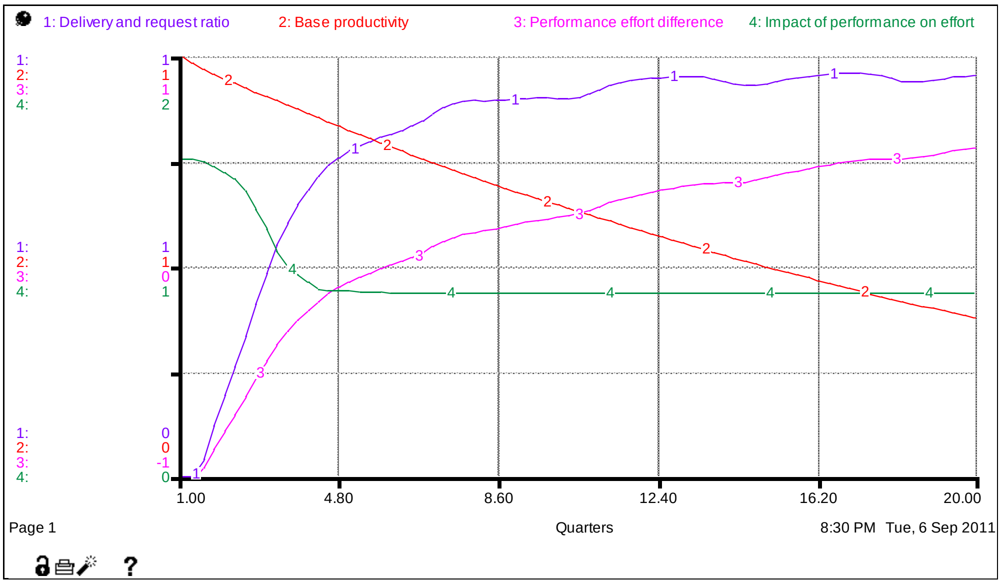


Figure 7.2 *Dynamic relationship between SM performance and effort of eGSMA staff*

Performance – effort. Dynamic relationships between SM performance and IS staff effort with respect to the IS staff base productivity in the eGSMA are depicted in Figure 7.2. Following the expectancy theory, the level of staff effort is determined by the difference between current level of SM performance (line 1) and normal or base productivity (line 2). The level of dynamic difference of these factors and the impacts of this difference are presented by lines 3 and 4 in Figure 7.2.

This figure shows that the difference between SM performance and base productivity (when the staff exert their base effort, that is 0.875 of official working hours, using their current level of competence) changes gradually from a negative to a positive level. The negative difference in early quarters causes the staff to exert more effort than the official working hours in this period. However, this effort tends to decrease to the lowest level, 0.875 of the official working hours, as they observe that the difference is positive. This is in line with the claim of the expectancy theory but with, innovatively, more descriptions in regard to the dynamic difference and impact. In contrast to the expectancy theory which does not specify a limit, the effort reduction by IS staff of the eGSMA will not exceed their minimum effort level. Practically, as claimed by the eGSMA staff, the staff will finish their maintenance tasks as soon as possible and do not delay the delivery of a completed SM task to meet the scheduled delivery time.

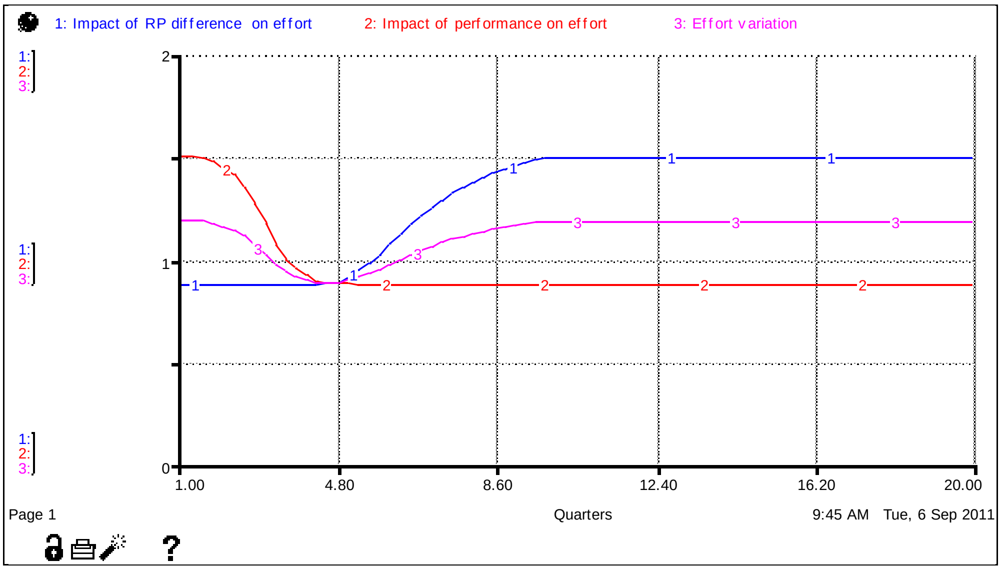


Figure 7.3 *Dynamic relationships between the rewards, performance and effort level of eGSMA staff*

Staff dynamic effort. The average of these two dynamic differences (performance – rewards – effort) and (performance – effort) is described by line 3 in Figure 7.3. This line represents the dynamic levels of effort that IS staff are willing to exert to undertake SM tasks as a result of the influences of the level of performance and rewards. The minimum effort occurs between quarters 4 and 5, the period when the staff observe that their performance is higher than that of normal effort and the received rewards are lower than their expected ones. Both factors cause IS staff to reduce their effort. It can be concluded from that figure that, in general, eGSMA staff tend or are willing to exert their effort more than the official working hours when they have to carry out a maintenance task. This fact arguably influences the capability of the eGSMA to maintain relatively high levels of its software availability over time.

In short, the IS staff of the eGSMA are highly motivated in carrying out their maintenance tasks. This high motivation level is caused by at least two things, as indicated by the case study data and the model: organisational (that is, the effective influence of provided rewards) and personal (that is, the IS staff like their jobs and are proud if “their” software can be used to deliver services and benefit other parties).

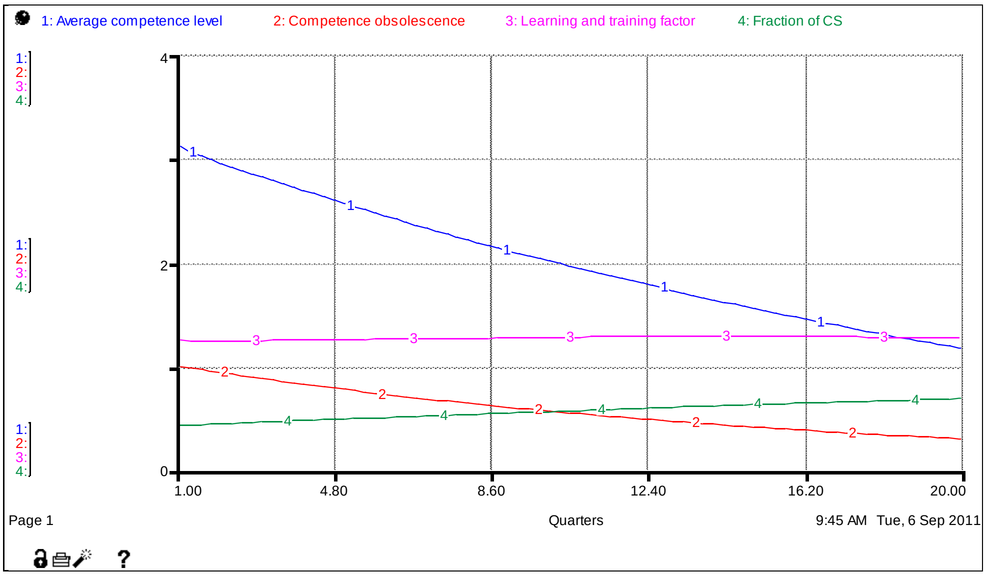


Figure 7.4 *Dynamic of competence level and its influencing factors of the eGSMA*

Dynamic competence level. The average staff competence level dynamic, which affects the staff productivity level, is continuously decreasing at a relatively fast pace, as shown by line 1 in Figure 7.4. The rapid IT advancement (line 2) causes the staff competence to become obsolete, which is indicated by a monotonic decrease. As can be seen, the learning and training factor (line 3) and the fraction of competent staff (line 4) which are expected to balance the effect of obsolescence tend to be constant or only slightly increase.

This shows that the decrease in the average IS staff competence level can be attributed mostly to the rapid advancement of IT (Joseph & Ang 2010; Lee & Mirchandani 2009; Tsai, Compeau & Haggerty 2007), which cannot be controlled by e-government decision-makers or managers. The e-government managers, however, can control how to cope with this natural degradation of competence. In this instance, the eGSMA has been on the right track in acknowledging the significance of skills training for its IS staff. It can be argued that if the eGSMA does not provide training to its IS staff, the average competence level decrease is very likely to be faster.

The relatively fast decrease in the competence level of the eGSMA staff resulting from the model can also be explained by the fact that in this model all newly recruited staff are categorised as less competent staff. Therefore, the ability of the eGSMA to recruit competent staff directly could see the decrease in the average staff competence level reduce. In addition, the association of competence obsolescence with computer processor power advancement in the model might also overstate the actual competence level decrease. Research conducted by Allen and Velden (2002) in the Netherlands found that tertiary education skills in a number of different occupations reduce to half in between 10 to 15 years, while in the model the "half-life" of the staff competence level is two years.

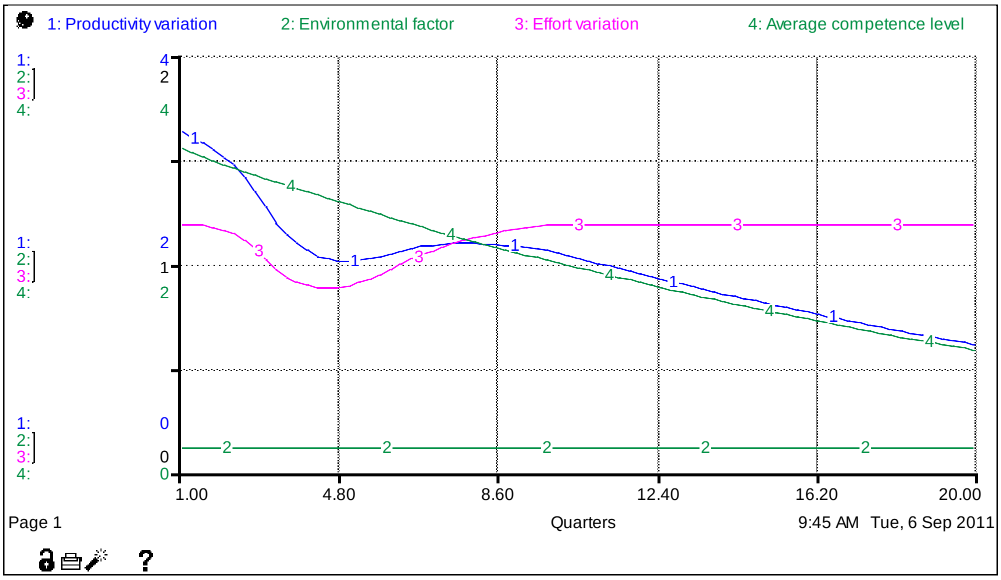


Figure 7.5 *Dynamic relationships between productivity, competence, effort and environmental factors levels of the eGSMA*

Dynamic productivity level. The average level of IS staff effort dynamically influences staff productivity. As shown in Figure 7.5, the dynamic level of staff productivity (line 1) in general exhibits a decreasing pattern although the decrease is not monotonic. Observing the behaviour over time depicted in Figure 7.5, the dynamic influence of the staff effort (line 3) and competence level (line 4) on the staff productivity is easily visible. A relatively easy communication with other staff to get support and with users to clarify requirements, as stated by the eGSMA staff, mean that the environmental factors do not significantly reduce staff productivity. As the effort level is influenced by rewards and performance, the change in the way the productivity level decreases is indirectly affected by the level of rewards and SM performance. It might be also be concluded that the staff effort level has been able to prevent the productivity level from steeper descent as a result of IT advancement.

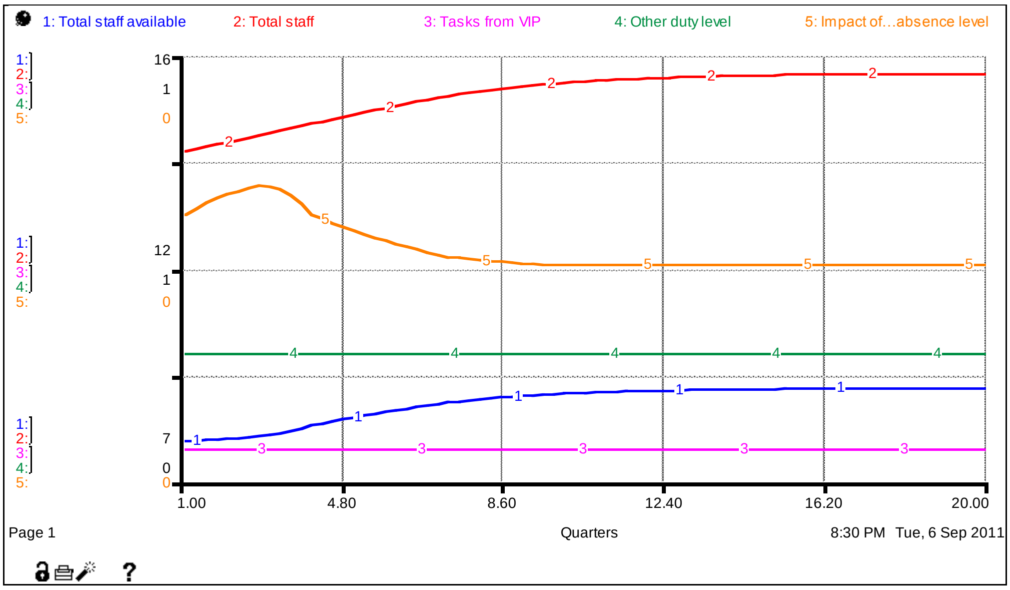


Figure 7.6 *Dynamic staff availability level for SM and its influencing factors of the eGSMA*

Dynamic staff availability level. The dynamic level of total IS staff of the eGSMA available to carry out maintenance task is described by line 1 in Figure 7.6 and is affected by some reducing factors. First, this level is constantly reduced by the level of the staff doing tasks from VIPs (line 3) and other duties (line 4). This reduction causes the level to be much less than the total IS staff. Second, as formulated, this level is also influenced by IS staff absence percentage levels over time, which in turn are affected by rewards. The dynamic impact of the difference between provided and expected rewards on the staff absence percentage level is described by line 5. During the early period, as the rewards difference is negative the absence level increases, reducing the total available staff which occurs around the third quarter. As the absence level decreases due to organisational rewards higher than the expected ones, the total staff available level again increases. The total IS staff (line 2) which is affected by the recruitment and exit rate, tends to level off after around the ninth quarter, which causes the total staff available level to level off also. It is interesting to note that in the eGSMA, the total IS staff and the total IS staff available for maintenance tasks do not fluctuate over time. This can be well understood because the affecting factors also do not fluctuate. For example, the IS staff turnover level is almost zero as the eGSMA staff spend on average 30 years in the unit, and the recruitment level of new staff is limited to one per year because the senior manager of eGSMA has considered that the IS staff number is sufficient. In addition, the maximum absence level of the eGSMA staff is relatively low, and other duties and tasks from VIPs can be reasonably managed at a constant level.

Compared with other organisations which are highly affected by significant turnover problems, this stable composition of highly motivated IS staff has been giving benefit to the eGSMA in ensuring its e-government success. This is consistent with Schwester (2009) and Ray (2010) in that sustaining stable full-time IT staff can significantly predict the level of e-government adoption or success. The way this factor benefits the eGSMA could be that knowledge can be retained within the eGSMA as a result of almost zero staff turnover (Martins & Martins 2011), enabling it to have a higher probability of achieving e-government success; with a small number of newly recruited IS staff, the eGSMA focus is not distracted away by new staff improvement in various aspects; better IS staff teamwork with better performance can be established because the IS staff can maintain good communication levels and knowledge-sharing through mutual trust (Sudhakar, Farooq & Patnaik 2011).

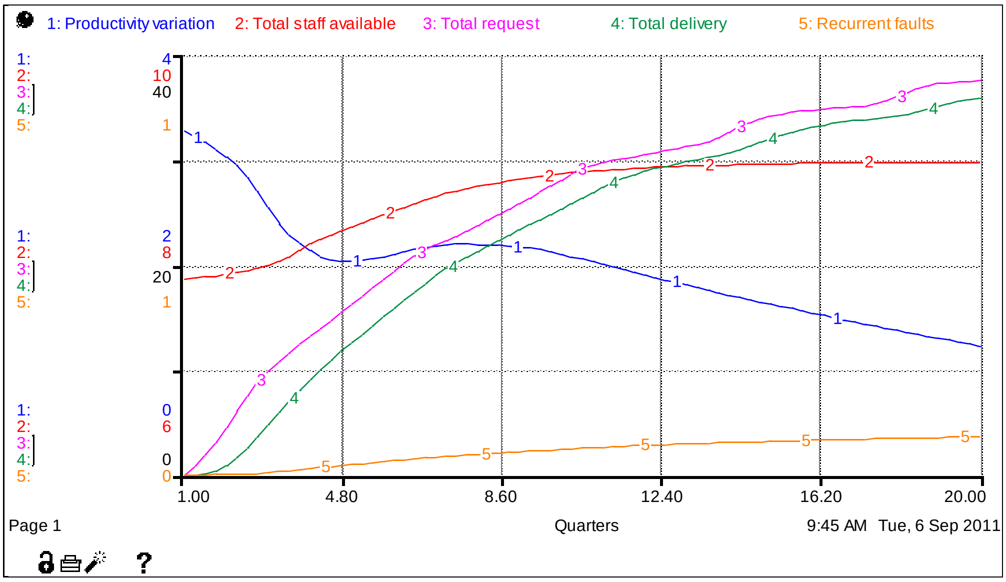


Figure 7.7 *Dynamic relationships between staff productivity, staff availability and total delivery of the eGSMA*

Dynamic of staff productivity, availability and SM delivery. IS staff productivity variation and the total actual number of IS staff available to undertake SM tasks dynamically influence the total delivery of SM. Behaviours over time of these factors are depicted by line 1, 2 and 4 in Figure 7.7.

By observing this figure, one might conclude that although the productivity variation level tends to decrease over time and the staff availability level increases only slightly, the delivery level is able to closely follow the pattern and level of cumulative total requests. It can be interpreted that the MRs that turns to maintenance tasks can be completed and delivered, although understandably there is a time lag between the request and delivery time.

In sum, the simulation of the model has suggested that the ability of the eGSMA to sustain the necessary level of total staff available for SM, to avoid steeper decreases in staff productivity and to preserve a low level of recurrent faults arguably influences the dynamic of the current level of completed maintenance delivery. This in turn determines the dynamic of the high level of software availability or SM performance. As was argued in a previous chapter, Chapter 5, this level of software availability is used as the operationalisation of the eGSMA's success. The model and its simulation run have shown how dynamic relationships of endogenous factors determine eGSMA success. This focus on endogenous factors does not necessarily imply a disregard of the important role of the dynamic nature of MRs, which among other things reflects the quality of software resulting from a software development project, as well as other exogenous factors for eGSMA success.

This model arguably also indicates that other e-government systems which have similar characteristics with the eGSMA but with "lower" parameter values, or "worse" success factors relationships structure might lead to an e-government system failure. This perspective is explored and simulated through some possible scenarios (Table 7.1) in the following section.

7.4 Recurrent fault increase scenario

This scenario is introduced to underline the importance of impact analysis for a successful SM delivery, as described in sections 3.3.4 and 3.3.7. Because successful SM deliveries are needed to ensure the highest level of e-government software availability over time, this scenario also emphasises the necessity of impact analysis for e-government success. An impact analysis is an activity conducted during the maintenance process in order to minimise ripple effects or recurrent faults associated with delivered SM. It is needed to ensure that dependency among software components has been properly addressed and avoids, for example, dangling software elements (de Boer et al. 2005). Therefore, a deficient impact analysis means a higher level of probability that other software components related to those being corrected or enhanced are not properly evaluated and tested. This can trigger other faults and cause a higher level of recurrent faults in the future. As stated in section 3.3.4, the percentage of recurrent faults also measures the quality of SM delivered.

