# Towards a Formal Model of E-Government Success Factors: System Dynamics Modelling of Human Resource Impacts During Maintenance

Gunadi Gunadi

*School of Management and Information Systems Victoria University*
*Melbourne, Australia*
and
*Gajayana University*
*Malang, Indonesia*

gunadi.gunadi@live.vu.edu.au

G. Michael McGrath

*Centre for Tourism and Services Research Victoria University*
*Melbourne, Australia*

Michael.McGrath@vu.edu.au

Geoffrey Sandy

*School of Management and Information Systems Victoria University,*
*Melbourne, Australia*

Geoff.Sandy@vu.edu.au

# Towards a Formal Model of E-Government Success Factors: System Dynamics Modelling of Human Resource Impacts During Maintenance

## Abstract

*One of our unknown futures concerns the role of Information technology (IT) for business and society. This paper describes a conceptual model of factors and their relationships that determine the success of effective and efficient use of IT for government service delivery. Specifically, it formally models the domain of Information Systems Management (ISM) using a system dynamics approach. The maintenance stage of this domain is important in ensuring sustainability. Expectancy theory from the field of human resource (HR) management is used as a basis to model the dynamic structure of some key success factors and their relationships. The next stage of this research will be to validate the model using desk checks and field tests.*

**Keywords:** e-government, success factors and relationships, system dynamics, software maintenance, expectancy theory

## INTRODUCTION

One of our unknown futures concerns the role of Information Technology (IT) for business and society. The internet revolution continually surprises and how it will progress is uncertain. A microcosm of this revolution is the application of electronic and mobile technologies for the development, delivery and maintenance of government service provision. This paper focuses on modelling the important relationships needed for success, plus testing its practical usefulness for key decision makers.

Electronic and mobile government applications are, essentially, types of information systems (IS) (Heeks 2005) and there is much about IS development and implementation that is uncertain (HarrisCollins & Hevner 2009, Jorgensen & Molokken 2002). Notably, of course, business rules may change due to volatility in the external environment over which organizations (private and government) have no control. However, IS project managers and their sponsors inevitably have to confront a variety of what Vennix (1996) refers to as 'messy' problems: i.e. problems characterized by complexity, uncertainty, inter-related sub-problems and recursive dependencies. Small wonder then that most IS applications are still delivered late and over-budget (Gauld 2007, Hallows 2005).

The rapid growth of the development and implementation of e-government service delivery has encouraged many researchers to study factors that affect success and failure. An extensive review of the literature on success factors reveals that there are many factors that have been conceptualized and proposed, or empirically confirmed. Some examples of these identified in previous studies include financial resources (Rose 2004, Sandy & McMillan 2005), ICT infrastructures (Furuholt & Wahid 2008, Harijadi 2004, Ndou 2004), political leadership (Furuholt & Wahid 2008, GrabowDruke & Siegfried 2004, Heeks 2008a, Misuraca 2007, Rosacker & Olson 2008), external pressures (Heeks 2008a, Reddick 2004), management (Furuholt & Wahid 2008, Heeks 2008a, Ndou 2004), and population size (Leenes 2004, Moon 2002).

A variety of models have been developed to study and analyse e-government and its success. For instance, Beynon-Davies (2007) presented a meta-model that organised inter-related horizontal and vertical dimensions of e-government. Gupta, Dasgupta and Gupta (2008) employed the Unified Theory and Use of Technology Model to study internal user acceptance of e-government. Evangelidis (2005) introduced a risk assessment model. Rosacker and Olson (2008) modelled critical success factors of e-government projects. Altameem, Zairi and Alshawi (2006) proposed a model for e-government implementation based on three categories of critical success factors. Gichoya (2005), adapting DeLone and McLean's (1992) earlier work, developed a framework of e-government success factors implementation in the Kenyan context.

However, these previous studies understated the complexity of e-government success factors in the sense that it is not only individual (and sets of) factors that promote the successful design, development and delivery of e-government service but also their relationships. Consequently, this study investigates success factors and their relationships as determinants of e-government success. Given the complexity of such relationships, much might be gained by conceptualising e-government success factors as a system. A similar observation regarding the complex relationships of e-government success factors was also taken by Titah and Barki (2008), but they recommended a conceptual development of success factors relationships from a multi-dimensional and multi-level point of view.