For this scenario, the base value of the *Recurrent_fault_percentage* is altered into two different levels, which are higher than the base value, that is 10 and 20 times higher. This percentage is chosen because it determines the recurrent faults rate. These settings cause the starting value of the *Recurrent_fault_ percentage to* become 0.05 and 0.1 respectively when the *Learning_and_training_factor* is 1. This scenario is performed by multiplying *Recurrent_fault_percentage* with a converter named *Multiplier* with 1 as its initial value (see Figure 7.8). The following equations are used to adjust the faults rate for CM and EM:

*CM_fault_rate=CM_delivered*Recurrent_faults_percentage*Multiplier*

*EM_fault_rate=EM_delivered*Recurrent_faults_percentage*Multiplier*

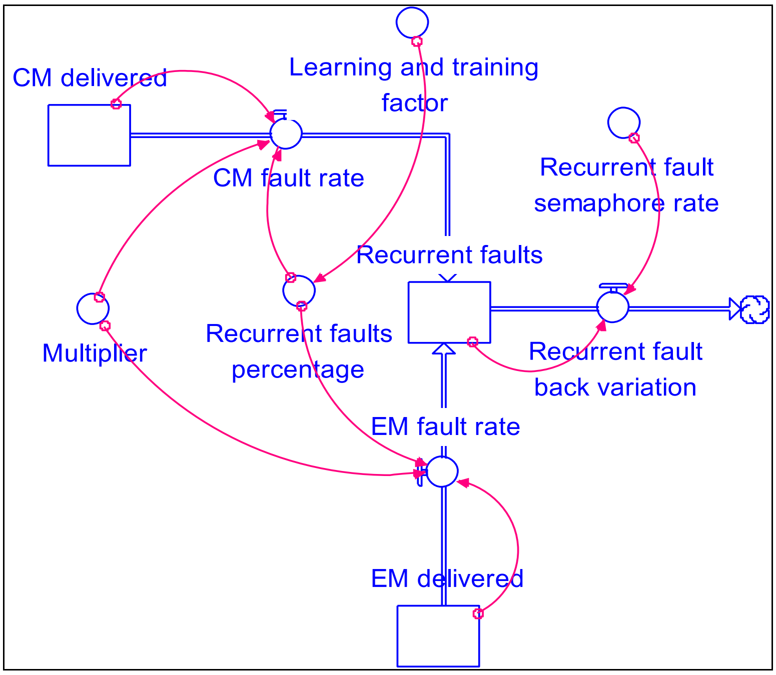


Figure 7.8 *A partial view of the modified model which alters the percentage values of recurrent faults*

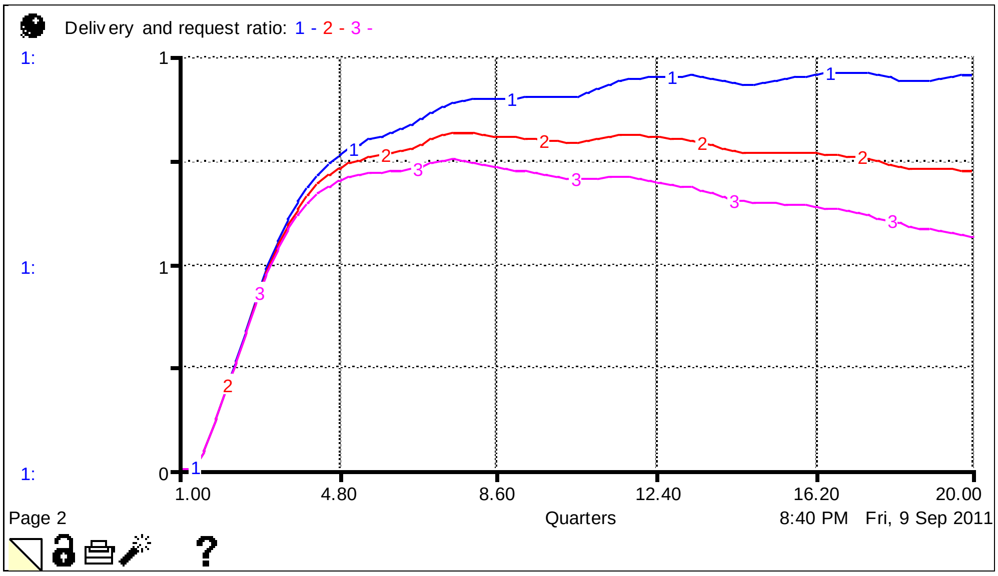


Figure 7.9 *The software availability level under three different values of the recurrent fault percentage*

Running the simulations of this scenario results in Figure 7.9. which shows the dynamic impacts of low levels of SM quality towards software availability levels. The delivery and request ratio indicates a continuous decreasing pattern. Figure 7.9 indicates the dynamic behaviour of the software availability level under three different settings: 1 – base value, 2 – base value * 10.5, and 3 – base value * 20. As predicted, the ratio or software availability level is lower when the recurrent faults level is higher. Thus, a low level of maintenance quality which persists over a long time period can cause the failure of e-government failure due to decreasing software availability levels.

Referring to the CLD presented in Figure 6.3, a reinforcing feedback loop relationship can be traced between recurrent faults and MRs. This reinforcing feedback loop relationship can be observed in Figure 7.10 as well as in Figure 7.9. It is indicated by the increasing distances between the base level (line 1) and other adjusted levels (lines 2 and 3) of the MRs and the recurrent faults percentage over time.

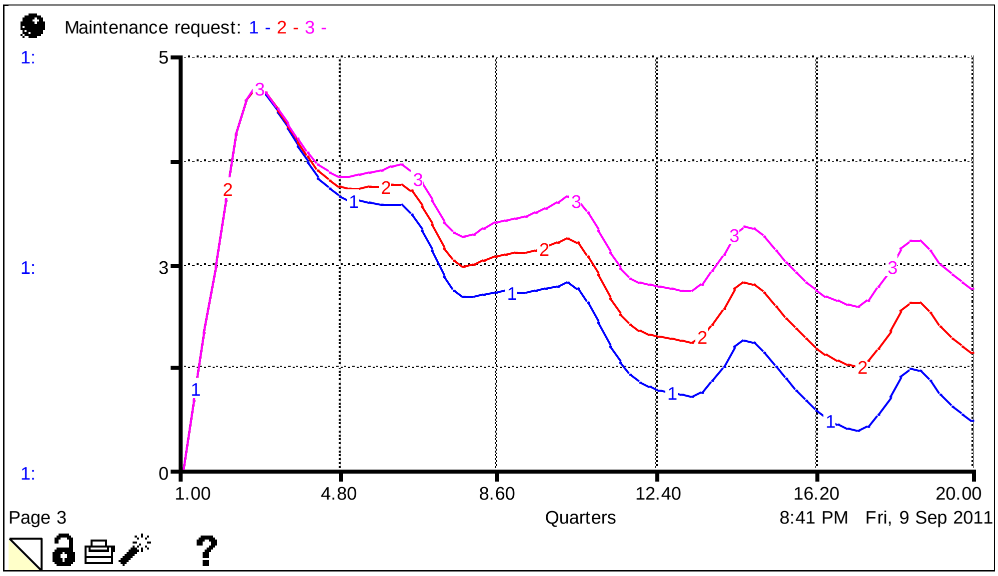


Figure 7.10 *The maintenance request levels under three different values of recurrent faults percentage*

Thus, this simulation run has shown the importance of undertaking impact analysis during SM. A lack of impact analysis causes an increase in the recurrent faults which in turn brings about an increase in MRs. This rise of MRs means a higher SM workload which makes the impact analysis more difficult within the current available resources. This in turn inclines recurrent faults gradually towards higher levels.

From the software availability level perspective, the necessary role of SM impact analysis on e-government success can be intuitively comprehended. For most of the SM being undertaken, the levels of software availability will be reduced in two ways: unavailability of service components required (Gaj & Germani 2008) and unavailability of the software during maintenance (Buckley et al. 2005). Additionally, one deficient maintenance, either for correction or enhancement, can cause more than one future fault. In turn this causes MRs to increase, meaning that the number of services unavailable for users also increases, lowering further the level of software availability.

In sum, this scenario underlines what has been suggested by previous SM and e-government research with regard to the importance of impact analysis and the negative impact of the quick-fix approach. It also suggests the importance of documentation for SM as it is required for undertaking impact analysis. Also, if the SM management is far from a maturity level, as might be found in many developing countries, it might be better to assign internal software developers an SM job, as in the case of eGSMA, to ease impact analysis because of their familiarity with the software. For the same purpose and argument, it might be better to develop small-scale software but operative as intended, and then incrementally improved, as is also the case with eGSMA.

7.5 Low-level quality of developed software scenario

This scenario simulation is intended to show the significant role of the quality of software resulting from a software development project (van Vliet 2008, Ch. 6), which is an exogenous factor in the validated model. The quality of software is denoted by the dynamic level of MRs. A higher level of and relatively constant dynamic MRs than the base one could mean lower-quality software, assuming the pressure from environment for software enhancement is relatively low and constant over time. This low-quality software can be attributed to a high level of software faults over time because of careless development and a high level of enhancement requests over time due to many unsatisfied user requirements during development. By inferring from the relationship between software complexity and faults presented in section 3.3.4, a high level of MRs over time can also be interpreted as that the software size or complexity is very high. Usually, this is the case when the software development uses a “big-bang” approach.

This scenario is implemented by altering the base value of the *Task_multiplier* into two different levels, higher than the base value and changing the *Impact_of_time_on_NMR* level such that its dynamic value decreases only slightly over time. This setting causes the level of MRs to be higher than its base level and only slightly decreasing. The *Tasks_multiplier* is set to three and five times higher, and the *Impact_of_time_on_NMR* is set as described by Figure 7.11. Figure 7.12 describes the dynamic level of MRs when the *Tasks_multiplier* is equal to: 1 – base value (line 1), 3 – three times higher than the base value (line 2), and 5 – five times higher than the base value (line 3).

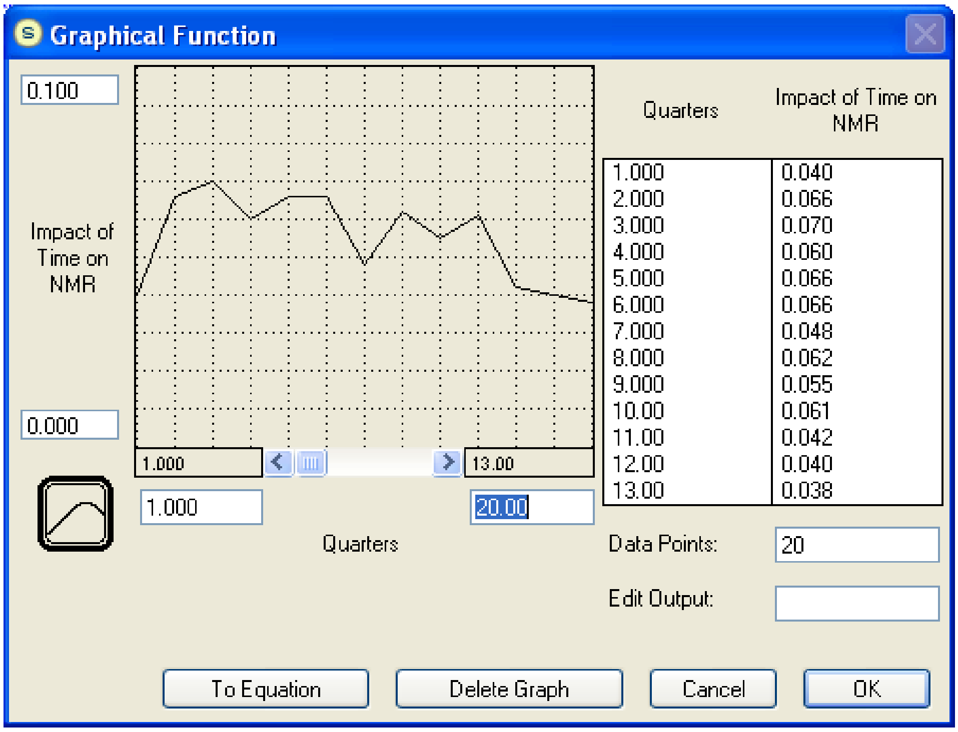


Figure 7.11 *The adjusted level of the impact of time on new MRs*

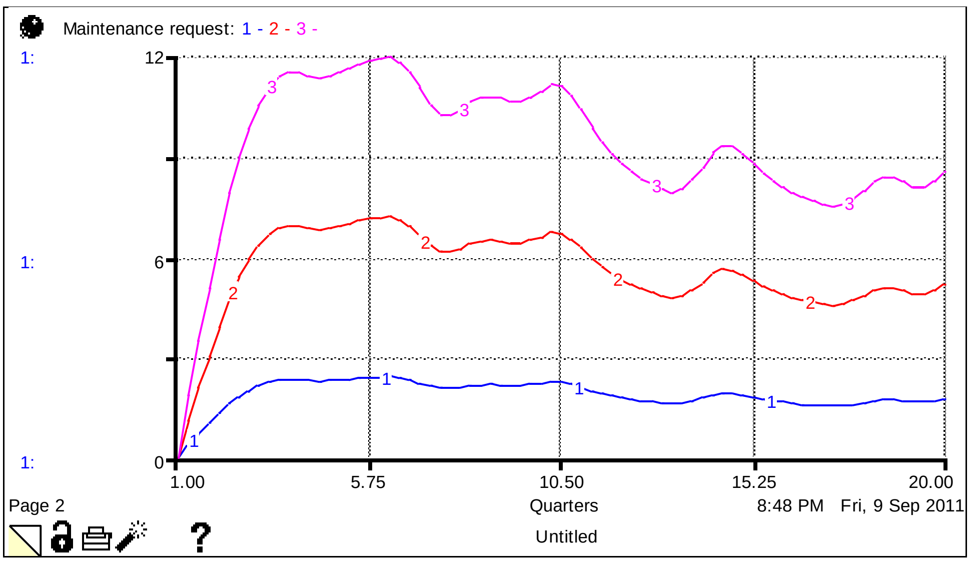


Figure 7.12 *Three different levels of MRs indicating three different software quality levels*

The delivery and request ratio levels over time as a result of the simulation of each of these settings are presented in Figure 7.13. This figure indicates that a high level of software MRs (line 3 of Figure 7.12) causes a low level of delivery and request ratio (line 3 of Figure 7.13).

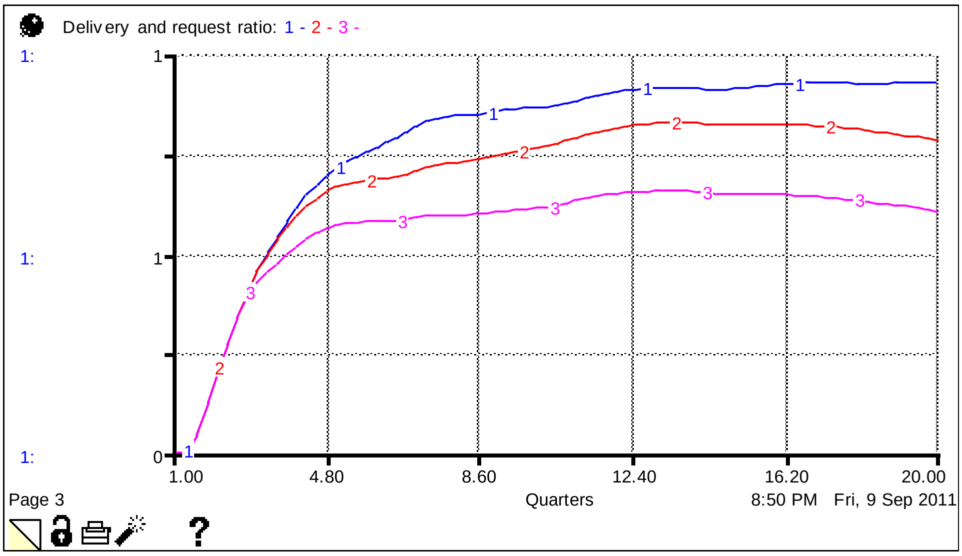


Figure 7.13 *The three software availability levels resulting from three different software quality levels*

This simulation indicates that within the context of the current system, software with low quality transferred to the operational and maintenance stage might cause difficulties for IS staff to maintain. This could be because the current level of available resources is insufficient and unable to cope with the dynamic level of MRs. Therefore, this simulation suggests explicitly the important role of the software development quality – small number of errors and satisfying user requirements, among other measures, which is indicated by a relatively monotonic decreasing pattern of MRs – towards software availability over time.

This simulation also suggests that building large and complicated e-government software in one project is undesirable (Goldfinch 2007), because a large size and complex software is more likely to create more error (Ostrand, Weyuker & Bell 2005), while on the other hand the available SM resources cannot cope with the degree of MRs. This also indicates that the eGSMA's approach of starting by building small-scale systems and in incremental manner is more appropriate with respect to the resources available for SM, which was also recommended for developing countries by Heeks (2005b) and Ray (2010).

7.6 Level of rewards and information systems staff turnover dependency scenario

This scenario assumes that the value of provided organisational rewards is less than the staff expectation of rewards. As distinct from the eGSMA that has never experienced IS staff turnover since its establishment about 20 years ago, in this scenario the rewards in various forms, intrinsic and/or extrinsic, influence IS staff turnover, as observed in much previous research (Kim 2005; Moore 2000). Some examples are an e-government project in Cambodia and telecentres in India, as presented in section 2.5.2, which cannot retain their competent IS staff for a long time because the IS staff have perceived the rewards received are less than their expectation. In these examples, the competent IS staff needed by the systems left those systems early, and the systems were unable to replace and retain them. These e-government systems were finally ceased.

For this scenario, the relationship between delivery and request ratio and organisational rewards is set up as in Figure 7.14. This graphical relationship accommodates the assumption that the rewards are much lower than the staff expectation, as set up in the base scenario. This perception influences IS staff turnover, in the sense that they leave current jobs in much less than the normal employment time. The relationship between the difference of provided and expected rewards and IS staff turnover is described in Figure 7.15. It is also assumed that the staff from both categories tend to exit with the same probability level. In addition, to implement this scenario, an outflow from *Less_competence_staff* stock is introduced; and the reward performance difference is now linked to the IS staff turnover of both categories as shown in Figure 7.16. New equations and graphical relationships:

Less_CS_exit_rate= Less_competence_staff Exit_adjusment_time*

CS_exit_rate= Competence_staff Exit_adjusment_time*

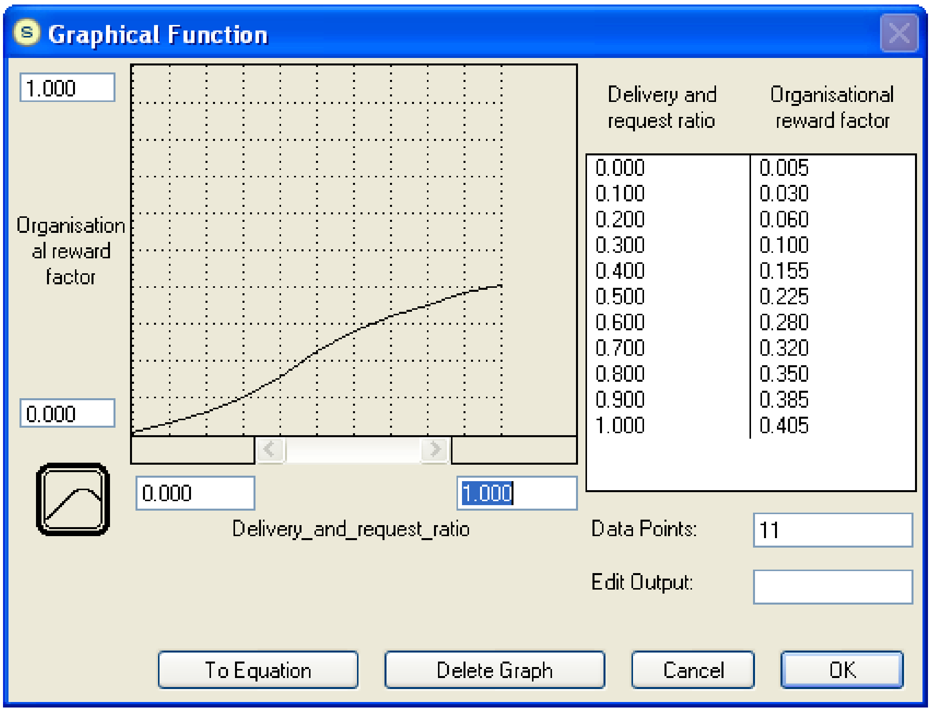


***Figure 7.14** Adjusted relationship between SM performance and provided organisational rewards*

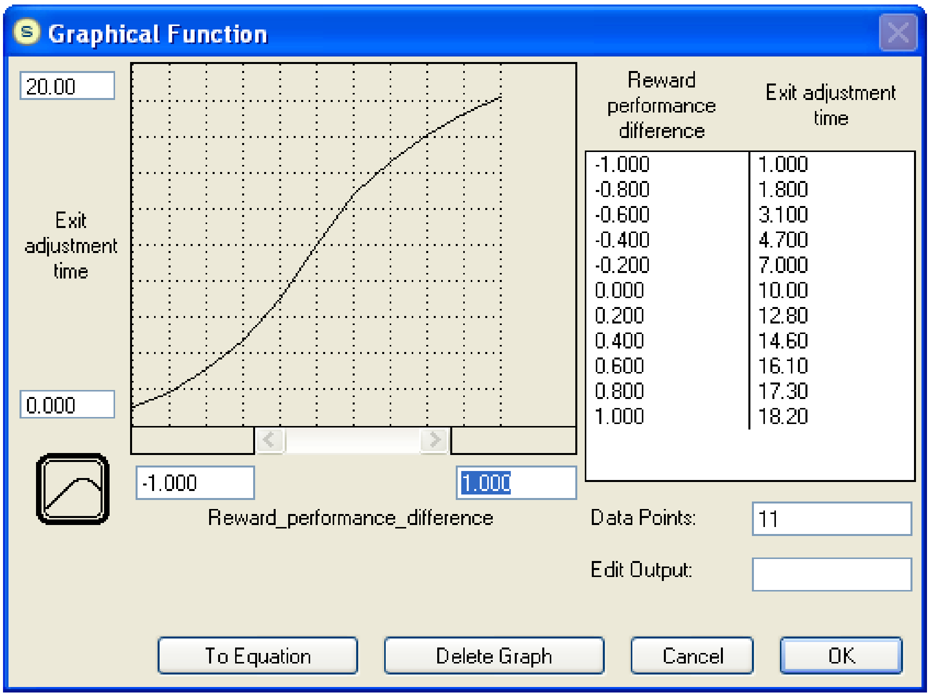


***Figure 7.15** A relationship between the difference of rewards and SM performance and employment time*

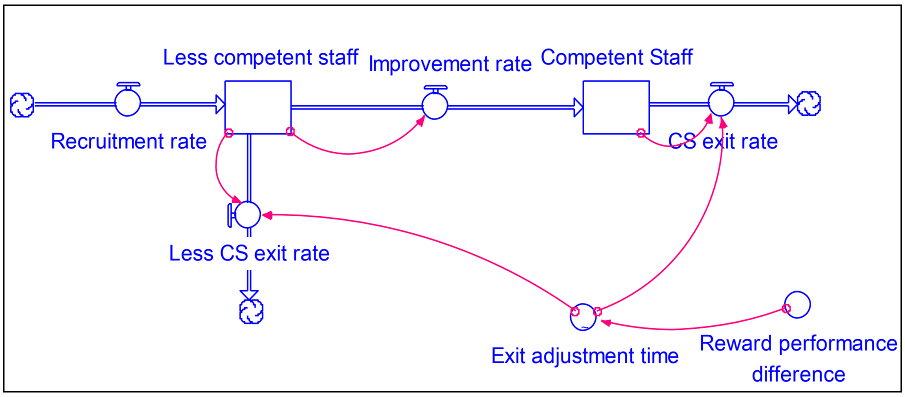


Figure 7.16 *A view of the partial model after some model components are added*

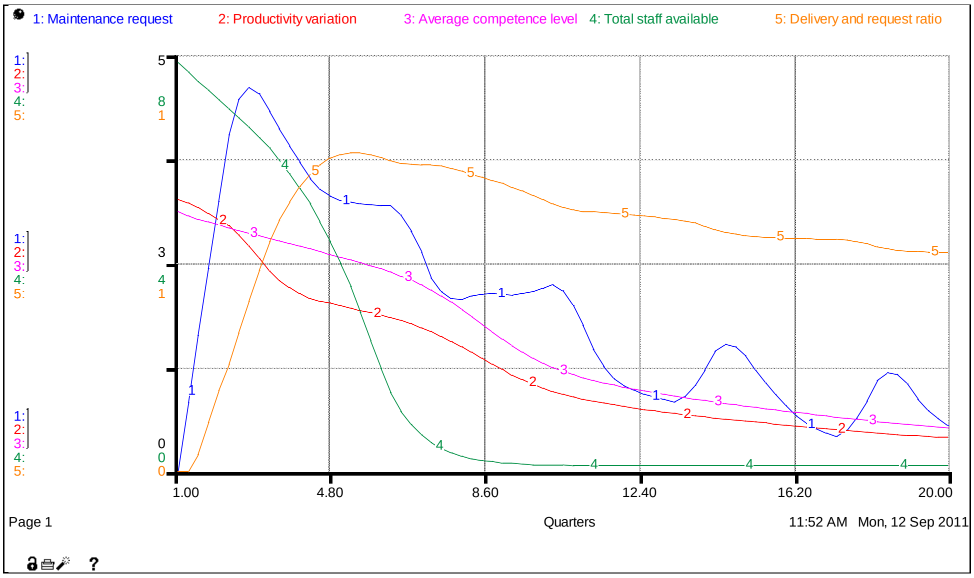


Figure 7.17 *Dynamic level of software availability and other key factors resulting from scenario 3*

The simulation based on this scenario suggests that although the level of MRs keeps decreasing (line1 of Figure 7.17), e-government success may not be achievable. Figure 7.17 shows that the software availability level (line 5) is continuously decreasing over time after around the 5th quarter, in addition to its inability to achieve a high level during the early period. However, it should also be noted that this level does not suddenly drop, which can be interpreted to mean that current functionalities of the e-government software can be utilised to deliver services, but as software errors emerge and enhancement requests keep coming, the level of software availability decreases. The decrease is not only because of the decreasing availability level of the IS staff (line 4) but also the decreasing level of existing IS staff productivity level (line 2). By tracing to the chain of relationships in the model, the decline of productivity level is caused by the decrease in the average staff competence level and staff effort. However, as the base effort is assumed to be equal to that of the eGSMA, the rest of the staff maintain the effort level at 0.875 of the official working hours (line 4 of Figure 7.18).

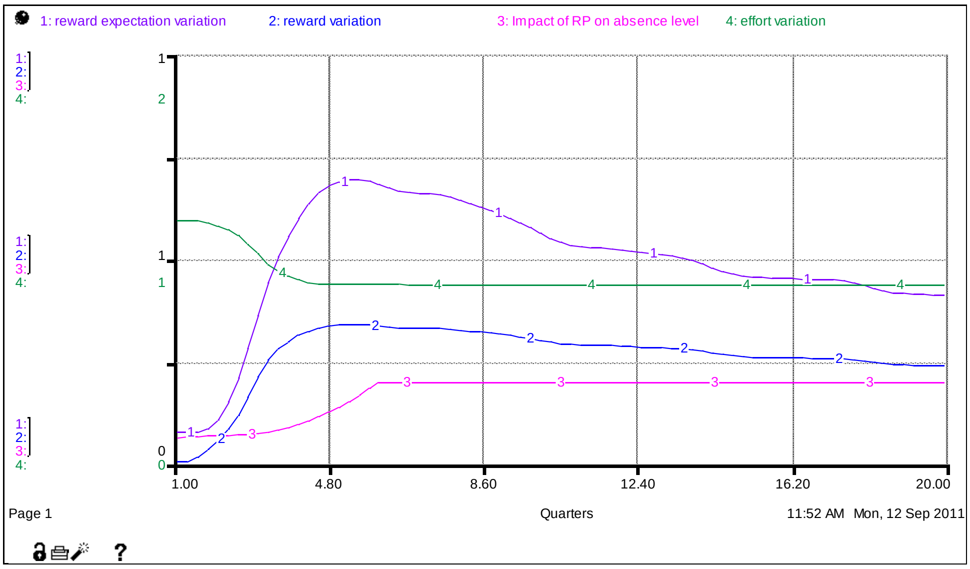


Figure 7.18 *A dynamic relationship of the rewards performance difference, staff absence level and effort*

Further, the decreasing level of staff availability can be checked from persistent and significant negative difference between the level of reward and expected reward variation (lines 2 and 1 of Figure 7.18), which according to this scenario causes a high level of staff turnover. This negative difference also affects staff absence level, although, as in the base scenario, the absence never goes beyond a particular level.

Accordingly, as might be expected, the dynamic relationships of this scenario suggest that, if the received reward is perceived consistently much less than the expected one and the staff prefer to pursue other jobs that offer more rewards, then eventually e-government success is less likely (Coombs 2009). However, the gradual decrease in the software availability might indicate that the staff turnover might be less problematic for SM in contrast to that in software development projects (Hall et al. 2008), as long as e-government managers are able to keep other factors as in the eGSMA model. The software availability level that does not decrease sharply also provides an opportunity for managers to reverse the level's direction in the long term.

7.7 Rewards, absence level and work morale dependency scenario

As distinct from the eGSMA, this scenario assumes that the provided organisational reward is less than the staff expectation of rewards; and this negative difference affects staff effort, absence levels and morale as observed in much previous research into public service in developing countries. This case might occur when the organisation or e-government managers only provide rewards according to the stipulated regulation, which is relatively lower than for external organisations based on comparison of similar jobs requiring similar expertise. In addition, the IS staff consider themselves as having invaluable competence with respect to the organisation and accordingly they consider the received rewards are much less than their expectation as indicated in the literature review presented in section 3.4.7, while at the same time they do not want to leave their current jobs as public servants, such as in the case of public service doctors in Indonesia (Chernichovsky & Bayulken 1995). They may consider working as a public servant to be a very secure job (Houston 2000, 2011) because they are guaranteed a lifetime career (Johnson 1999) which is also supported by the eGSMA staff, especially in volatile economic conditions As a consequence, they "moonlight" to earn additional income from outside jobs which means an increase in absence levels (Chernichovsky & Bayulken 1995). At the same time, the morale level of staff becomes low in the sense that when they are working in the office, a significant portion of official working hours may be used to perform their outside jobs; therefore the SM quality is sacrificed. As shown in scenario 1, a lack of completed SM quality means a high level of recurrent faults because the IS staff do not perform impact analysis properly. This situation is in sharp contrast with that in eGSMA, where staff morale in undertaking their jobs is high.

This scenario is distinct from the associated success factors and relationships of the eGSMA in which overall provided rewards are higher than staff expectation; IS staff never reduce their effort to less than 0.875 of the official working hours although some of them do have outside jobs; and the IS staff like the job in such a way that they take pride in the fact that the eGSMA systems users can be satisfied.

To accommodate the scenario, the relationship of the delivery and request ratio and the organisational rewards is as described in scenario 3, Figure 7.14. The relationships between this difference and effort and between the performance and effort are accommodated by the S-type graphical relationships of *Impact_of_RP_difference_on_effort* and *Impact_of_performance_on_effort* with 0.4 as the lowest limit. These are represented by Figure 7.19 and Figure 7.20 respectively, describing that if the differences become more negative then the staff effort becomes lower. The impact of the difference in the provided and expected rewards on absence level is now multiplied twice as high as the base level; therefore, this affects the staff availability level via the following adjusted equation:

Average_availability_per_staff =
*MAX (0, 1-(Impact_of_RP_difference_on_absence_lvl*2 + Other_duty_level + Tasks_from_VIP)).*

In addition, the influence of the rewards difference to the maintenance quality is represented by a graphical relationship, Figure 7.21, and structural relationships as follows:

CM_fault_rate = CM_delivered(Recurrent_fault_percentage + Impact_of_RP_difference_on_recurrent_faults)*,

*EM_fault_rate = EM_Delivered * (Recurrent_fault_percentage + Impact_of_RP_difference_on_recurrent_faults)*.

This graphical relationship and the above equations describe that as the difference between provided and expected rewards becomes more negative, the fault rate becomes higher. The subset of the model describing modified structural relationships is depicted by Figure 7.22.

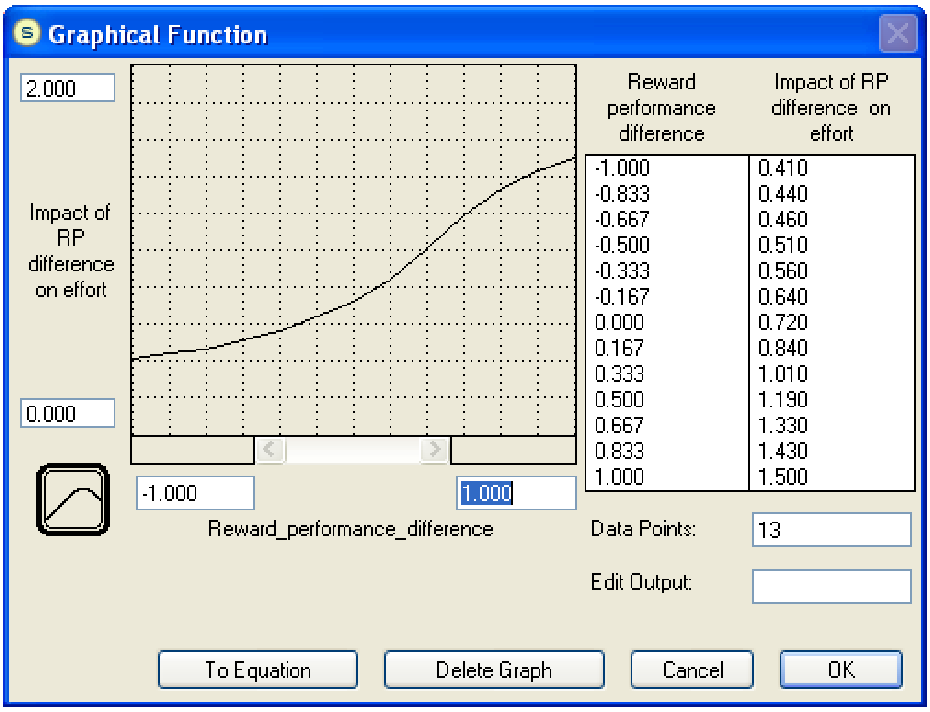


Figure 7.19 *An adjusted relationship between rewards-performance difference and effort*

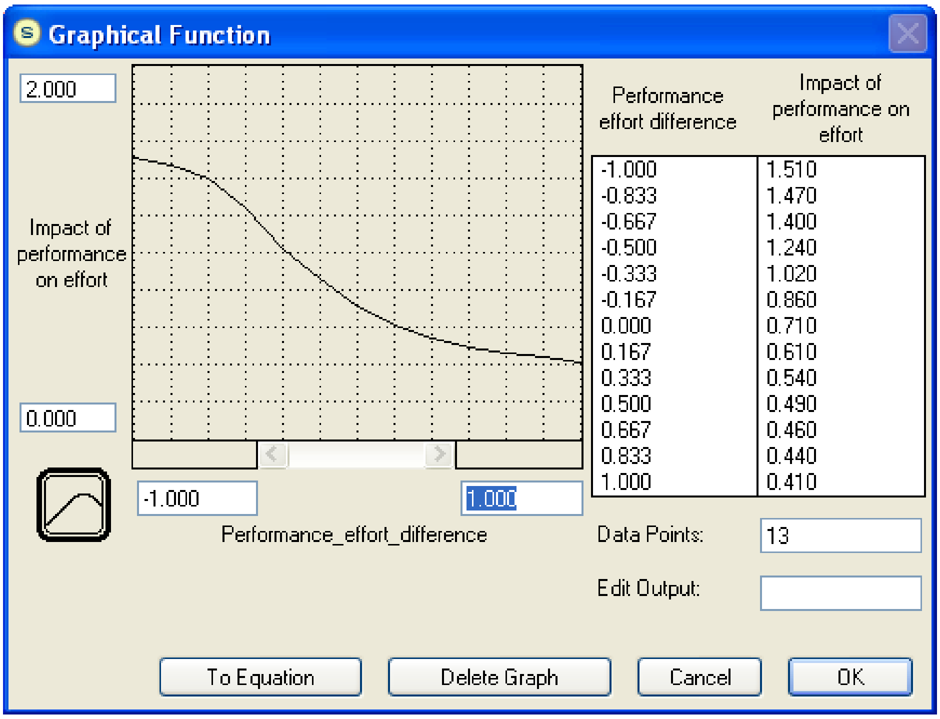


Figure 7.20 *An adjusted relationship between performance-effort difference and effort*

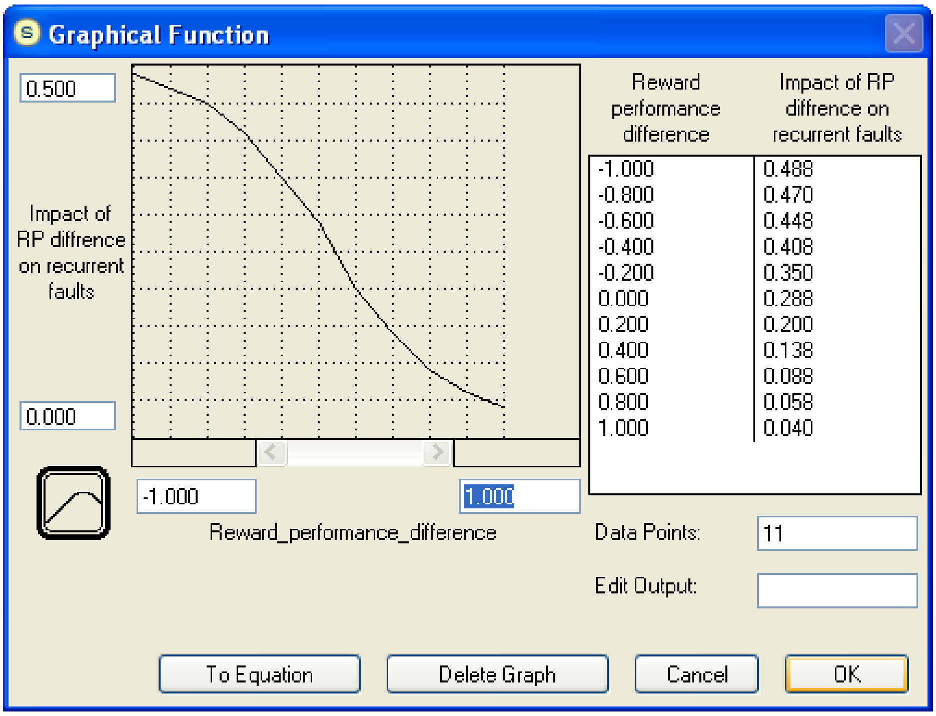


Figure 7.21 *A relationship between rewards-performance difference and recurrent faults*

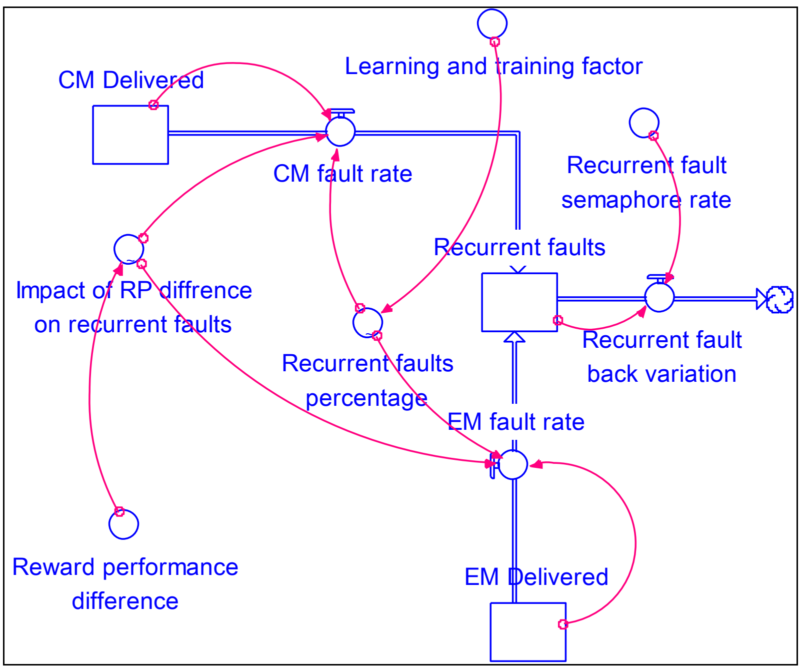


Figure 7.22 *A partial view of the model after a structural relationship between rewards-performance difference and recurrent faults is added*

The simulation of this scenario results in the level of software availability over time (line 5 of Figure 7.23), which only reaches a low "peak" at around the 5th quarter and then continuously decreases. This kind of dynamic behaviour can be traced from the dynamic of staff availability levels for SM (line 4) and staff productivity variation levels (line 2). It can be observed that both levels continuously decline; while on the other hand, the average of MRs (line 1) only very slightly goes down.

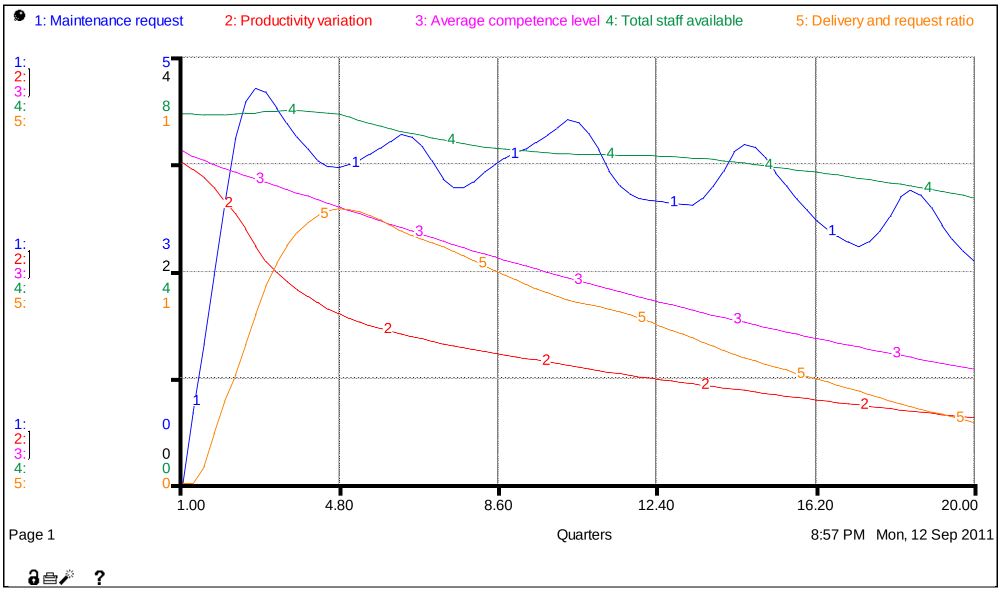


Figure 7.23 *Dynamic behaviour of the software availability level and other key factors resulting from scenario 4*

Further, through the relationships between the eGSFs in the model, it can be seen from Figure 7.24 that the decline of staff availability levels is affected by the increase of the proportions of absenteeism (line 3), which in turn are influenced by the negative difference of provided and expected rewards (line 1). This continuously more negative difference also affects the recurrent faults similar to the absenteeism level. Line 2, which describes dynamic behaviour of recurrent faults, indicates that as the difference is wider, the recurrent faults increase over time. This in turn causes the average of MRs to go down only slightly, which means the maintenance workloads never decline as in the base scenario. To make the case worse, also as a result of the widening difference, the average of staff effort also decreases although it levels off after around the 6th quarter (line 4). These relatively low effort levels over time along with decreasing staff competence levels (line 3 of Figure 7.23), also bring about a decrease in staff productivity.