E-government success is specified by its attributes and measures. Fulfilment of all required measures of e-government success attributes is the way in which success will be realized. Fulfilment of the required measures means there is a significant change level, either qualitatively or quantitatively, for each e-government success measure. However, the change of a success measurement level depends on the success factors that influence the measure. This is complex because of the existence of relationships amongst success factors, the feedback effect of change levels of success measures, and the time required for effects to fully manifest themselves.

As distinct from the approach adopted by Titah and Barki (2008), however, this study employs a system dynamics approach to model e-government success factors and their relationships. This approach has the capability to address the inherent complexity of the domain and to reveal, explicitly, its dynamic structure (Sterman 2000). It might also identify instances of success factors that appear to have a major impact on other factors, or that significantly influence overall e-government success.

As an IS, e-government systems success can be defined by referring to the antecedents variables of the well-known DeLone and McLean's (2003) IS success model and the importance of e-government sustainability (Horiuchi 2006). Therefore, an e-government system is defined as successful if the system is able to sustainedly achieve a quality system that delivers quality information and services. The usual measures of information success are also employed: specifically, systems should be free from errors, produce accurate outputs, deliver what users really need and do so within agreed budgets and time constraints (Yourdon 1989).

## RESEARCH AIMS

From a review of the relevant literature, it is apparent that many e-government systems have not been successful. The major aim of this research is to model e-government success factors and their relationships. System dynamics will be used as the main modelling method.

The model is expected to be able to unearth the success factors and dynamic structure of their relationships in influencing the success of the e-government within a specific domain. When implemented (as a decision support system with powerful simulation capabilities) the hope is that this model will be able to help key stakeholders of e-government systems in the planning, development, implementation and maintenance of their systems.

# THE BROAD MODEL

## E-Government Success Factors System

There are a large number of e-government success factors and taking all of these into account when developing an e-government system is an extremely difficult undertaking. Therefore, the success factors need to be organized into a manageable size so that system sponsors, developers and users can focus on a particular aspect while, at the same time, still adopt a holistic approach to decision-making.

In IS analysis and design, decomposition has long been one of the most important tools employed in dealing with large and complex systems (Paulson & Wand 1992). Generally, the aim is to ensure that decomposition results in highly cohesive success factors within each subsystem and loose coupling between subsystems. However, the process that lead to a “good” systems decomposition mainly relies on personal expertise (Paulson & Wand 1992) and, to date, there is no comprehensive and prescriptive theory that provides all that is necessary to ensure a good decomposition in conceptual modelling (Burton-Jones & Meso 2006).

Here, consistent with Curtis, Kellner and Over (1992), the success factors will be decomposed into a manageable number of loosely-interrelated success factor subsystems according to their functions. The e-government literature indicates that the success factors can be organized according to:

- those that have a critical impact on the whole e-government life-cycle (AlShihi 2006);
- those that are associated with IS that are necessary to support human resources activities in delivering services and values to customers (Beynon-Davies 2007);
- those that are concerned with creating and delivering value to government customers (Beynon-Davies 2007); and
- those that are related to entities and activities in the external environment (Heeks 2005).

Accordingly, this study argues that there are four high-level e-government success factor subsystems: i) one that deals with driving and governing factors; ii) one dealing with IS management issues; iii) one concerning the use of the e-government to deliver services; and, iv) the subsystems associated with factors relevant to the external environment. All of these subsystems are also systems themselves because of the hierarchical nature of any system. These four high-level subsystems can be further decomposed into lower-level sub-subsystems. Following the decomposition model representation scheme presented in Whitten and Bentley (2007), this two level decomposition is depicted in Figure 1.

**Figure 1 is inserted here.**

A top-level representation of the causal relationships among these subsystems is presented in Figure 2. The arrows between subsystems represent possible relationships. However, precisely how subsystems influence each other will not be explored at this stage. It is expected that many connections will emerge from the development of the success factors relationships at lower levels.