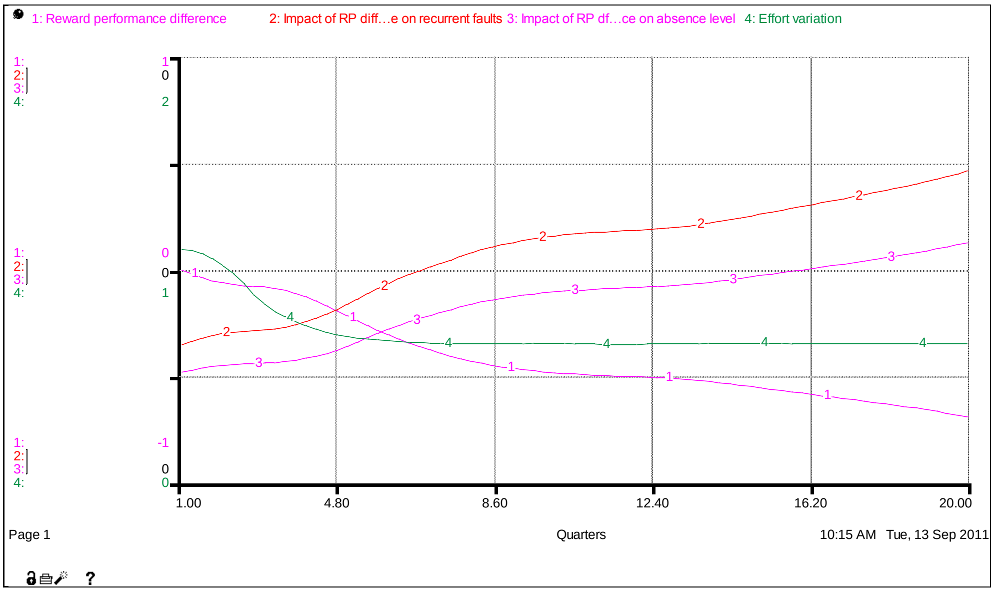


Figure 7.24 *Dynamic relationships between rewards-performance difference and recurrent faults, absence level, and effort*

Thus, if this scenario is experienced by an e-government system it is very likely that the ability of the system to deliver services will be in jeopardy; hence it will eventually be very difficult for the system to achieve success. This is so because of the low and declining level of the e-government software availability. Comparison between this scenario and those experienced by eGSMA reemphasises the importance of a policy in organisational rewards that can motivate IS staff to exert their effort as shown by much of the previous research, as discussed in section 3.4.3. In addition, this scenario simulation underlines the significant role of the staff who are keen on their jobs, which can prevent them from belittling their SM tasks, therefore avoiding unnecessary recurrent faults. In the case of eGSMA, the staff like their SM jobs and even continue to think about a task when at home, and are proud if their software can give benefit to other parties, which is in line with Coombs' (2009) findings. In addition, moonlighting, which is quite common for public servants who have specific skills in developing countries (Hyder & Ahmed 2011), was acknowledged by the eGSMA staff. However, as they stated, they moonlight after official working hours and even use it as a way of maintaining their skills level. Therefore, referring to the eGSMA case, moonlighting for IS maintenance staff might be allowable as long as the success factors as in the eGSMA case can be kept in place without sacrificing their SM tasks.

7.8 Lack of competence level and training scenario

This scenario deals with the inability of a e-government organisation to recruit high-calibre IS staff because of the low wages offered, bad perceptions of maintenance jobs and failure to implement training programs adequately to improve staff competence levels. One of the reasons for this inability is a lack of funding. Funding, in general, has been considered one of the most significant factors in determining e-government success (AlShihi 2006; Schwester 2009) and, specifically, e-government sustainability (Best & Kumar 2008; Best, Thakur & Kolko 2009). Lack of funding is mostly experienced by developing countries. In many cases, the e-government units in developing countries are unable to compete with private sectors that offer more financial rewards to competent IS personnel (Sang, Lee & Lee 2009); therefore only IS personnel with relatively low competence can be recruited by e-government system units. Inability to recruit highly competent IS staff might also occur because of the SM job characteristic, which is generally perceived by IS personnel as much less “glamorous” than the software development jobs, as mentioned in Chapter 3, section 3.3.8. In addition, necessary training for IS staff, which could be expected to increase IS staff competence significantly, cannot be provided by an e-government system unit because of lack in funding.

To accommodate this scenario, the initial values of the *Competence_level_of_CS* and *Competence_level_of_LCS* are set to half of the base value. Also, the *Training_improvement_adjustment_time* is set to 4, meaning that the training is only conducted once a year, and the *Average_improvement_per_training* is set to 0.05, which means one training session can only improve the IS staff competence level to 0.05 higher than the current level. This scenario also needs to adjust the *Training_factor* graphical relationship. The adjusted relationship is now presented by Figure 7.25.

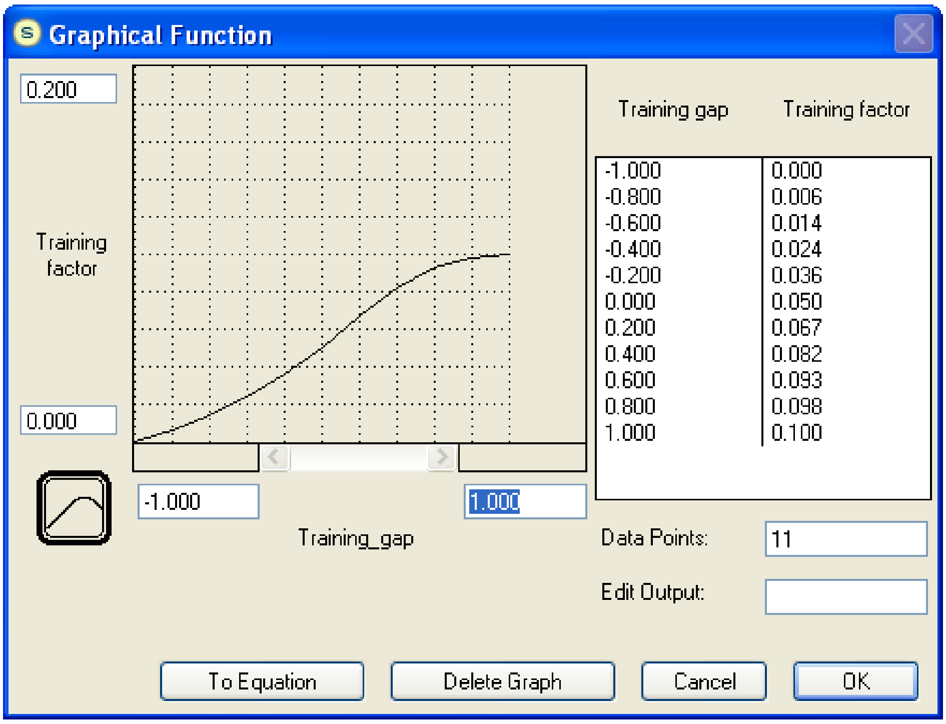


Figure 7.25 *The adjusted impact of training on the competence level*

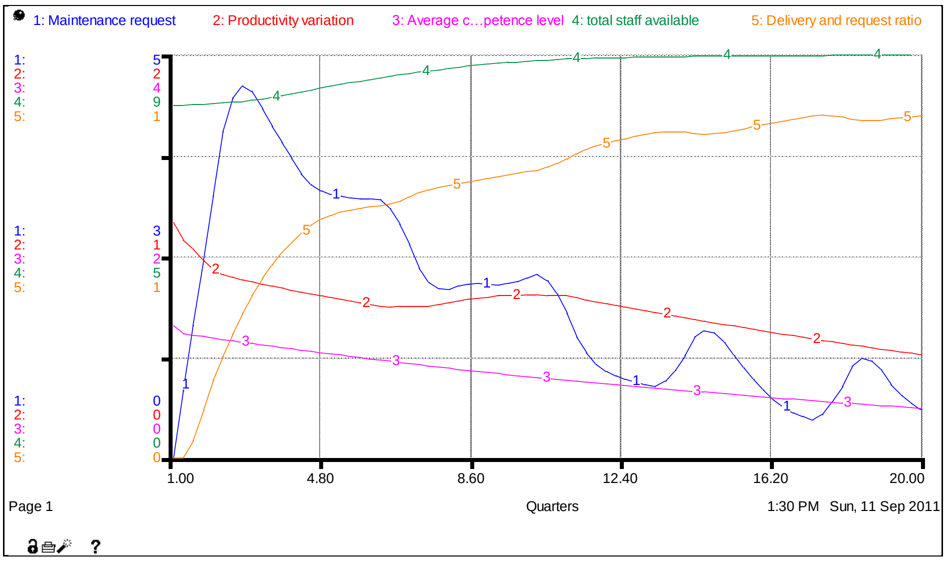


Figure 7.26 *Dynamic behaviour of the software availability level and other key factors resulting from scenario 5*

The simulation run results in Figure 7.26 describes the dynamic behaviour of the software availability and other key factors. In this figure, staff productivity is relatively low and decreasing (line 2), following the similar pattern of the staff average competence level (line 3), while the number of staff available for SM (line 4) is the same as that of the base scenario. Compared with the dynamic software availability level of the base scenario, the dynamic level resulting from this scenario shows a different pattern of increase (line 5). In general, the level is much lower than the base one, although it gradually increases. It takes a longer time to reach its highest level than in the base scenario and, in general, the levels are much less than those of the base scenario.

The average staff competence level, representing the number of maintenance tasks that can be completed and delivered per person per quarter, in addition to its relatively low level, continuously declines due to the inability of training to increase this initially low level to a higher one, as depicted in Figure 7.27. This is shown by the widening gap between the training needed (line 2) and the actual training (line 1) over time. This decreasing level is also caused by IT advancement.

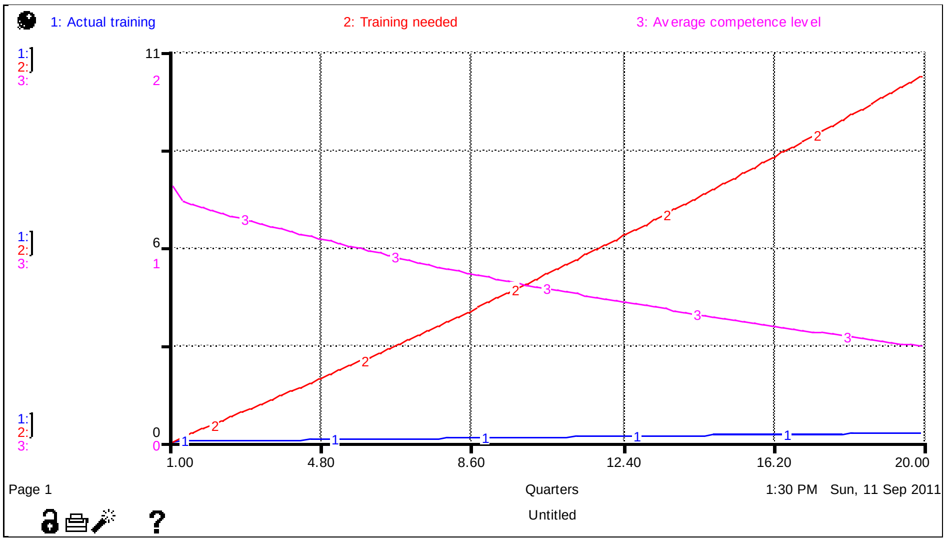


Figure 7.27 *Dynamic increase of trainings gap and its impact on the competence level*

On reviewing the simulation results, it can be concluded that, to some extent, the e-government system is still able to preserve the software availability level but with very limited service delivery. This might occur because of the characteristic of the adopted dynamic level of MRs by the model, which reflect the maintenance workload. As can be seen in Figure 7.26, the high level of MRs in early periods might make the job of the maintainers, who have low competence, is very difficult; however, as the MRs level is decreasing and the IS staff availability level is increasing, the software availability level increases. So, if the MRs level is low as a result of high-quality software produced by a software development project in a non-volatile environment, which generates fewer organisational and user enhancement requests (Conboy & Lang 2011), then high levels of software availability over time may be achievable, as indicated in scenario 2. This is, indeed, by assuming that critical MRs can always be managed.

This is in contrast with the case of the eGSMA. The relatively competent staff available for SM can make the availability reach a much higher level more quickly than in this scenario. In addition, the training attended by IS staff has been able to prevent the average competence level from a steeper descent. In some cases, as stated by the eGSMA IS staff, the training even matched up with the problems being faced by the staff, and therefore the improvement level resulting from training could enable them to reach more than 75% of their current competence level. It must be noted that during early periods of the eGSMA, the competence level of IS staff with respect to organisational needs was also low. However, as the eGSMA sent them to formal and intensive training programs relevant to organisational needs, the improvement of the competence levels was significant.

7.9 Reallocate idle staff scenario

The scenario is to construct a relationship between maintenance requests and other duty levels to describe the possibility that when the MRs go down, then other duty levels go up. As revealed from previous research and the eGSMA case described in previous chapters, the level of MRs of closed-developed software in general is decreasing over time, leading to a zero level, and the requests arrive at random frequency per quarter. A reallocation of idle IS staff scenario might occur when e-government system managers decide to allocate IS staff resources to carry out other duties based on the level of SM workload. This assignment to other duties is meant to take advantage of idle resources because of the low level of SM workload, which is a quite plausible practice.

Unlike this scenario, within the eGSMA IS staff might be assigned to non-SM tasks irrespective of the current level of overall maintenance tasks. As described in Chapter 6, the eGSMA sets up the other-duty level as constant, on average, at 30% of total workload. However, by distinguishing staff into two categories, functional and structural, the eGSMA can limit the average of the other-duty level in the sense that the functional staff are given only a small percentage of other duties.

To accommodate this scenario, the *Other_duty_level* is now influenced by the *Maintenance_request* stock. The relationship between these two factors is described by Figures 7.28 and 7.29. Figure 7.29 describes when, in a particular quarter, the MRs level is zero, then about 90% of staff availability is allocated to other duties, and if the level is 13 (or one MR per week), then only about 10% of staff availability is allocated to other duties. It is also assumed that the relationship pattern between the requests and allocation to other tasks is S-form.

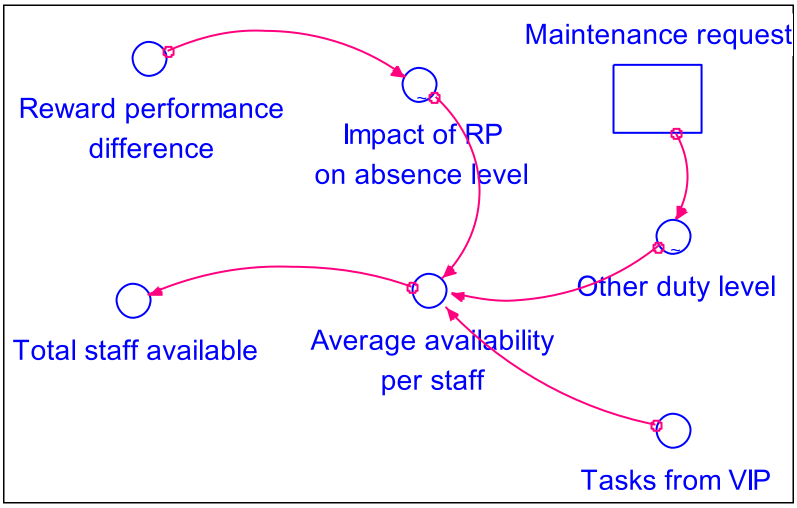


***Figure 7.28** A partial view of the model after a structural relationship between MRs and* Other_duty_level *is added*

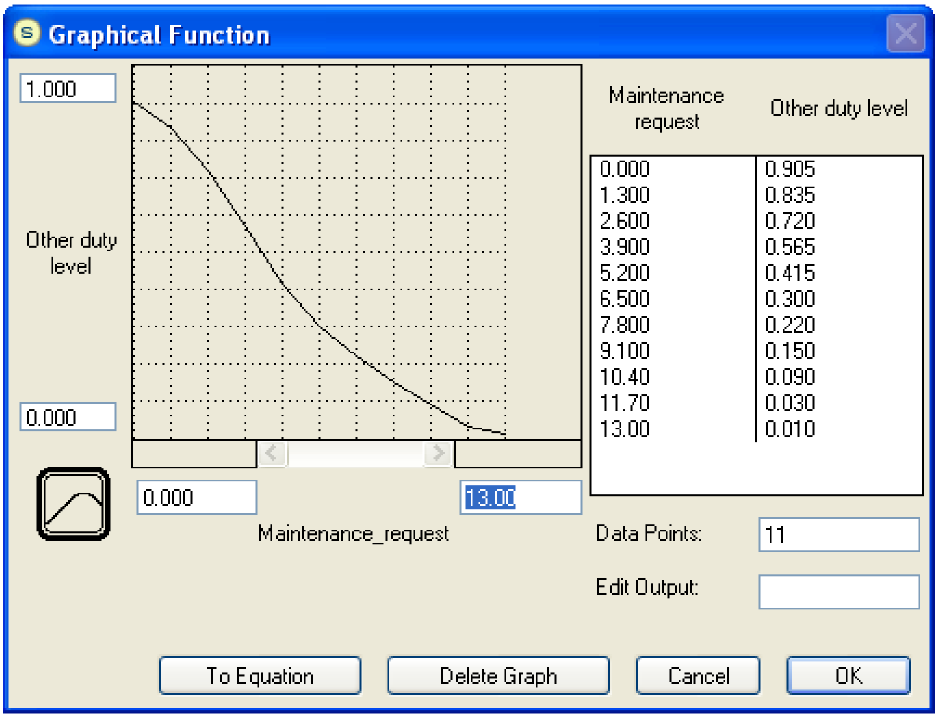


***Figure 7.29** A relationship between MRs and* Other_duty_level

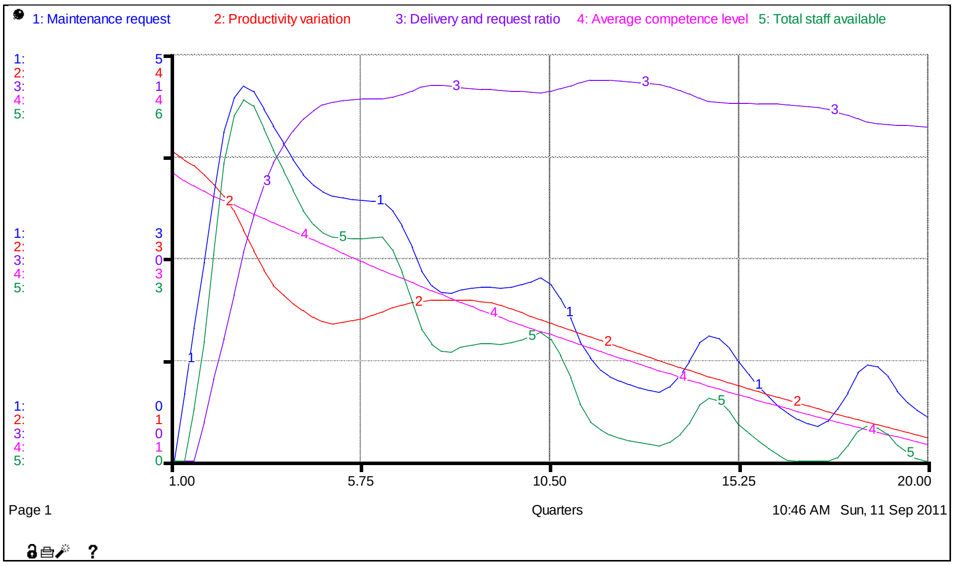


***Figure 7.30** Dynamic behaviour of the software availability level and other key factors after MRs and* Other_duty_level *are linked*

After running the simulation, this scenario shows that allocating IS staff to carry out other jobs when they are idle might not jeopardise software availability level in the short term. The delivery and request ratio in Figure 7.30 indicates that the level starts to decline slightly around the 12th quarter (line 3). However, this decline seems to continue. It suggests that allocating a proportional level of IS staff with respect to MRs to perform other tasks could lower software availability in the long term, although the level of MRs also becomes low over time. Accordingly, Figure 7.30 also suggests that it might be better to stipulate a certain minimum level of staff availability for SM, to prevent the ratio from a continuous decrease, although the SM workload continuously decreases.

The decreasing pattern of MRs over time might also suggest to managers to consider an arrangement of time and other resources in regard to the release of a new software version. This is in the sense of how to take advantage of and allocate idle resources to create and implement better software release planning. According to the second law of software evolution (Lehman & Ramil 2001), a planning to regulate the release time of a new version of software is necessary because any implemented and delivered SM will increase software complexity, which in turn makes the software more difficult to maintain, resulting in a shorter software lifetime as shown by previous scenario. From the e-government point of view, a shorter software lifetime means e-government success cannot last long. Regulating the release time by taking advantage of idle IS staff could control the increase in the software complexity, therefore ensuring a higher possibility of software sustainability.

7.10 High dependency on external expertise scenario

This scenario simulation concerns the impact of the high dependence of an e-government system on an external expertise factor which is different from the eGSMA case. A study reported by Nengomasha, McHombu and Ngulube (2010) illustrated e-government systems that depend on external consultants from overseas due to a lack of IT-capable internal human resources. This dependence has given rise to concern over, among many other aspects, e-government system maintenance and sustainability. This concern was also indicated by Heeks (2002) over externally developed e-government systems, that is, about the sustainability of the system when the key IS staff from an external organisation of the system exit. This scenario will be implemented in two different settings, because there are two possible ways for external staff to exit: suddenly or gradually.

The first setting is a scenario simulation when external staff exit suddenly. In this situation, the e-government systems may have a sufficient number of IS staff who maintain the e-government software but, as the internal IS staff are still lacking in competence, they depend highly on other personnel outside the e-government unit. These personnel might be the external software developers who developed all or a subset of the currently operating software and, based on a contract, they support internal IS staff to maintain the software for a particular period of time. Once the contract expires, the support to the internal IS staff is suddenly ceased and the internal IS staff, who are lacking in capability, have to undertake incoming maintenance tasks in full.

To accommodate this sudden exit scenario, the value of the *Communication_level_with_experienced_user,* which could reflect the dependency of internal IS staff on external personnel, is now replaced with a STEP function. It is assumed that the input is *0.05+STEP(0.9,12)*, which means the communication level with and support from the developer can be obtained relatively easily for about 12 quarters; beyond that period, the support is no longer available.

After running this scenario simulation, Figure 7.31 indicates that the software availability level behaves as in the base scenario, except after the 12th quarter it starts to monotonically decrease (line 5). As the support from the external personnel comes to an end, the productivity variation drops to zero for the rest of the time (line 2), causing the maintenance delivery to stop. So, the latest level of the software availability is maintained for the rest of the time. However, because the MRs will never be zero, the later incoming MRs cannot be carried out and delivered, then they slowly accumulate. This eventually causes the software availability level to go down.

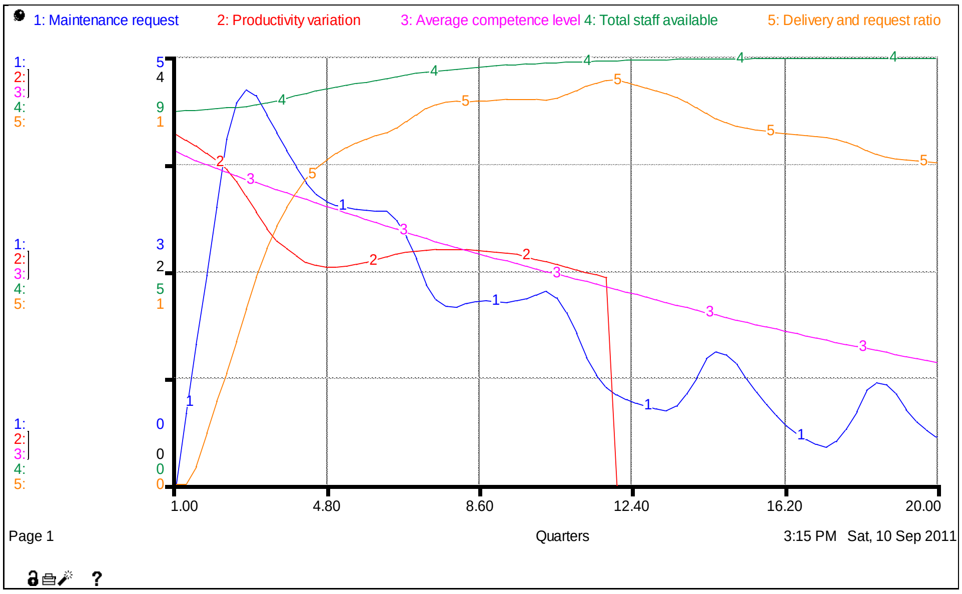


Figure 7.31 *Dynamic behaviour of the software availability level and other key factors when the support from external experienced personnel is withdrawn suddenly*

As might be expected, the simulation shows that high reliance on external human resources, while the internal IS staff are not prepared to take over because of a lack of skills, could jeopardise e-government sustainability in the long term. However, if the requirement levels for SM from the users or organisations are not high and the quality of the software is appropriate, the ceasing of support from external resources does not necessarily mean a sudden halt in e-government service delivery, but rather a persistent decrease in software availability level – and hence service delivery level – that will be experienced by the e-government system. The length of time over which the system can sustainably deliver service much depends on the criticality of the new MRs and the frequency of MRs per period of time. As mentioned earlier in

Chapter 5, the base scenario assumes that all critical MRs are given high priority and can always be completed.

Thus, it might be wise only to develop and implement an e-government system, when assisted by external resources, which addresses a relatively stable system in the sense that it does not persistently change over time and is simple (Goldfinch 2007). In unstable systems, where rules and regulations as well as organisational changes are highly dynamic, dependence on external personnel could be a serious threat to achieving e-government success.

The second setting of the scenario is that it might occur that the external resources which support SM are withdrawn gradually. This withdrawal causes the productivity variation to go down because the internal IS staff now take more time to complete a maintenance task compared to the time when the external resources were easily reachable. These are described by relationships between time and communication level shown in Figure 7.32, and between environmental factor, external resource, and the *Productivity_adjustment_time* as in Figure 7.33. The partial view of the modified model is presented in Figure 7.34.

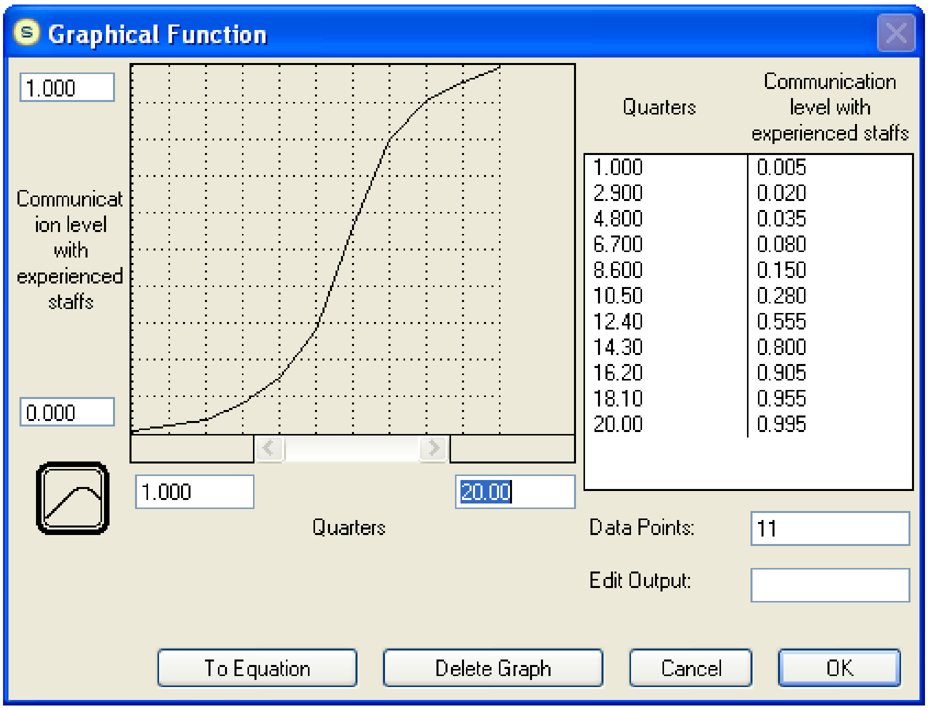


Figure 7.32 *Dynamic relationship between time and the support from external personnel*

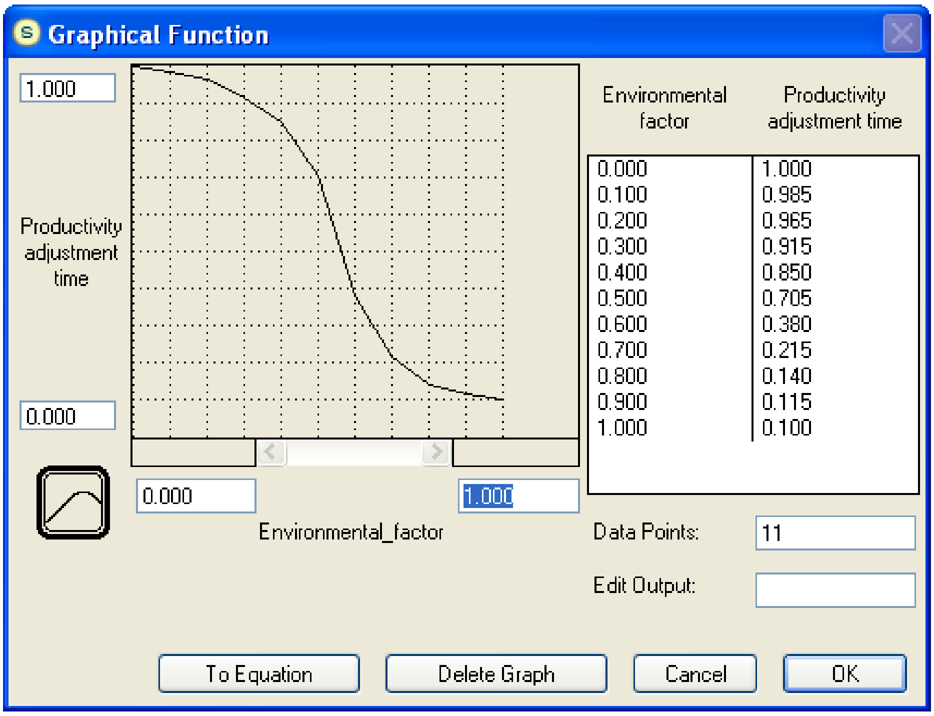


Figure 7.33 *Dynamic relationship between the level of support from external personnel and productivity adjustment time*

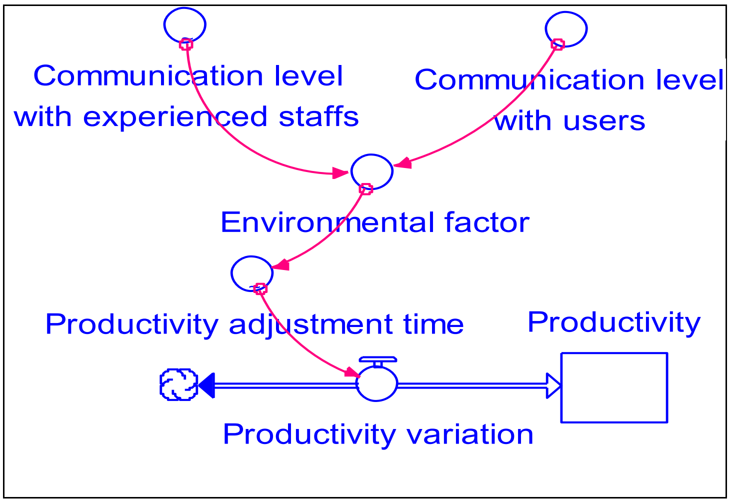


Figure 7.34 *A partial view of the model after a structural relationship between the support from external personnel and productivity adjustment time is added*

The dynamic behaviour of some key factors in the simulation of this scenario is presented in Figure 7.35.

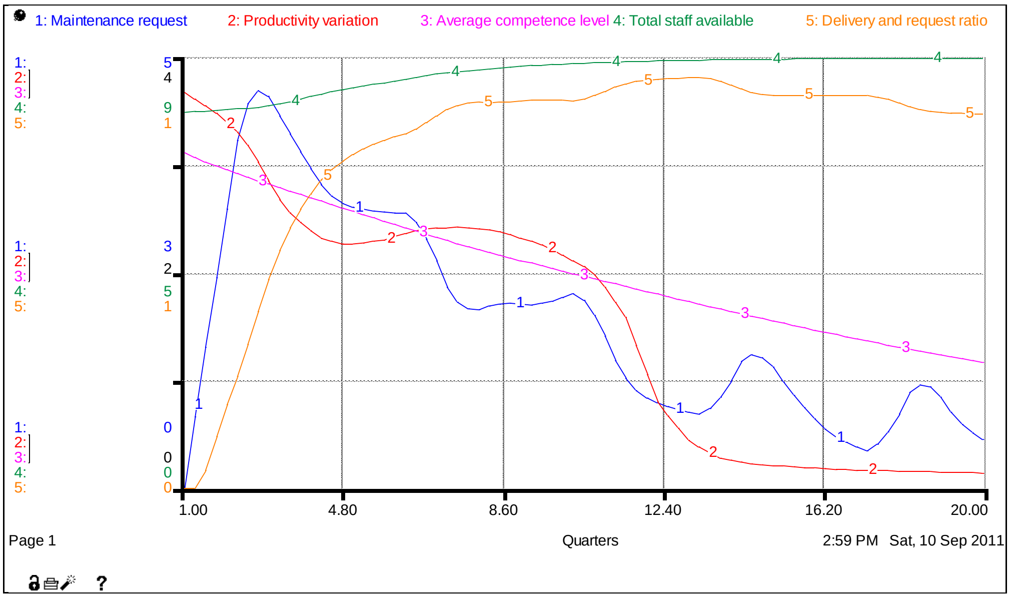


Figure 7.35 *Dynamic behaviour of the software availability level and other key factors when the support from external personnel is withdrawn gradually*

As with the previous case where the support from the external resources stops suddenly, a gradual withdrawal also causes a decrease in the software availability level, although with a slower rate of decrease (line 5 of Figure 7.35). This indicates that if the e-government system is able to prevent productivity variation from a decline by, for example, improving self-reliance on the part of internal IS staff, the decrease in software availability level could be prevented.

7.11 Summary

This chapter has presented the simulation of the validated model and of the scenarios implemented in the model. The simulations have shown the usability of the model and present some interesting results. They show how the dynamic levels of success factors influencing other success factor levels eventually lead to the success or failure of an e-government system. They also suggest how to reverse a declining e-government service delivery level into an increasing one.

The simulation of the model has shown explicitly the dynamic relationships between the success factors of SM and IS staff-management subsystems in influencing eGSMA success. The simulation outputs indicate how, from the point of view of the dynamic relationships between endogenous success factors, eGSMA success can be maintained. The eGSMA has been able to maintain a sufficient level of IS staff through retaining a low level of absenteeism which, in turn, is influenced by a level of provided rewards higher than the staff expectation. The almost-zero turnover level also causes stable full-time IS staff to be maintained. It is also able to maintain other tasks at a sufficiently low level. In addition, the simulation also shows that the eGSMA can prevent the maintenance productivity variation from a steeper decline. This is because the eGSMA management has been able to motivate IS staff to exert the necessary level of effort by means of providing higher rewards than staff expectation. The management has also been able to motivate the staff to be keen on their jobs.

The simulation of the scenarios applied to the eGSMA SD model has also revealed explicitly the roles of a number of success factors and relationships in influencing e-government success or failure. Through recurrent faults factors, a scenario simulation has been able to show the importance of impact analysis in SM; and through the dynamic maintenance requests level, another scenario simulation shows the significant role of the quality of software product resulting from a software development project in determining e-government success.

The scenario simulations also reveal that:

- Increasing IS staff turnover, because of a widening difference between provided and expected rewards over time, makes e-government failure likely to occur. However, a sufficient degree of opportunity to reverse this to e-government success is available for e-government decision-makers.
- A widening difference between provided and expected rewards over time, which increases the tendency of IS staff to moonlight and to undermine the impact analysis of SM, can cause an increase in the probability of e-government failure.
- A persistent inability to recruit competent staff or to improve less competent staff makes it difficult for an e-government system to achieve success.
- A reassignment of idle IS staff with respect to the maintenance workload level, without a minimum limit to carry out other duties, does not affect e-government success in the short term but might cause failure in the long term.
- A ceasing of full support from external expertise in maintenance does not make an e-government system fail in the short term.

In the final chapter following, the research contributions and limitations will be presented along with a discussion of possible future research.

Chapter 8 Summary and Future Research

8.1 Introduction

This chapter presents the contributions made by and limitations associated with this research, as well as possible avenues for future research. The research endeavour described in previous chapters has been able to answer the research questions and to elicit a number of important findings relevant to the research aims. The findings are believed to contribute significantly to the body of e-government knowledge and fruitfully support decision-making processes and management practices in achieving e-government success, especially in developing countries.

This chapter will first present the research contributions to the body of knowledge and practice. Second, a set of identified research limitations is described. As with any other research, this study also contains inherent limitations such as adopted assumptions and views, a research approach and methods, and the way in which a conclusion is drawn. Last, several areas for possible future research associated with this study are presented.

8.2 Research contributions

The eGSF relationships model developed in this research has been established and grounded in a number of disciplines: software engineering, human resources management and SD modelling. This modelling study contributes to the study of e-government, particularly in developing countries, by offering an innovative approach and a formal model for exploring the relationships between the e-government success factors and the critical impacts of those relationships on the success of e-government. The contributions are both to the body of knowledge and to the practices of e-government.

8.2.1 Contribution to body of knowledge

This study contributes to the improvement of understanding in studying e-government success in developing countries, from the perspective of the relationships between eGSFs. In particular the study provides a theoretical framework for sustaining e-government success during the operation and maintenance period of the e-government system life cycle. More particularly, the study advances knowledge in this field by developing a formal model of the dynamic relationships between the eGSFs derived from the SM elements and processes, and drawn from expectancy theory and other motivation-related factors in the IS human resources management field.

This research expands the e-government body of knowledge in the following ways:

- While most e-government studies explore e-government success from the point of view of e-government development or adoption, this study broadens the existing knowledge of e-government success by exploring the sustainability of e-government during the operation and maintenance stages of the e-government system life cycle.
- The existing eGSF models focus mostly on linear or one-way directional influences of the success factors on e-government success, assuming the success factors are mutually independent. However, this study offers an innovative extension to this existing knowledge in the form of a success factors model that reveals explicitly the dynamic relationships between the success factors in influencing e-government success, and allows non-linear and feedback relationships. Therefore, this extends the knowledge that it is not only the success factors that determine e-government success but also the relationships between them.
- This research extends the implementation of the dynamic model of classic expectancy theory to model the productivity, availability and motivation factors of IS staff in e-government SM associated with a case in a developing country. The model also expresses separately and explicitly the dynamic of rewards provided to the staff and the dynamic of rewards expected by the staff, based on the overall SM performance.
- This study has modelled the dynamic feedback dependency of the elements and processes of SM in an innovative way, proposing a different perspective for conceptualising software maintenance and software maintenance performance.
- The introduction of a way of incorporating and signifying dynamic feedback relationships between IS staff average competence level, competence obsolescence, training and experience has also improved insight into their roles in realising successful e-government SM.

8.2.2 Contribution to practice

The model development was grounded on the perspective of the importance of e-government SM management during the operation and maintenance stages, instead of focusing on e-government development management. E-government decision-makers or managers in developing countries could find that this study has practical significance, particularly in harnessing e-government resources and efforts to ensure a sustainable e-government system. The model is able to show its usefulness for decision-makers and managers by revealing explicitly the dynamic dependency of the success factors and demonstrating the impacts of various different policies on e-government success. Specifically, the model shows the importance of considering other factors that might be affected when a decision or policy is made on a particular factor. The model explains the chain of relationships between the success factors, which can affect the final outputs and outcomes of a decision in the long term that might not be visible in the short term.

Specific contributions of this study to the work of e-government decision-makers or managers in developing countries could include:

- From the perspective of e-government SM in developing countries, the model suggests that e-government success is much more likely if the e-government unit develops relatively small-scale but operative software of a relatively stable business process for e-government service delivery. The scale and scope of the software can be improved incrementally. By adopting this policy, the SM workload can be kept to a relatively low level appropriate to the limited resources available, and the internal software developer can be assigned as the software maintainer, which can also reduce the negative impacts of improper maintenance impact analysis.
- The model has shown that IS staff perception of being highly rewarded is an influential success factor for e-government success. The case study showed that the rewards provided can be in various forms: a secure job, additional income through additional relevant tasks, relevant training, and the real use of the software developed and maintained by the staff. Providing rewards that are perceived as having a value higher than IS staff expectation, via reductions in turnover and absence level as well as promoting better morale, can eventually sustain e-government system success.

- The model also indicates and suggests the dynamic impacts of policy on software availability over time: effective training in accordance with organisation and individual IS staff needs, maintenance-competence levels of newly recruited staff, staff allocation to other tasks and external expertise roles.

8.3 Research limitations

As with any other research, this study inherently contains various limitations. It is necessary to acknowledge and express these limitations explicitly, so that the use of this study can be put into a proper context.

One of the noticeable limitations of this study is that it concentrates mainly on the SM and IS staff management domains; it has been acknowledged and modelled that the success factors encompass a very broad range of factors as presented by the success factors system decomposition in Chapter 5. The study also views e-government sustainability from the perspective of software availability; on the other hand, the hardware and network components of e-government systems, which can certainly affect e-government sustainability and capacity to deliver services, are not considered. Moreover, the model development in this study does not take other success factors from other subsystems into account. This was intended to avoid an exceedingly complex undertaking with respect to the resources available and the usefulness of the results (Sterman 2000, Ch. 3).

This SD model of eGSF relationships has been built on a case study for a specific purpose. No claim for statistical generalisation resulting from the study will be proposed, as with any case study (Yin 2003). The formal model developed in this study has, however, provided a facility for exploring and explaining the complex relationships between the success factors (Yin 2003), hence improving the insight of both academia and practitioners in e-government into the way e-government success can be realised. Suitable parameter values from any comparable e-government systems can substitute for the current values, and the ability of the model to explain the dynamic behaviour of interest in this system extends the usefulness of the model.

The data collection process of this case study could be another possible source of limitation, although every effort has been made to ensure case study data-collection process rules have been fully adhered to. As might be observed in Chapter 6, much of the data was based on the subjective judgement of and relied on the respondents' personal memory. Also, concepts and definitions might be conceived differently from their intended meanings. These could be threats to the internal validity. However, as the interviews were conducted with a number of senior IS staff who have been dealing closely with their own SM tasks for many years, this threat could be minimised. Additionally, the interviews were also conducted and guided using structured diagrams to assist interviewees to stimulate their memories or to understand a concept properly.

The plausible assumptions taken from the judgements of the researcher and from previous studies were also introduced to the model for parameters and relationships for which the required data was not available from the case study. These assumptions could not be perfect substitutes for real-world behaviour. Therefore, behaviour sensitivity and extreme tests were applied to the model in order to evaluate and ensure that small changes applied to the adopted assumptions did not cause erratic and unacceptable behaviour of the model.

8.4 Future research

There are avenues for future research to extend and improve this study, which could contribute to the knowledge of academia and practitioners in e-government. Some of these possibilities are outlined below:

- Inclusion of other elements of information systems. While this study restricts the scope of e-government sustainability to the perspective of software availability over time, the inclusion of computer hardware and network availability over time would provide more insight into more general service availability over time, extending knowledge of the relationships between the success factors of e-government sustainability.
- Inclusion of other success factors from other eGSF subsystems. As shown in the eGSFs system decomposition in Chapter 6, e-government success can be influenced by a myriad of success factors – many of which are not a subset of the IT area, such as leadership, funding, staff who operate the e-government system for service delivery, private companies, etc. Future

research that includes a specific set of success factors within these areas could reveal the dynamic roles of these success factors in relation to e-government success.