**Figure 2 is inserted here.**

**System dynamics modelling focus**

The e-government success factors system decomposition facilitates the selection of a particular aspect of the system as the study's focus. As stated by Heeks (2005), the core of an e-government system is the IS and, as a consequence, this paper will pursue success factors associated with the ISM subsystem of the decomposition model and how these success factors relate to each other in influencing e-government success. However, as the IS domain is very broad, the focus will be further narrowed down to the maintenance stage of the IS life-cycle and to HR issues associated with that stage. Reasons for concentrating on these two areas are: i) with the move to pre-packaged systems, the maintenance stage (while always important) is now more critical than ever (Grubb & Takang 2003, Layzell & Macaulay 1994); ii) it has long been recognised that, whatever the technical brilliance of an IS, little will be achieved if 'people' factors are ignored (Boehm 1981, Faraj & Sproull 2000); and iii) uncertainty is a central theme of this paper and there is little in IS work more uncertain than issues associated with the individuals that constitute a project team.

Sustainability is one of the most significant attributes associated with e-government success (Horiuchi 2006). It represents the ability of the system to deliver its intended function during its life-time, especially to keep the system up and running smoothly, and to be able to adapt to changing business requirements. In addition, incremental development (currently used with most IS) demands sustainable improvement. Thus, as a starting point for this study, a focus on the maintenance phase was considered appropriate. Furthermore, there is much research that points to the importance of 'people' issues along all phases of the systems development and maintenance life-cycle (see, e.g. Abdel-Hamid & Madnick 1991). Consequently, as noted above, HR aspects during system maintenance were selected as the domain for development of the initial version of our detailed model.

System dynamics (SD) techniques and tools were chosen as the means to specify and implement the detailed model. The SD approach was originally developed by Forrester (1961) and popularised more recently by Senge (2006). It is a particularly appropriate modelling approach where time and feedback loops are important, and where there is considerable complexity, ambiguity and uncertainty (Vennix 1996). Figure 3 displays part of our model developed using the SD modelling tool, STELLA/iThink® (High Performance System 1994). Users of STELLA/iThink® (and similar products) develop much of their models visually. Little mathematical sophistication is required of the user since the system provides considerable guidance in creating the difference equations that underpin SD models. In general, SD modelling tools may be employed for both descriptive and predictive purposes.

**Figure 3 is inserted here.**

Figure 3 represents a popular theory of motivation: namely expectancy theory (DuBrin 2001) and this is used as the conceptual basis that underpins the motivational aspects of the detailed model presented in this paper. The basic building blocks of SD models are stocks (represented as rectangles), flows (represented as arrows with circular flow regulators attached) and converters (represented as circles). The stocks in our model are *Effort, Performance* and *Rewards.* There is a level associated with each stock, which can be an actual value or a value bounded by some artificial scale (as is the case with our model, where all stock levels are measured on a 0-100 scale). For example, a performance level of 50 is average, while values of 70 and 30 indicate good and poor performance levels respectively. Stock levels vary with flows, which may be inflows, outflows or bi-flows. For example, *effort varn* is a bidirectional flow such that:

$$Effort_t = f(ep\ expectancy_t,\ effort\ adj\ time_t).$$

That is, in our model, the effort level at time, *t,* is a function of *ep expectancy* and *effort adj time* at time *t.* These equations are the foundation of iThink®'s formidable simulation capabilities. The third of our basic constructs, converters, serve a utilitarian role: they hold values for constants, calculate mathematical relationships and serve as repositories for graphical functions. In general, they convert inputs into outputs (hence, the name, 'converter').

Expectancy theory is actually a group of theories based on a rational-economic view of people. All versions of the theory are underpinned by the principle that a party's actions are based on: i) the expectation that an act will result in a given outcome; and ii) the attractiveness of that outcome to the party. Effort is linked to performance and performance, in turn, is linked to rewards. If a breakdown occurs at any point in this chain then, ultimately, less effort will be put in (i.e. motivation will suffer). These linkages and feedback loops are evident in the model presented in Figure 3. Note also that, through our various adjustment time parameters *(effort, perf* and *reward adj time),* this model allow for delays in the system.

The above provides a brief overview of SD modelling. We now turn our attention to a more comprehensive coverage of the detailed model.