- Linking the success factors of e-government SM and e-government software development. Some research has stressed the importance of considering system maintenance from the early stages of system development. This opens possible research avenues for studying the dynamic relationships between the success factors at the development stage and the maintenance stage in influencing e-government success.
- Evaluate the model against comparatively similar e-government systems (Morecroft 2007**:** 408; Pornphol & McGrath 2011; Sterman 2000**:** 881). This kind of e-government system is a member of a broader family of e-government systems which, to some extent, display the same characteristics or relatively the same structure. Evaluative and iterative case study research could be performed by substituting parameters specific to the new case into the model. This would recalibrate the model in a new situation. Simulations of the recalibrated model, which could show the ability of the model to reproduce the dynamic behaviour of the case, could extend the usefulness of the model and improve the general understanding embodied in the model.

References

Abdel-Hamid, T & Madnick, SE 1991, *Software project dynamics: an integrated approach*, Prentice Hall, Engelwood Cliffs, New Jersey.

Abran, A & Nguyenkim, H 1991, 'Analysis of maintenance work categories through measurement', in *Proceeding of the Conference on Software Maintenance 1991*, Sorrento , Italy

—— 1993, 'Measurement of the maintenance process from a demand-based perspective', *Software Maintenance: Research and Practice*, vol. 5, pp. 63-90.

Abu-Shanab, E, Al-Rub, SA & Md Nor, K 2010, 'Obstacles Facing the Adoption of E-Government Services in Jordan', *Journal of E-Governance*, vol. 33, no. 1, pp. 35-47.

Acuna, ST, Juristo, N & Moreno, AM 2006, 'Emphasizing human capabilities in software development', *Software, IEEE*, vol. 23, no. 2, pp. 94-101.

Ahn, Y, Suh, J, Kim, S & Kim, H 2003, 'The software maintenance project effort estimation model based on functions points', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 15, pp. 71-85.

Akkermans, HA & van Oorschot, KE 2005, 'Relevance Assumed: A Case Study of Balanced Scorecard Development Using System Dynamics', *The Journal of the Operational Research Society*, vol. 56, no. 8, pp. 931-41.

Alalwan, J & Thomas, MA 2011, 'A Holistic Framework to Evaluate E-government Systems', in *Proceeding of the AMCIS 2011*, Detroit, USA.

Alhujran, O 2009, 'Determinants of e-government services adoption in developing countries: a field survey and a case study', Doctor of Philosophy thesis, University of Wollongong.

Ali, M & Bailur, S 2007, 'The Challenge of "Sustainaibility" in ICT4D - Is *Bricolage* the Answer?', in *Proceeding of the 9th International Conference on Social Implications of Computers in Developing Countries*, Sao Paulo, Barzil.

Allen, J & van der Velden, R 2002, 'When do skills become obsolete, and when does it matter?', in A de Grip, J van Loo & K Mayhew (eds), *The Economics of Skills Obsolescence: Theoretical Innovations and Empirical Applications*, Emerald Group Publishing, vol. 21, pp. 27–50.

Almahamid, S, McAdams, AC, Kalaldeh, TA & Al-Sa'eed, M 2010, 'The relationship between perceived usefulness, perceived ease of use, perceived inofrmation quality, and intention to use e-government', *Journal of Theoretical and Applied Information Technology*, vol. 11, no. 1, pp. 30-44.

AlShihi, H 2006, 'Critical Factors in the Adoption and Diffusion of E-Government Initiatives in Oman', Ph D thesis, Victoria University.

Altameem, T, Zairi, M & Alshawi, S 2006, 'Critical Success Factors of E-Government: A Proposed Model for E-Government Implementation', in *Proceeding of the Conference in Innovations in Information Technology, 2006*, Dubai, pp. 1-5.

Andersson, C & Runeson, P 2007, 'A replicated quantitative analysis of fault distributions in complex software systems', *IEEE Transactions on Software Engineering*, vol. 33, no. 5, pp. 273-86.

Anonymous 2005, 'Building Local E-Government trough Public-Private Partnerships: Conference Conclusions', paper presented to E-Government Eastern European Conference (eGEEC 2005), Sinaia, Romania, 12-13 September 2005.

—— 2011, 'The 2011 Waseda University World e-Government Ranking', *Journal of E-Governance*, vol. 34, no. 2, pp. 56-63.

Antoniol, G, Ayari, K, Penta, MD, Khomh, F & Guéhéneuc, Y-G 2008, 'Is it a bug or an enhancement?: a text-based approach to classify change requests', in *Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: meeting of minds*, Ontario, Canada.

April, A & Abran, A 2008, *Software maintenance management: evaluation and continuous improvement*, John Wiley & Sons, Inc, Hoboken, New Jersey.

April, A, Huffman Hayes, J, Abran, A & Dumke, R 2005, 'Software Maintenance Maturity Model (SMmm): the software maintenance process model', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 17, no. 3, pp. 197-223.

Avison, D & Fitzgerald, G 2006, *Information systems development: Methodologies, techniques & tools*, 4th edn, The McGraw-Hill Companies, London.

Baker, J, Mapes, J, New, C & Szwejczewski, M 1997, 'A hierarchical model of business competence', *Integrated Manufacturing Systems*, vol. 8, no. 5, pp. 265-72.

Baldersheim, H 2004, 'In the Virtual Town Hall, New Information Regimes and A Third Face of Power, Nordic Cities on the Web', in *City Futures. An International Conference on Globalism and Urban Change*, University of Illinois at Chicago.

Banker, RD, Datar, SM & Kemerer, CF 1987, 'Factors affecting software maintenance productivity: an exploratory study', in *The 8th International Conference on Information Systems*, Pittsburgh, PA, pp. 160-75.

Banker, RD, Datar, SM & Kemerer, CF 1991, 'A model to evaluate variables impacting the productivity of software maintenance projects', *MANAGEMENT SCIENCE*, vol. 37, no. 1, pp. 1-18.

Banker, RD, Datar, SM, Kemerer, CF & Zweig, D 1993, 'Software complexity and maintenance costs', *Communication of the ACM*, vol. 36, no. 11, pp. 81-94.

Banker, RD, Datar, SM, Kemerer, CF & Zweig, D 2002, 'Software errors and software maintenance management', *Information Technology and Management*, vol. 3, no. 1, pp. 25-41.

Banker, RD, Davis, GB & Slaughter, SA 1998, 'Software developement practices, software complexity, and software maintenance: A field study', *MANAGEMENT SCIENCE*, vol. 44, no. 4, pp. 433-50.

Baqir, MN & Iyer, L 2010, 'E-government Maturity over 10 Years: A Comparative Analysis of E-government Maturity in Select Countries Around the World', in CG Reddick (ed.), *Comparative E-Government*, Springer New York, vol. 25, pp. 3-22.

Barlas, Y 1996, 'Formal aspects of model validity and validation in system dynamics', *System Dynamics Review (Wiley)*, vol. 12, no. 3, pp. 183-210.

Basu, S 2004, 'E-Government and Developing Countries: An Overview', *International Review of Law, Computers & Technology*, vol. 18, no. 1, p. 24.

Becker, J, Niehaves, B, Algermissen, L, Delfmann, P & Falk, T 2004, 'e-Government Success Factors', in R Traunmüller (ed.), *Electronic Government*, Springer Berlin / Heidelberg, vol. 3183, pp. 503-6.

Belanger, F & Carter, L 2009, 'The impact of the digital divide on e-government use', *Communication of the ACM*, vol. 52, no. 4, pp. 132-5.

Belanger, F & Hiller, JS 2006, 'A framework for e-government: privacy implications', *Business Process Management Journal*, vol. 12, no. 1, pp. 48-60.

Bendifallah, S 1987, 'Understanding software maintenance work', *IEEE Transactions on Software Engineering*, vol. 13, no. 3, pp. 311-23.

Benestad, HC, Anda, B & Arisholm, E 2010, 'A Method to Measure Productivity Trends during Software Evolution', in LA Maciaszek, C González-Pérez & S Jablonski (eds), *Evaluation of Novel Approaches to Software Engineering*, Springer Berlin/Heidelberg, vol. 69, pp. 147-62.

Berman, AM, Bowman, JS, West, JP & Wart, MV 2006, *Human resouce management in public service: Paradoxes, processes, and problems*, Sage Publication, Inc., Thousand Oaks.

Best, ML & Kumar, R 2008, 'Sustainability failures of rural telecenters: Challenges from the sustainable access in rural india (sari) project', *Information Technology and International Development*, vol. 4, no. 4, pp. 31-45.

Best, ML, Thakur, D & Kolko, BE 2009, 'The contribution of user-based subsidies to the impact and sustainability of telecenters - the eCenter project in Kyrgyzstan', in *Proceeding of the 2009 International Conference on Information and Communication Technologies and Development (ICTD)*, Doha, pp. 192-200.

Beynon-Davies, P 2007, 'Models for e-government', *Transforming Government: People, Process and Policy*, vol. 1, no. 1, p. 22.

Bhatnagar, S 2002, 'Egovernment: Lessons from Implementation in Developing Countries', *Regional Development Dialogue*, vol. 24, no. Autum, p. 9.

—— 2004, *E-Government: from vision to implementation: a practical guide with case studies*, Sage Publications India Pvt Ltd, New Delhi 110 017.

Bhatt, P, Shroff, G, Anantaram, C & Misra, AK 2006, 'An influence model for factors in outsourced software maintenance', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 18, pp. 385-423.

Bhuiyan, SH 2010, 'E-Government in Kazakhstan: Challenges and Its Role to Development', *Public Organization Review*, vol. 10, no. 1, pp. 31-47.

Black, S 2001, 'Computing ripple effect for software maintenance', *Journal of Software Maintenance & Evolution: Research & Practice*, vol. 13, no. 4, pp. 263-79.

—— 2006, 'The Role of Ripple Effect in Software Evolution', in Nazim H. Madhavji, Juan C. Fernández-Ramil & DE Perry (eds), *Software Evolution and Feedback: Theory and Practice*, John Wiley & Sons, Ltd, Chichester, UK, pp. 249-68.

Blanton, JE, Wingreen, SC & Schambach, T 2000, 'Relationships between personal demographics and motivation of information technology professionals to participate in professional development research-in-progress', paper presented to The 2000 ACM SIGCPR Conference on Computer Personnel Research, Chicago, Illinois, United States, 06 - 08 April 2000.

Blum, BI 1995, 'Resolving the Software Maintenance Paradox', *Journal of Software Maintenance: Research & Practice*, vol. 7, no. 1, pp. 3-26.

Boehm, BW 1981, *Software Engineering Economics*, Prentice-Hall, Inc, Englewood Cliffs, New Jersey 07632.

Boehm, BW & Lane, J 2007, 'Using the incremental commitment model to integrate system acquisition, systems engineering, and software engineering', *CrossTalk*, vol. 19, no. 10, pp. 4-9.

Bostrom, RP & Heinen, JS 1977, 'MIS Problems and Failures: A socio-Technical Perspectives', *MIS Quarterly*, vol. 1, no. 3, p. 16.

Buckley, J, Mens, T, Zenger, M, Rashid, A & Kniesel, G 2005, 'Towards a taxonomy of software change', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 17, no. 5, pp. 309-32.

Burch, E & Kung, H-J 1997, 'Modeling software maintenance request: a case study', in *Proceeding of the 13th International Conference on Software Maintenance (ICSM'97)*, Bari, Italy.

Burchill, G & Fine, CH 1997, 'Time versus Market Orientation in Product Concept Development: Empirically- Based Theory Generation', *MANAGEMENT SCIENCE*, vol. 43, no. 4, pp. 465-78.

Burton-Jones, A & Meso, PN 2006, 'Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis', *Information Systems Research*, vol. 17, no. 1, pp. 38-60.

Calow, H 1991, 'Maintenance productivity factors-a case study', in *Proceedings of the Conference on Software Maintenance 1991* Sorrento , Italy pp. 250-3.

Cappelli, P 2001, 'Why is it so hard to find information technology workers?', *Organizational Dynamics*, vol. 30, no. 2, pp. 87-99.

Carter, L & Bélanger, F 2005, 'The utilization of e-government services: citizen trust, innovation and acceptance factors', *Information Systems Journal*, vol. 15, no. 1, pp. 5-25.

Chapin, N 2009, 'Software maintenance in complying with IT governance: A report from the field', in *Proceeding of the IEEE International Conference on Software Maintenance 2009 (ICSM 2009)*, Edmonton, AB, Canada pp. 499-502.

Chaudhury, N, Hammer, J, Kremer, M, Muralidharan, K & Rogers, FH 2006, 'Missing in Action: Teacher and Health Worker Absence in Developing Countries', *The Journal of Economic Perspectives*, vol. 20, no. 1, pp. 91-116.

Chen, CC, Ford, CM & Farris, GF 1999, 'Rewards [personnel motivation]', *Potentials, IEEE*, vol. 18, no. 2, pp. 10-2.

Cheng, J, Huang, X & Peng, K 2010, 'Exploring e-government in Beijing: from technical infrastructure to full workflow management', in *Proceedings of the 4th International Conference on Theory and Practice of Electronic Governance*, Beijing, China, pp. 167-72.

Chernichovsky, D & Bayulken, C 1995, 'A pay-for-performance system for civil service doctors: The Indonesian experiment', *Social Science & Medicine*, vol. 41, no. 2, pp. 155-61.

Chutimaskul, W, Funilkul, S & Chongsuphajaisiddhi, V 2008, 'The Quality Framework of e-Government Development', in *Proceeding of the 2nd International Conference on Theory and Practice of Electronic Governance* Cairo, Egypt, pp. 105-9.

Collofello, J & Orn, M 1989, 'Improving software maintenance skills in an industrial environment', in N Gibbs (ed.), *Software Engineering Education*, Springer Berlin / Heidelberg, vol. 376, pp. 26-36.

Conboy, K & Lang, M 2011, 'The Paradox of “Structured” Methods for Software Requirements Management: A Case Study of an e-Government Development Project Information Systems Development', in WWW Song, S Xu, C Wan, Y Zhong, W Wojtkowski, G Wojtkowski & H Linger (eds), *Information Systems Development*, Springer New York, pp. 223-32.

Coombs, CR 2009, 'Improving retention strategies for IT professionals working in the public sector', *Information & Management*, vol. 46, no. 4, pp. 233-40.

Couger, JD & Zawacki, RA 1980, *Motivating and Managing Computer Personnel*, John Wiley & Sons, New York.

Coursey, D & Norris, DF 2008, 'Models of E-Government: Are They Correct? An Empirical Assessment', *Public Administration Review*, vol. 68, no. 14, pp. 523-36.

Courtois, P-J 1985, 'On Time and Space Decomposition of Complex Structures', *Communication of the ACM*, vol. 28, no. 6, pp. 590-603.

Coyle, G 2000, 'Qualitative and quantitative modelling in system dynamics: some research questions', *System Dynamics Review*, vol. 16, no. 3, pp. 225-44.

Coyle, RG 1983, 'The technical elements of the system dynamics approach', *European Journal of Operational Research*, vol. 14, no. 4, pp. 359-70.

Coyle, RG 1996, *System Dynamics Modelling: A practical approach*, Chapman & Hall, London.

Curtis, B, Kellner, ML & Over, J 1992, 'Process Modeling', *Communication of the ACM*, vol. 35, no. 9.

Dabalen, A & Wane, W 2008, 'Informal Payments and Moonlighting in Tajikistan's Health Sector', in World Bank Policy Research Working Paper No. 4555 SSRN.

Dada, D 2006, 'The Failure of e-Government in Developing Countries: A Literature Review', *The Electronic Journal of Information Systems in Developing Countries*, vol. 26, no. 7, p. 10.

Davis, GB 2005, *Management Information System*, 1-55786-948-0.

Davison, RM, Harris, RW, Qureshi, S, Vogel, DR & Vreede, G-Jd (eds) 2005, *Information Systems in Developing Countries: Theory and Practice*, City University of Hong Kong Press, Hong Kong.

Dawes, SS 2002, 'The Future of E-Government', Center for Technology in Government, University at Albany/SUNY.

de Boer, FS, Bonsangue, MM, Groenewegen, LPJ, Stam, AW, Stevens, S & van der Torre, L 2005, 'Change impact analysis of enterprise architectures', in *Proceeding of the IEEE International Conference on Information Reuse and Integration 2005 (IRI -2005)* pp. 177-81.

de Kool, D 2010, 'Learning with GIS-Based Monitors in the Netherlands', in CG Reddick (ed.), *Comparative E-Government*, Springer, New York, pp. 467-82.

De', R & Sarkar, S 2010, 'Rituals in E-Government Implementation: An Analysis of Failure', in M Wimmer, J-L Chappelet, M Janssen & H Scholl (eds), *Electronic Government*, Springer Berlin / Heidelberg, vol. 6228, pp. 226-37.

DeLone, WH & McLean, ER 2003, 'The DeLone and McLean Model of Information System Success: A Ten-Year Update', *Journal of Management Information Systems*, vol. 19, no. 4, pp. 9-30.

Detlor, B, Hupfer, ME & Ruhi, U 2010, 'Internal factors affecting the adoption and use of government websites', *Electronic Government, an International Journal*, vol. 7, no. 2, pp. 120-36.

Dictionary.com LLC 2011, *Free Online Dictionary for English Definitions*, IAC Corporation, viewed 30 March 2011, <http://dictionary.reference.com/>.

Druke, H (ed.) 2005, *Local Electronic Government: A Comparative Study*, Routledge Taylor & Francis Group, London and New York.

Ebrahim, Z & Irani, Z 2005, 'E-government adoption: architecture and barriers', *Business Process Management Journal*, vol. 11, no. 5, pp. 589-611.

Edwards, C 1984, 'Information systems maintenance: An integrated perspective', *MIS Quarterly*, vol. 8, no. 4, pp. 237-56.

Elahi, G, Yu, E & Annosi, MC 2009, 'Modeling Knowledge Transfer in a Software Maintenance Organization – An Experience Report and Critical Analysis', in J Stirna & A Persson (eds), *The Practice of Enterprise Modeling*, Springer Berlin Heidelberg, vol. 15, pp. 15-29.

Elliman, T, Irani, Z & Jackson, P 2007, 'Establishing a Framework for eGovernment Research: Project VIEGO', *Transforming Government: People, Process and Policy*, vol. I, no. 4.

Fairchild, AM, Ribbers, PMA & Nooteboom, AO 2004, 'A Success Factor Model for Electronic Markets: Defining Outcomes Based on Stakeholder Context and Business Process', *Business Process Management Journal*, vol. 10, no. 1, p. 17.

Fang, Z 2002, 'E-Government in Digital Era: Concep, Practice, and Development', *International Journal of The Computer, The Internet and Management*, vol. 10, no. 2, pp. 1-22.

Forrester, JW 1961, *Industrial Dynamics*, The MIT Press, Cambridge, MA.

Forrester, JW & Senge, PM 1980, 'Tests for building confidence in system dynamics models', *TIMS Studies in the Management Sciences*, vol. 14, pp. 209-28.

Fryling, M 2010, 'Total Cost of Ownership, System Acceptance and Perceived Success of Enterprise Resource Planning Software: Simulating a Dynamic Feedback Perspective of ERP in the Higher Education Environment ', Doctor of Philosophy thesis, The University at Albany, State University of New York.

Furuholt, B & Wahid, F 2008, 'E-government Challenges and the Role of Political Leadership in Indonesia: the Case of Sragen', in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)* Waikoloa, Hawaii.

Gaj, F & Germani, G 2008, 'Dealing with Availability in an international Service Management scenario', in A Mazzeo, R Bellini & G Motta (eds), *E-Government ICT Professionalism and Competences Service Science*, Springer Boston, vol. 280, pp. 235-44.

Gauld, R 2007, 'Public sector information system project failures: Lessons from a New Zealand hospital organization', *Government Information Quarterly*, vol. 24, pp. 102 -14.

Gauld, R & Goldfinch, S 2006, *Dangerous enthusiasms: e-government, computer failure and information*, Otago University Press, Dunedin, New Zealand.

Georgantzas, NC 2003, 'Tourism dynamics: Cyprus' hotel value chain and profitability', *System Dynamics Review (Wiley)*, vol. 19, no. 3, pp. 175-212.

Gibson, JP & McGaley, M 2008, 'Verification and maintenance of e-voting systems and standards', in D Remenyi (ed.), *Proceedings of the 8th European Conference on e-Government 2008*, Lausanne, Switzerland, pp. 283–90.

Gichoya, D 2005, 'Factors Affecting the Successful Implementation of ICT Projects in Government', *The electronic journal of e-government*, vol. 3, no. 4, pp. 175-84.

Gil-Garcia, JR 2005, 'Exploring the success factors of state website functionality: an empirical investigation', paper presented to Proceedings of the 2005 National Conference on Digital Government Research, 15-18 May 2005, Atlanta, Georgia.

—— 2006, 'Enacting State Website: A mixed method study exploring e-government success in multi-organizational setting', in *Proceeding of the 39th Hawaii International Conference on System Sciences (HICSS'06)*, Kauai, Hawaii

Gil-Garcia, JR & Luna-Reyes, LF 2006, 'Integrating conceptual approaches to e-government', in M Khosrow-Pour (ed.), *Encyclopedia of e-commerce, e-*

government, and mobile commerce, Idea Group Reference, Hershey, vol. I.

Gil-Garcia, JR & Pardo, TA 2005, 'E-government success factors: Mapping practical tools to theoretical foundations', *Government Information Quarterly*, vol. 22, p. 29.