## THE DETAILED MODEL

The detailed model is actually comprised of two inter-dependent and inter-connected sub-models: one covering system maintenance and the other dealing with HR aspects. The models are reasonably complex, so space constraint means that a detailed specification is not presented here. Instead, a brief overview of the two models and an indication of how instantiations of model variables and relationships (critical to STELLA/iThink® simulations producing sensible outputs) were arrived at are provided.

The maintenance model is presented in Figure 4. Maintenance begins when users report an apparent problem or observation to an ISM unit. Maintenance requests (MRs) are generated randomly over time. They may be triggered by a software fault found by users or, more occasionally, by ISM personnel. Any accepted MR will be categorized, prioritized and assigned to a staff member or a team. In general, MRs will be classified as either corrective or enhancive (International Organization for Standardization 2006). A higher priority will be assigned to corrective than enhancive MRs. Once work is completed on MRs, they flow to *Correction Delivered* and *Enhc Delivered* stocks (towards the right of Figure 4, top and bottom). The maintenance may also cause ripple effects that, in turn, produce further errors or cause previously undetected errors to emerge (the *Recurrent Faults* stock to the right of Figure 4). These are fed back into *Maintenance Request* as new MRs. Staff maintenance performance can be evaluated from *total delivery* irrespective of whether the delivery causes further errors or not. On the other hand, the *Recurrent Faults* can be interpreted as the degree of the maintenance quality if it can be assumed that system development has produced an error-free system. In this case, the rate of recurrent faults is determined by *learning and training factor*. Further, overall system maintenance can be evaluated from *delivery and request ratio* in which a value close to 1 is expected (note that there is a delay between MRs and maintenance deliveries). If the maintenance is efficient and effective for most MRs (especially critical ones) then e-government system sustainability is all but guaranteed.

A quick glance at Figure 4 reveals that the speed and effectiveness with which MRs are processed depends on many factors. These include system size, staff productivity, available manpower and staff learning. As noted previously, the extent to which these variables are accurately instantiated is one of the major determinants of simulation accuracy. As an example, consider *System Size* where, intuitively, it seems reasonable to assume that larger systems are generally more complex and, consequently, will generate more MRs than smaller systems. Fortunately, a great deal of previous software engineering research has been devoted to precisely this issue and we may take advantage of this.

**Figure 4 is inserted here.**

System size or complexity depends on several factors, including the problem domain, computing environment, and component variety (Schneberger 1995). Software complexity can be determined from its dynamic complexity, coordinative complexity (BankerDavis & Slaughter 1998), source line of code (SLOC), or function points (AhnSuhKim & Kim 2003). Mainly due to its simplicity (and the fact that data is often readily available), SLOC has been used as the basic unit of analysis in many studies (Andersson & Runeson 2007, Jorgensen 1995, LuciaPompella & Stefanucci 2005, OstrandWeyuker & Bell 2005, Zhang 2009); in particular those addressing the relationship between SLOC and faults. Although the evidence is not entirely conclusive, many studies have indicated a direct relationship between size and faults. For example, Ostrand, Weyuker and Bell (2005) showed that 'the number of faults is proportional to the number of lines of code', and '... the model generally predicts a large number of faults for large files'. Zhang (2009) using Weibull's distribution model showed that there is a relationship between SLOC of the software module and defect numbers. More specifically, this study demonstrated that larger modules tend to have more defects, and that the first 20% of the 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 first 10% of the largest modules.

As a consequence, a decision was made to use SLOC as a surrogate for the system complexity. Furthermore, by using Burch and Kung's (1997) more detailed analysis of 651 reported faults, the graph presented in Figure 5 was used to establish the relationship between system size and MRs, and is represented as the converter *IofSysSonNMR (Impact of System Size on New MRs)* in Figure 4.

**Figure 5 is inserted here**.

We now turn our attention to the HR sub-model and the core of this, presented in Figure 6, is actually a customisation of the generic expectancy theory model used to introduce SD concepts in the previous section (as should be evident from the *Staff Effort* → *Staff Productivity* → *Reward to Staff* links). The *Staff Productivity* is one key variable from this sub-model that is fed back into the maintenance model.