—— 2006, 'Multi-method approaches to understanding the complexity of e-government', *International Journal of Computers, Systems and Signals*, vol. 7, no. 2, p. 3.

Goldfinch, S 2007, 'Pessimism, computer failure, and information systems development in the public sector', *Public Administration Review*, vol. 67, no. 5, pp. 917-29.

Gonzalez, P, Adenso-Diaz, B & Gemoets, LA 2010, 'A Cross-National Comparison E-government Success Measures: A Theory-Based Empirical Research', in *Proceeding of the AMCIS 2010* Lima, Peru, p. 354.

Gonzalez, R, Gasco, J & Lopis, J 2007, 'E-government success: some principles from a Spanish case study', *Industrial Management and Data Systems*, vol. 107, pp. 845-61.

Grabow, B, Druke, H & Siegfried, C 2004, 'Factors for success for local community e-government', *German Journal of Urban Studies (DfK)*, vol. 45, no. 2, pp. 1-22.

Griep, RH, Rotenberg, L, Chor, D, Toivanen, S & Landsbergis, P 2010, 'Beyond simple approaches to studying the association between work characteristics and absenteeism: Combining the DCS and ERI models', *Work and Stress*, vol. 24, no. 2, pp. 179-95.

Gronlund, A 2005, 'What's In a Field - Exploring the eGovernment Domain', paper presented to Proceeding of the 38th Hawaii International Conference on System Sciences (HICSS '05), Hawaii, 03-06 January 2005.

Grubb, P & Takang, AA 2003, *Software maintenance: concepts and Practice*, 2nd edn, World Scientific, New Jersey.

Gupta, B, Dasgupta, S & Gupta, A 2008, 'Adoption of ICT in a government organisation in a developing country: An empirical study', *Journal of Strategic Information Systems*, vol. 17, pp. 140-54.

Hafeez, K & Abdelmeguid, H 2003, 'Dynamics of human resource and knowledge management', *Journal of the operational research society*, vol. 54, no. 2, pp. 153-64.

Hair, JF, Black, WC, Barry Babin, J, Anderson, RE & Tatham, RL 2006, *Multivariate Data Analysis*, Pearson Educ. Inc. , Upper Saddle River, N.J.

Hall, T, Baddoo, N, Beecham, S, Robinson, H & Sharp, H 2009, 'A systematic review of theory use in studies investigating the motivations of software engineers', *ACM Transaction on Software Engineering Methodology*, vol. 18, no. 3, pp. 1-29.

Hall, T, Beecham, S, Verner, J & Wilson, D 2008, 'The impact of staff turnover on software projects: the importance of understanding what makes software practitioners tick', paper presented to Proceedings of the 2008 ACM SIGMIS CPR Computer Personnel Doctoral Consortium and Research Conference, Charlottesville, VA, USA, 3 - 5 April 2008.

Heeks, R 2002, 'Information Systems and Developing Countries: Failure, Success, and Local Improvisations', *The Information Society*, vol. 18, no. 2, pp. 101-12.

—— 2003, *Most e-Government-for-Development Projects Fail: How can Risks be Reduced?*, IDPM.

—— 2005a, 'e-Government as a Carrier of Context', *Journal of Public Policy*, vol. 25, no. 1, p. 24.

—— 2005b, *Implementing and Managing eGovernment: An International Text* SAGE Publications.

—— 2008a, *Causes of eGovernment Success and Failure: Design-Reality Gap Model*, eGovernment for Development, viewed 18 August 2008, <www.egov4dev.org/success/evaluation/index.shtml>.

—— 2008b, *A Factor Model for eGovernment Success and Failure*, eGovernment for Development, viewed 1 April 2009, <www.egov4dev.org/success/evaluation/factormodel.shtml>.

—— 2008c, *Success and Failure in eGovernment Projects*, eGovernment for Development, viewed 12 August 2008, <www.egov4dev.org/success/index.shtml>.

—— 2008d, *Success and Failure Rates of eGovernment in Developing/Transitional Countries: Overiew*, eGovernment for Development, viewed 12 August 2008, <www.egov4dev.org/success/sfrates.shtml>.

—— 2008e, *What is eGovernment?*, eGoverment for Development, viewed 12 August 2008, <http://www.egov4dev.org/success/definitions.shtml>.

Heeks, R & Bhatnagar, S 2001, 'Understanding success and failure in information age reform', in R Heeks (ed.), *Reinventing Government in the Information Age: International Practice IT-enabled Public Sector Reform*, Routledge, London.

Heeks, R & Davies, A 2001, 'Different approaches to information age reform', in R Heeks (ed.), *Reinventing government in the information age: international practice in IT-enabled public sector reform*, Routledge, London.

Herrera, L & Gil-Garcia, JR 2011, 'Implementation of E-Government in Mexico: The Case of Infonavit Practical Studies in E-Government', in S Assar, I Boughzala & I Boydens (eds), *Practical Studies in E-Government*, Springer New York, pp. 29-47.

Hesson, M 2007, 'Business process reeingineering in UEA public sector: a naturalization and residency case study', *Business Process Management Journal*, vol. 13, no. 5, p. 21.

Hirschfeld, R, Schmitt, L & Bedeian, A 2002, 'Job-content Perceptions, Performance–reward Expectancies, and Absenteeism among Low-wage Public-sector Clerical Employees', *Journal of Business and Psychology*, vol. 16, no. 4, pp. 553-64.

Hjorth, P & Bagheri, A 2006, 'Navigating towards sustainable development: A system dynamics approach', *Futures*, vol. 38, no. 1, pp. 74-92.

Ho, ATK & Ni, AY 2004, 'Explaining the Adoption of E-Government Features', *The American Review of Public Administration*, vol. 34, no. 2, p. 164.

Hoffer, JA, George, JF & Valacich, JS 2005, *Modern system analysis and design*, 4th edn, Pearson Education, Inc., Upper Sadle River, New Jersey 07458.

Horiuchi, C 2006, 'Creating information systems quality in government setting', in EW Duggan & H Reichgelt (eds), *Measuring information systems delivery quality*, Idea Group Publishing, Hershey, pp. 311-27.

Hosman, L & Fife, E 2008, 'Improving the prospects for sustainable ICT projects in the developing world', *International Journal of Media and Cultural Politics*, vol. 4, no. 1, pp. 51-69.

Houston, DJ 2000, 'Public-Service Motivation: A Multivariate Test', *Journal of Public Administration Research and Theory*, vol. 10, no. 4, pp. 713-28.

—— 2011, 'Implications of Occupational Locus and Focus for Public Service Motivation: Attitudes Toward Work Motives across Nations', *Public Administration Review*, vol. 71, no. 5, pp. 761-71.

Hovy, E 2008, 'An outline for the foundations of digital government research', in H Chen, L Brandt & V Gregg (eds), *Digital government: E-government research, case studies, and implementation*, Springer Science+Business Meida, New York.

Howard, KW 1989, 'A comprehensive expectancy motivation model: Implications for adult education and training', *Adult Education Quarterly*, vol. 39, no. 4, p. 199.

Hu, G, Pan, W, Lu, M & Wang, J 2009, 'The widely shared definition of e-Government: An exploratory study', *Electronic Library, The*, vol. 27 no. 6, pp. 968 - 85.

Hu, Y, Xiao, J, Pang, J & Xie, K 2005, 'A Research on the Appraisal Framework of e-Government Project Success', in *Proceeding of the 7th International Conference on Electronic Commerce - ICEC 2005* Xi'an, China.

Huff, SL 1990, 'Information system maintenance', *Business Quarterly*, vol. 55, no. 2.

Hughes, OE 2003, *Public management and administration: an introduction*, 3rd edn, Palgrave Macmillan, Hampshire RC21 6XS.

Hung, SY, Chang, CM & Yu, TJ 2006, 'Determinants of user acceptance of the e-Government services: The case of online tax filing and payment system', *Government Information Quarterly*, vol. 23, no. 1, pp. 97-122.

Hunt, P 2001, 'True stories: telecentres in Latin America and the Caribbean', *The Electronic Journal of Information Systems in Developing Countries*, vol. 4, no. 5, pp. 1-17.

Hussein, R, Karim, NSA, Mohamed, N & Ahlan, AR 2007, 'The influence of organizational factors on information system success in e-government agencies in Malaysia', *The Electronic Journal on Information Systems in Developing Countries*, vol. 29, no. 1, p. 17.

Hussein, R, Karim, NSA & Selamat, MH 2007, 'The impat of technological factors on information systems success in the electronic-government context', *Business Process Management Journal*, vol. 13, no. 5, p. 15.

Hyder, A & Ahmed, AM 2011, 'The Dynamics of Moonlighting in Pakistan', *SSRN eLibrary*, viewed 29th October 2011, <SSRN: http://ssrn.com/abstract=1875061>.

IEEE Computer Society 1993, *IEEE standard for software maintenance*, Institute of Electrical and Electronic Engineers, Inc, New York, 3 December 1992.

—— 1998, *IEEE Standard for Software Maintenance*, the Institute of Electrical and Electronics Engineers, Inc.

Igbaria, M & Greenhaus, JH 1992, 'Determinants of MIS employees' turnover intentions: A structural equation model', *Communication of the ACM*, vol. 35, no. 2, pp. 35-49.

Igbaria, M & Shayo, C (eds) 2004, *Strategies for managing IS/IT personnel*, Idea Group Pub., Hershey PA.

International Organization for Standardization 2006, *International Standard ISO/IEC 14764*, ISO/IEEE, 01-09-2006, <http://webstore.iec.ch/preview/info_isoiec14764%7Bed2.0%7Den.pdf>.

International Telecommunication Union 2009, *ICT Statistic Database, (4). Internet indicators: subscribers, users and broadband subscribers*, International Telecommunication Union, viewed 02 September 2009, <http://www.itu.int/ITU-

D/icteye/Reporting/ShowReportFrame.aspx?ReportName=/WTI/InformationTechnologyPublic&RP_intYear=2008&RP_intLanguageID=1>.

Isaac, RG, Zerbe, WJ & Pitt, DC 2001, 'Leadership And Motivation: The Effective Application Of Expectancy Theory', *Journal of Managerial Issues*, vol. 13, no. 2, p. 212.

Islam, MS & Alawadhi, S 2008, 'Sustainable Rural IS Implementation Process for Developing Countries: Agriculture Market Information Services (AMIS) in Bangladesh', in *M4D 2008*, Karlstad University, Sweden.

Jackson, MC 2003, *Systems Thinking: Creative Holism for Managers*, John Wiley & Sons, Ltd, West Sussex PO19 8SQ, England.

Jansen, AJ 2011, 'E-Government - Just a Matter of Technology?', in *Proceeding of the 44th Hawaii International Conference on System Sciences (HICSS), 2011, 4-7 January 2011* Kauai, Hawaii, pp. 1-9.

Jeet, K, Mago, VK, Prasad, B & Minhas, RS 2010, 'An Influence Diagram-Based Approach for Estimating Staff Training in Software Industry', *Arxiv preprint arXiv:1008.4268*.

Johnson, NJ 1999, 'Recruiting and retaining information systems staff for information age reform', in R Heeks (ed.), *Reinventing government in the information age: international practice in IT-enabled public sector reform*, Routledge, London and New York, pp. 350-62.

Jorgensen, M 1995, 'An empirical study of software maintenance tasks', *Software Maintenance: Research and Practice*, vol. 7, pp. 27-48.

Jorgensen, M & Sjoberg, DIK 2002, 'Impact of experience on maintenance skills', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 14, pp. 123-46.

Joseph, BK 2010, 'E-government Adoption Landscape Zambia: Context, Issues, and Challenges', in CG Reddick (ed.), *Comparative E-Government*, Springer New York, vol. 25, pp. 241-58.

Joseph, D & Ang, S 2010, 'Threat of Professional Obsolescence and Mobility Intentions: The Mediating Role of Coping Strategies', in *Proceeding of the AMCIS 2010* Lima, Peru, p. Paper 527.

Joseph, D, Kuan-Koh, CS & Hao-Foo, AC 2010, 'Sustainable IT-specific human capital: coping with the threat of professional obsolescence', in *Proceeding of the ICIS 2010*, St. Louis, MO, USA

Junchao, X, Qing, W, Mingshu, L, Ye, Y, Fan, Z & Lizi, X 2008, 'A constraint-driven human resource scheduling method in software development and maintenance process', in *Proceeding of the IEEE International Conference on Software Maintenance, 2008 (ICSM 2008)*, Beijing, pp. 17-26.

Junor, A, O'Brien, J & O'Donnell, M 2009, 'Welfare wars: public service frontline absenteeism as collective resistance', *Qualitative Research in Accounting & Management*, vol. 6, no. 1/2, pp. 26 - 40.

Kahraman, C, Demirel, NC & Demirel, T 2007, 'Prioritization of e-Government strategies using a SWOT-AHP analysis: the case of Turkey', *European Journal of Information Systems*, vol. 16, p. 15.

Kajko-Mattsson, M, Forssander, S & Olsson, U 2001, 'Corrective maintenance maturity model (CM^3): maintainer's education and training', in *Proceedings of the 23rd International Conference on Software Engineering* Toronto, Ontario, Canada, pp. 610-9.

Kamal, MM 2006, 'IT innovation adoption in the government sector: identifying the critical success factors', *Journal of Enterprise Information Management*, vol. 19, no. 2, pp. 192-222.

Kaylor, C, Deshazo, R & Eck, DV 2001, 'Gauging e-Government: A Report on Implementing Service Among American Cities', *Government Information Quarterly*, vol. 18, p. 15.

Kemerer, CF & Slaughter, SA 1997, 'Determinants of software maintenance profiles: an empirical investigation', *Software Maintenance: Research and Practice*, vol. 9, pp. 235-51.

Ketler, K & Turban, E 1992, 'Productivity improvements in software maintenance', *International journal of information management*, vol. 12, no. 1, pp. 70-82.

Khoo, HM, Robey, D & Rao, SV 2011, 'An exploratory study of the impacts of upgrading packaged software: a stakeholder perspective', *Journal of Information Technology*, vol. 26, pp. 153-69.

Kim, S 2005, 'Factors Affecting State Government Information Technology Employee Turnover Intentions', *The American Review of Public Administration*, vol. 35, no. 2, pp. 137-56.

Kim, S & Wright, BE 2007, 'IT Employee Work Exhaustion', *Review of Public Personnel Administration*, vol. 27, no. 2, pp. 147-70.

Kimaro, HC & Nhampossa, JL 2005, 'Analyzing the problem of unsustainable helth information systems in less-developed economies: case studies from Tanzania and Mozambique', *Information Technology for Development*, vol. 11, no. 3, pp. 273-98.

Kitchenham, BA, Travassos, GH, Mayrhauser, Av, Niessink, F, Schneidewind, NF, Singer, J, Takada, S, Vehvilainen, R & Yang, H 1999, 'Towards an ontology of software maintenance', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 11, pp. 365-89.

Klint, P, van der Storm, T & Vinju, J 2010, 'DSL Tools: Less Maintenance?', in *Preliminary Proceedings of the Tenth Workshop on Language Descriptions Tools and Applications LDTA 2010*, Cyprus.

Kraemer, K & King, JL 2008, 'Information technology and administrative reform: will e-governmetn be different?', in D Norris (ed.), *E-government Research: Policy and Management*, IGI Publishing, Hershey, New York.

Krishna, S & Walsham, G 2005, 'Implementing Public Information Systems in Developing Countries: Learning from a Success Story', *Information Technology for Development*, vol. 11, no. 2, p. 18.

Kumar, R & Best, ML 2006, 'Impact and Sustainability of E-Government Services in Developing Countries: Lessons Learned from Tamil Nadu, India', *The Information Society*, vol. 22, p. 12.

Kumar, V, Mukerji, B, Butt, I & Persaud, A 2007, 'Factors for Succesful e-Government Adoption: a Conceptual Framework', *'The Electronic Journal of e-Government'*, vol. 5, no. 1, p. 14.

Kunstelj, M & Vintar, M 2004, 'Evaluating the progress of e-government development: a critical analysis', *Information Polity*, vol. 9, pp. 131-48.

Lal, R & Haleem, A 2009, 'A structural modelling for e-governance service delivery in rural India', *International Journal of Electronic Governance*, vol. 2, no. 1, pp. 3-21.

Lanubile, F & Visaggio, G 1995, 'Iterative reengineering to compensate for quick-fix maintenance', in *Proceedings of the 11th International Conference on Software Maintenance 1995 (ICSM '95)*, Opio (Nice), France pp. 140-6.

Laumer, S 2009, 'Non-Monetary Solutions for Retaining the IT Workforce', in *Proceeding of the AMCIS 2009* San Francisco, California, USA, p. 720.

Layne, K & Lee, J 2001, 'Developing fully functional E-government: A four stage model', *Government Information Quarterly*, vol. 18, p. 15.

Lee, G & Jimenez, BS 2010, 'Does Performance Management Affect Job Turnover Intention in the Federal Government?', *The American Review of Public Administration*, vol. XX, no. X, pp. 1-17.

Lee, J, Kim, HJ & Ahn, MJ 2011, 'The willingness of e-Government service adoption by business users: The role of offline service quality and trust in technology', *Government Information Quarterly*, vol. 28, no. 2, pp. 222-30.

Lee, K & Mirchandani, D 2009, 'Analyzing the dynamics of skill sets for the U.S. information systems workforce using latent growth curve modeling', in *Proceedings of the SIGMIS 47th Annual Conference on Computer Personnel Research*, Limerick, Ireland, pp. 113-20.

Lee, M-G & Jefferson, TL 2005, 'An empirical study of software maintenance of a web-based Java application', in *Proceeding of the 21st IEEE International Conference on Software Maintenance (ICSM'05)*, Budapest, Hungary.

Lee, SM, Tan, X & Trimi, S 2005, 'Current practices of leading e-government countries', *Communication of the ACM*, vol. 48, no. 10, pp. 99-104.

Leenes, R 2004, 'Local e-Government in the Netherlands: From Ambitious Policy Goals to Harsh Reality', in ITA manuscript, ITA-04-04, Vienna.

Lehman, MM & Ramil, JF 2001, 'Rules and Tools for Software Evolution Planning and Management', *Annals of Software Engineering*, vol. 11, no. 1, pp. 15-44.

Lehman, MM, Ramil, JF, Wernick, PD, Perry, DE & Turski, WM 1997, 'Metrics and laws of software evolution-the nineties view', in *Proceedings of the Fourth International Software Metrics Symposium 1997* Albuquerque, NM , USA pp. 20-32.

Liangwen, S 2009, 'A Study on E-Government Success Framework Based on IS Success Model', in *Proceeding of of the 1st International Conference on Information Science and Engineering (ICISE) 2009* Nanjing, China, pp. 2255-8.

Lin, F, Fofanah, SS & Liang, D 2011, 'Assessing citizen adoption of e-Government initiatives in Gambia: A validation of the technology acceptance model in information systems success', *Government Information Quarterly*, vol. In Press, Corrected Proof.

Liu, X & Mills, A 2007, 'Motivation and performance-related pay', in AS Preker (ed.), *Public Ends, Privater Means: Strategic Purchasing of Health Services*, The World Bank, Washington.

Lucia, AD, Pompella, E & Stefanucci, S 2005, 'Assessing effort estimation models for corrective mainenance through empirical studies', *Information and Software Technology*, vol. 47, pp. 3-15.

Luk, SCY 2009, 'The impact of leadership and stakeholders on the success/failure of e-government service: using the case study of e-stamping service in Hong Kong', *Government Information Quarterly*.

Luna-Reyes, LF & Andersen, DL 2003, 'Collecting and analyzing qualitative data for system dynamics: methods and models', *System Dynamics Review (Wiley)*, vol. 19, no. 4, pp. 271-96.

Luna-Reyes, LF & Gil-García, JR 2009, 'Using Institutional Theory and Dynamic Simulation to Understand Complex E-Government Phenomena', *Ponencia presentada en la 27th Intenational Coference of the System Dynamics Society. Albuquerque, Mew Mexico. Estados Unidos. Julio*, pp. 26-30.

Maani, KE & Cavana, RY 2007, *Systems Thinking, System Dynamics: Managing Change and Complexity*, 2 edn, Pearson Education New Zealand, North Shore 0362 New Zealand.

Maani, KE & Cavana, RY 2009, *Introduction to Systems Thinking*, Pearson Education New Zealand, Auckland.

Madachy, RJ 2008, *Software process dynamics*, IEEE / John Wiley & Sons, Hoboken, New Jersey.

Mahaney, RC & Lederer, AL 2006, 'The effect of intrinsic and extrinsic rewards for developers on information systems project success', *Project Management Journal*, vol. 37, no. 4, p. 42.

Mak, BL & Sockel, H 2001, 'A confirmatory factor analysis of IS employee motivation and retention', *Information & Management*, vol. 38, no. 5, pp. 265-76.

Marks, ML, Mirvis, PH, Hackett, EJ & Grady, JF 1986, 'Employee participation in a Quality Circle program: Impact on quality of work life, productivity, and absenteeism', *Journal of Applied Psychology*, vol. 71, no. 1, p. 61.

Martinez-Moyano, IJ & Gil-Garcia, JR 2004, 'Rules, Norms, and Individual Preferences for Action: An Institutional Framework to Understand the Dynamics of e-Government Evolution', in R Traunmüller (ed.), *Electronic Government*, Springer Berlin / Heidelberg, vol. 3183, pp. 194-9.

Martins, E & Martins, N 2011, 'The role of organisational factors in combating tacit knowledge loss in organisations', *Southern African Business Review*, vol. 15, no. 1.

Mazor, S 2009, 'Moore's law, microcomputer, and me', *Solid-State Circuits Magazine, IEEE*, vol. 1, no. 1, pp. 29-38.

McGrath, GM & More, E 1998, 'The expectancy theory of motivation applied to software engineering process modelling', in *Proceedings of the IASTED International Conference on Software Engineering* Las Vegas, Nevada, USA, pp. 142-5.

McLucas, AC 2005, *System dynamics applications: a modular approach to modelling complex world behaviour*, Argos Press, Canberra, Australia.

McMillan, S 2009, 'Government electronic and mobile service delivery: a success factors model', PhD thesis, Victoria University.

McNurlin, BC & Sprague Jr, RH 2006, *Information systems management in practice*, 7th edn, Pearson Education, Inc, Upper Saddle River, New Jersey 07458.

Melin, U & Axelsson, K 2009, 'Managing e-service development–comparing two e-government case studies', *Transforming Government: People, Process and Policy*, vol. 3, no. 3, pp. 248-70.

Memary, R & Kuan Yew, W 2009, 'An Investigation of the Expectancy Theory Elements for Motivating Employees', in *Proceediing of the 2009 International Association of Computer Science and Information Technology - Spring Conference, IACSITSC '09*, Singapore pp. 286-90.

Mendenhall, W, Beaver, RJ & Beaver, BM 1995, *A brief course in business statistics*, Wadsworth Publishing Company.

Mishra, A & Misra, S 2007, 'People management in software industry: the key to success', *SIGSOFT Software Engineering Notes*, vol. 35, no. 6, pp. 1-4.

Misra, DC 2007, 'Defining E-Government: A Citizen-centric Criteria-based Approach', in *Proceeding of the 10th National Conference on eGovernance*, Bhopal, India.

Mohamed, N, Hussin, H & Hussein, R 2006, 'Enabling change factors and IT success in the Malaysian e-government implementation', in *Proceeding of the 10th Pacific Asia Conference on Information Systems (PACIS 2006)* Kuala Lumpur, Malaysia.

Moon, MJ 2002, 'The Evolution of E-Government among Municipalities: Rhetoric or Reality?', *Public Administration Review*, vol. 62, no. 4, pp. 424-33.

Moore, JE 2000, 'One Road to Turnover: An Examination of Work Exhaustion in Technology Professionals', *MIS Quarterly*, vol. 24, no. 1, pp. 141-68.

Morecroft, JDW 2007, *Strategic Modelling and Business Dynamics: A Feedback System Approach*, John Wiley & Sons, Ltd, West Sussex PO19 8SQ, England.

Morgan, LG & Herman, JB 1976, 'Perceived consequences of absenteeism', *Journal of Applied Psychology*, vol. 61, no. 6, p. 738.

Murthy, NN, Challagalla, GN, Vincent, LH & Shervani, TA 2008, 'The Impact of Simulation Training on Call Center Agent Performance: A Field-Based Investigation', *MANAGEMENT SCIENCE*, vol. 54, no. 2, pp. 384-99.

Myers, M & Kaposi, A 2004, *A First Systems Book: Technology and Management*, 2nd edn, Imperial College Press, London.

Narayanan, S, Balasubramanian, S & Swaminathan, JM 2009, 'A matter of balance: Specialization, task variety, and individual learning in a software maintenance environment', *MANAGEMENT SCIENCE*, vol. 55, no. 11, pp. 1861-76.

Ndou, VD 2004, 'E-government for Developing Countries: Opportunities and Challenges', *The Electronic Journal on Information Systems in Developing Countries*, vol. 18, no. 1, pp. 1-24.

Nelson, RR & Todd, PA 2004, 'Peopleware: The Hiring and Retention of IT Personnel', in M Igbaria & C Shayo (eds), *Strategies for Managing IS/IT Personnel*, Idea Group Publishing, Hershey.

Nengomasha, CT, McHombu, K & Ngulube, P 2010, 'Electronic Government Initiatives in the Public Service of Namibia', *African Journal of Library, Archives & Information Science*, vol. 20, no. 2, pp. 125-37.

Ng, CSP, Gable, GG & Chan, T 2002, 'An ERP-client benefit-oriented maintenance taxonomy', *The Journal of Systems and Software*, vol. 64, pp. 87-1009.

Niessink, F & van Vliet, H 1998, 'Two case studies in measuring software maintenance effort', in *Proceeding of the 14th IEEE International*

Conference on Software Maintenance (ICSM'98), Bethesda, Maryland, USA.

Obi, T & Hai, NTT 2010, 'E-Government Project Implementation: Insight from Interviews in Vietnam', *Journal of Asia-Pacific Studies (Waseda University)*, no. 14, pp. 37-55.

Ojo, A, Janowski, T, Estevez, E & Khan, IK 2007, *Human Capacity Development for e-Government*, United Nations University - International Institute for Software Technology (UNU-IIST), Macau.

Olphert, W & Damodaran, L 2007, 'Citizen Participation and engagement in the Design of e-Government Service: The Missing Link in Effective ICT Design and Delivery', *Journal of the Association for Information Systems*, vol. 8, no. 9, p. 17.

Ostrand, TJ, Weyuker, EJ & Bell, RM 2005, 'Predicting the location and number of faults in large software systems', *IEEE Transactions on Software Engineering*, vol. 31, no. 4, pp. 340-55.

Ozkan, S & Kanat, IE 2011, 'e-Government adoption model based on theory of planned behavior: Empirical validation', *Government Information Quarterly*, vol. In Press, Corrected Proof.

Palmer, J, Speier, C, Buckley, M & Moore, JE 1998, 'Recruiting and retaining IS personnel: factors influencing employee turnover', in *Proceedings of the 1998 ACM SIGCPR Conference on Computer Personnel Research* Boston, Massachusetts, United States, pp. 286-8.

Papantoniou, A, Hattab, E, Afrati, F, Kayafas, E & Loumos, V 2001, 'Change management, a critical success factor for e-government', paper presented to The 12th International Workshop on Database and Expert Systems Applications, Munich, Germany 03-07 September 2001.

Pardo, TA & Tayi, GK 2007, 'Editorial: Interorganizational information integration: A key enabler for digital government', *Government Information Quarterly*, vol. 24, pp. 691-715.

Patel, SC & Zu, X 2009, 'E-government application development using the Six Sigma approach', *Electronic Government, an International Journal*, vol. 6, no. 3, pp. 295-306.

Paulson, D & Wand, Y 1992, 'An Automated Approach to Information Systems Decomposition', *IEEE Transaction on Software Engineering*, vol. 18, no. 3.

Perry, WE 1981, *Managing system maintenance*, Q.E.D. Information Sciences, Inc., Wellesley, MA 02181.

Petter, S, DeLone, W & McLean, E 2008, 'Measuring Information Systems Success: Models, dimensions, measures, and interrelationships', *European Journal of Information Systems*, vol. 17, p. 28.

Pidd, M 2004, 'Complementarity in Systems Modelling', in M Pidd (ed.), *Systems Modelling: Theory and Practice*, John Wiley & Sons, Ltd, West Sussex.

Pigoski, TM 1997, *Practical Software Maintenance: Best practices for managing your software investment*, John Wiley & Sons, Inc, New York.

Pokharel, M & Park, JS 2009, 'Cloud computing: future solution for e-governance', paper presented to Proceedings of the 3rd International Conference on Theory and Practice of Electronic Governance, Bogota, Colombia, 10 - 13 November 2009.

Popović, M, Atlagć, B & Kovacević, V 2001, 'Case study: a maintenance practice used with real-time telecommunications software', *Journal of Software Maintenance & Evolution: Research & Practice*, vol. 13, no. 2, pp. 97-126.

Pornphol, P & McGrath, GM 2011, 'Information systems development as an emergent knowledge research process', *The International Journal of Interdisciplinary Social Sciences*, vol. 6.

Prasad, HCS & Suar, D 2010, 'Performance assessment of Indian software professionals', *Journal of Advances in Management Research*, vol. 7, no. 2, pp. 176-93.

Pressman, RS 2005, *Software Engineering: A Practitioner's Approach*, 6th edn, McGraw-Hill Higher Education, New York.

Prybutok, VR, Zhang, X & Ryan, SD 2008, 'Evaluating leadership, IT quality, and net benefits in an e-government environment', *Information & Management*, vol. 45, no. 3, pp. 143-52.

PUSDATIN 2006, *Visi, Misi, Tupoksi, Organisasi, Fasilitas dan Layanan, serta SDM*, Pusat Data dan Informasi Pertanian, Kementerian Pertanian Republik Indonesia viewed 2 May 2010 2010, <http://www.deptan.go.id/pusdatin/pusdatin1.htm#SUMBER_DAYA_MANUSIA>.

Raja, U, Hale, DP & Hale, JE 2009, 'Modelling software evolution defects: a time series approach', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 21, no. 1, pp. 49-71.

Rasch, RH & Tosi, HL 1992, 'Factors affecting software developers' performance: an integrated approach', *MIS Quarterly*, vol. 16, no. 3, pp. 395-413.

Rashid, A, Wang, WYC & Tan, FB 2010, 'Information Systems Maintenance: A key driver of Business Process Innovation', in *Proceeding of the AMCIS 2010* Lima, Peru.

Rashid, N & Rahman, S 2010, 'An Investigation into Critical Determinants of e-Government Implementation in the Context of a Developing Nation', in K Andersen, E Francesconi, Å Grönlund & T van Engers (eds), *Electronic Government and the Information Systems Perspective*, Springer Berlin / Heidelberg, vol. 6267, pp. 9-21.

Ray, S 2010, 'Conceptualizing and Implementation of E-government in India', in CG Reddick (ed.), *Comparative E-Government*, Springer, pp. 391-407.

Reddick, CG 2004, 'Empirical Models of E-Government Growth in Local Governments', *e-Service Journal*, vol. 3, no. 2, pp. 59-84.

Rehman, M, Mahmood, AK, Salleh, R & Amin, A 2011, 'Understanding Motivators and De-motivators for Software Engineers – A Case of Malaysian Software Engineering Industry Software Engineering and Computer Systems', in JM Zain, WMb Wan Mohd & E El-Qawasmeh (eds), *Proceedings of the Second International Conference Software Engineering and Computer Systems, ICSECS 2011* Kuantan, Pahang, Malaysia, vol. 181, Part III, pp. 205-14.

Robbins, S, Bergman, R, Stagg, I & Coulter, M 2009, *Foundations of management*, 3rd edn, Pearson Education Australia, French Forest NSW Australia 2086.

Rorissa, A & Demissie, D 2009, 'The state of the art of e-government services in Africa: an analysis of relevant websites', in *Proceeding of the 42nd Hawaii International Conference on System Sciences 2009* Waikoloa, Big Island, Hawaii

Rosacker, KM & Olson, DL 2008, 'Public sector information system critical success factors', *Transforming Government: People, Process and Policy*, vol. 2, no. 1, pp. 60-70.

Sabherwal, R, Jeyaraj, A & Chowa, C 2006, 'Information System Success: Individual and Organizational Determinants', *MANAGEMENT SCIENCE*, vol. 52, no. 12, p. 16.

Şandor, SD 2006, 'Romania's Digital Divide and the Failures of E-Government', *Revista Transilvană de Ştiinţe Administrative*, no. 1, pp. 154-62.

Sang, S, Lee, J & Lee, J 2009, 'A Study on the Contribution Factors and Challenges to the Implementation of E-Government in Cambodia', *Journal of Software*, vol. 4, no. 6, p. 529.

Schalk, R & Van Rijckevorsel, A 2007, 'Factors influencing absenteeism and intention to leave in a call centre', *New Technology, Work and Employment*, vol. 22, no. 3, pp. 260-74.

Schneberger, SL 1995, 'Software maintenance in distributed computer environments: system complexity versus component simplicity', in *Proceeding of the 11th International Conference on Software Maintenance (ICSM '95)*, Opio (Nice), France, pp. 304-13.

Scholl, HJ 2007, 'Central research questions in e-government, or which trajectory should the study domain take?', *Transforming Government: People, Process and Policy*, vol. 1, no. 1, p. 22.

Schwester, R 2009, 'Examining the barriers to e-government adoption', *Electronic Journal of e-Government*, vol. 7, no. 1, p. 10.

Seaman, C 2002, 'The information gathering strategies of software maintainers', in *Proceeding of the 18th IEEE International Conference on Software Maintenance (ICSM'02)*, Montreal, Quebec, Canada p. 0141.

Sedera, D & Gable, G 2004, 'A Factor and Structural Equation Analysis of the Enterprise Systems Success Measurement Model', in *Proceeding of the Twenty-Fifth International Conference on Information Systems* Washington, DC., USA.

Sedera, D, Gable, G & Chan, T 2003, 'Survey Design: Insights from A Public Sector-ERP Success Study', in *The Seventh Pacific Asia Conference on Information Systems*, Adelaide, South Australia.

—— 2004, 'Measuring Enterprise Systems Success: The Importance of A Multiple Stakeholder Perspective', in *Proceeding of the 12th European Conference on Information Systems, ECIS 2004* Turku, Finland.

Senge, PM 2006, *The Fifth Discipline: The Art and Practice of the Learning Organization*, 2nd edn, Random House, London.

Shareef, MA, Archer, N, Sharan, V & Kumar, V 2010, 'Critical Factors for Adoption of E-government: Validity of Adoption Model in Indian Context', in CG Reddick (ed.), *Comparative E-Government*, Springer New York, vol. 25, pp. 371-89.

Shareef, MA, Kumar, U, Kumar, V & Dwivedi, YK 2009, 'Identifying critical factors for adoption of e-government', *Electronic Government, an International Journal*, vol. 6, no. 1, pp. 77-96.

Shareef, MA, Kumar, V, Kumar, U & Dwivedi, YK 2011, 'e-Government Adoption Model (GAM): Differing service maturity levels', *Government Information Quarterly*, vol. 28, no. 1, pp. 17-35.

Shi, W 2002, 'The contribution of organizational factors in the success of electronic government commerce', *International Journal of Public Administration*, vol. 25, no. 5, p. 29.

Silcock, R 2001, 'What is E-government', *Parliamentary Affairs*, vol. 54, no. 1, pp. 88-101.

Singh, Y & Goel, B 2007, 'A step towards software preventive maintenance', *ACM SIGSOFT Software Engineering Notes*, vol. 32, no. 4.

Smith, ML 2011, 'Limitations to building institutional trustworthiness through e-government: a comparative study of two e-services in Chile', *Journal of Information Technology*, vol. 26, pp. 78-93.

Sneed, HM 1997, 'Measuring the performance of a software maintenance department', in *Proceeding of the First Euromicro Conference on Software Maintenance and Reengineering 1997 (EUROMICRO97)*, Berlin , Germany p. 119.

Song, C 2010, 'Validating IS Success Factors: An Empirical Study on Webbased State or Local E-government Systems', in *Proceeding of the AMCIS 2010*, Lima, Peru, p. 191.

Sterman, JD 2000, *Business Dynamics: System Thinking and Modeling for a Complex World*, The McGraw-Hill Companies, Inc, Boston.

Sudhakar, GP, Farooq, A & Patnaik, S 2011, 'Soft factors affecting the performance of software development teams', *Team Performance Management*, vol. 17, no. 3/4, pp. 187 - 205.

Swanson, EB & Beath, CM 1989, *Maintaining information systems in organizations*, John Wiley & Sons, Chichester.

Tan, CW & Pan, SL 2003, 'Managing e-transformation in the public sector: an e-government study of the Inland Revenue Authority of Singapore (IRAS)', *European Journal of Information Systems*, vol. 12, pp. 269-81.

Tan, T, Li, Q, Boehm, B, Yang, Y, He, M & Moazeni, R 2009, 'Productivity trends in incremental and iterative software development', in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, Lake Buena Vista, FL, USA, pp. 1-10.

Tan, W-G & Gable, GG 1998, 'Attitudes of maintenance personnel towards maintenance work: a comparative analysis', *Journal of Software Maintenance: Research and Practice*, vol. 10, no. 1, pp. 59-74.

Taylor, MJ, Moynihan, E & Laws, A 1998, 'Training for software maintenance', *Journal of Software Maintenance: Research and Practice*, vol. 10, no. 6, pp. 381-93.

Taylor, MJ & Wood-Harper, AT 1996, 'Methodologies and software maintenance', *Software Maintenance: Research and Practice*, vol. 8, pp. 295-308.

Teo, TSH, Srivastava, SC & Jiang, LI 2008, 'Trust and Electronic Government Success: An Empirical Study', *Journal of Management Information Systems*, vol. 25, no. 3, pp. 99-131.

Titah, R & Barki, H 2008, 'E-Government Adoption and Acceptance: A Literature Review and Research Framework', in D Norris (ed.), *E-Government Research: Policy and Management*, IGI Publishing, Hershey, New York.

Tolbert, CJ, Mossberger, K & McNeal, R 2008, 'Institutions, Policy Innovation, and E-Government in the American States', *Public Administration Review*, vol. 68, no. 3, pp. 549-63.

Tsai, H-Y, Compeau, D & Haggerty, N 2007, 'Of races to run and battles to be won: Technical skill updating, stress, and coping of IT professionals', *Human Resource Management*, vol. 46, no. 3, pp. 395-409.

Tsai, N, Choi, B & Perry, M 2009, 'Improving the process of E-Government initiative: An in-depth case study of web-based GIS implementation', *Government Information Quarterly*, vol. 26, no. 2, pp. 368-76.

Turver, RJ & Munro, M 1994, 'An early impact analysis technique for software maintenance', *Journal of Software Maintenance: Research and Practice*, vol. 6, no. 1, pp. 35-52.

United Nations 2008, *UN E-Government Survey 2008, From E-Government to Connected Governance*, United Nations, <http://unpan1.un.org/intradoc/groups/public/documents/UN/UNPAN028607.pdf>.

—— 2010, *United Nations E-Government Survey 2010: Leveraging e-government at a time of financial and economic crisis*, Department of Economic and Social Affairs, United Nations, <http://unpan1.un.org/intradoc/groups/public/documents/un/unpan038851.pdf>.

van Vliet, H 2008, *Software engineering: principles and practice*, John Wiley & Sons, Chichester.

Vennix, JAM 1996, *Group model building : facilitating team learning using system dynamics*, J. Wiley, Chichester ; New York

von Bertalanffy, L 1950, 'An Outline of General System Theory', *The British Journal for the Philosophy of Science*, vol. I, no. 2, p. 32.

Vroom, VH 1995, *Work and Motivation*, The Jossey-Bass Management Series, Jossey-Bass Inc., San Fransisco.

Wand, Y & Weber, R 1990, 'An Ontological Model of an Information System', *IEEE Transaction on Software Engineering*, vol. 16, no. 11, p. 11.

Wang, S, Schach, SR & Heller, GZ 2001, 'A case study in repeated maintenance', *Journal of Software Maintenance & Evolution: Research & Practice*, vol. 13, no. 2, pp. 127-41.

Wang, Y-S & Liao, Y-W 2008, 'Assessing eGovernment System Success: A Validation of the DeLone and McLean Model of Information Systems Success', *Government Information Quarterly*, vol. 25, p. 17.

Wangpipatwong, S, Chutimaskul, W & Papasratorn, B 2008, 'Understanding citizen’s continuance intention to use e-government website: A composite view of technology acceptance model and computer self-efficacy', *'The Electronic Journal of e-Government*, vol. 6, no. 1, pp. 55–64.

Weerakkody, V, El-Haddadeh, R & Al-Shafi, S 2011, 'Exploring the complexities of e-government implementation and diffusion in a developing country: Some lessons from the State of Qatar', *Journal of Enterprise Information Management*, vol. 24, no. 2, pp. 172-96.

Whitten, JL & Bentley, LD 2007, *Systems Analysis and Design Methods* 7th edn, McGraw-Hill/Irwin, New York.

Wolstenholme, EF 1999, 'Qualitative vs Quantitative Modelling: The Evolving Balance', *The Journal of the Operational Research Society*, vol. 50, no. 4, pp. 422-8.

Wood-Harper, T, Ibrahim, O & Ithnin, N 2004, 'An Interconnected Success Factor Approach for Service functional in Malaysian Electronic Government', in M Janssen, HG Sol & RW Wagenaar (eds), *Proceeding of the Sixth International Conference on Electronic Commerce, ICEC'04*, Delft, The Netherlands, pp. 446-50.

Xiong, CJ, Xie, M & Ng, SH 2011, 'Optimal software maintenance policy considering unavailable time', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 23, no. 1, pp. 21-33.

Yeh, D & Jeng, J-H 2002, 'An empirical study of the influence of departmentalization and organizational position on software maintenance', *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 14, no. 1, pp. 65-82.

Yildiz, M 2007, 'E-government research: Reviewing the Literature, Limitation, and Ways Forward', *Government Information Quarterly*, vol. 24, p. 20.

Yin, RK 2003, *Case Study Research, Design and Methods*, Third edn, Applied Social Research Methods Series, Sage Publishing, Inc, Thousand Oaks, California.

Yokota, A 2011, 'Maintenance Trends in ERP Systems', in *Proceedings of the AMCIS 2011*, Detroit, USA.

Zarei, B, Ghapanchi, A & Sattary, B 2008, 'Toward national e-government development models for developing countries: A nine-stage model', *The International Information & Library Review*, vol. 40, no. 3, pp. 199-207.

Zhang, H 2009, 'An investigation of the relationships between lines of code and defects', in *Proceeding of the 2009 IEEE International Conference on Software Maintenance, ICSM 2009*, Edmonton, Canada, pp. 274-83.

Zhang, P, Mao, X, Tang, X & Zeng, D 2008, 'A Hybrid e-Government Model: Case Studies in Shanghai', in H Chen, L Brandt, V Gregg, R Traunmüller, S Dawes, E Hovy, A Macintosh & CA Larson (eds), *Digital Government*, Springer US, vol. 17, pp. 697-718.

Appendices