**Figure 6 is inserted here.**

Staff effort is obviously a prime determinant of productivity and has been defined by Abdel-Hamid and Madnick (1991) as the 'actual fraction of a man-day on a project'. To model this, a bi-flow that specifies effort variation over time and a stock that represents cumulative effort was implemented as in (McGrath & More 1998). The first is represented by *staff effort varn,* while the second is modelled by *Staff Effort*. The *staff effort varn* is the variation of working hours solely dedicated to maintenance activities by staff in a single working day. An increase or decrease in the *staff effort varn* will result in variation to the *Staff Effort* and this, in turn, will impact on productivity (assuming that other influencing factors are constant).

The staff effort is firstly determined by nominal effort, modelled as *staff base effort* and this is the actual normal fraction of working hours in one working day spent by staff on an assigned maintenance task. This is to accommodate the fact that, normally, staff also spend their working hours undertaking administrative and personal activities (such as emailing, preparing coffee, etc.). Following Hamid and Madnick (1991), the value of the *staff base effort* is set to 0.6. This value means that if in a working day there are 8 working hours then staff dedicate *(0.6 x 8)* working hours on maintenance activities in a day. This value was also employed to initiate the level of the *Staff Effort*.

The staff effort is also influenced by their judgement on the comparison between expended effort and resultant performance. There is a tendency to increase effort to a certain limit if the performance is lower than expected (negative difference) or to slow down (reduce) the effort if the performance is higher than expected (positive difference) (Abdel-Hamid & Madnick 1991 Ch. 7). This concept is modelled by *perf eff difference* and *IofPerf on Effort.* The first variable is used to evaluate the difference between actual performance and effort. The range of the *perf eff difference* value is from -2 to 2. The value will be minus if the performance is less than effort. The dimension of this variable is *task/person-quarter*. The second variable is expressed as a graph. It is a multiplier of the *staff base effort.* The value of the *perf eff difference* is used as input, and the value of the *IofPerf on Effort* is the output and ranges from -1 to 1. It is *dimensionless*. The graph of the *IofPerf on Effort* is presented in Figure 7. Regarding the rewards factor, intuitively, staff will decrease (increase) their effort if the accepted reward is less (more) than they expect considering their performance. This concept is modelled by *rwd perf difference* and *IofRPdiff on Effort*.

Total value of the *IofPerf on Effort* and the *IofRPdiff on Effort* can be negative, positive or zero. Multiplication of this value with the *staff base effort* causes the *Staff Effort* to vary. The value of the

*Staff Effort* is set to be non negative; and a value larger than 1.0 reflects staff's willingness to work overtime.

**Figure 7 is inserted here.**

Having specified the model and implemented it within STELLA/iThink®, users (particularly IS department and project managers) may use it to assist with their planning by posing 'what if' style questions. For example, they may wish to determine possible impacts of increasing or decreasing their staff maintenance levels, of altering the corrective maintenance/enhancements staff allocation ratio, of using overtime to attempt to clear backlogs, of long and short-term impacts of additional training, of the effects of an increase in staff turnover etc. A typical simulation output is presented in Figure 8. This graph represents cumulative values of corrective and enhancive maintenance delivered. Within the mid-period, corrections delivered overtake enhancements delivered. This is because there is a period in which the percentage of correction requests is much larger than enhancement requests. The cumulative pattern is logarithmic (not exponential) because, as one might expect, detected faults reduce over time. On the other hand, users will always find ways in which systems might be improved and, consequently, enhancement requests will never drop off entirely.

**Figure 8 is inserted here.**

## MODEL VALIDATION

The model will be validated through a combination of desk checking and field tests. The former is being conducted in combination with model development and, to date, results appear to be reasonable: i.e. simulation outputs appear to conform with what the software engineering literature suggests. Field testing will take place in late-2010, where data from actual e-government system will be used to evaluate the validity of the model resulting from desk checking and the accuracy of simulation results.

It should be noted that SD models are notoriously difficult to validate (Richardson & Pugh 1981). As noted by Forrester and Senge (1980), there is no single test which might be employed to validate an SD model but, rather, confidence in the model accumulates gradually as it passes more tests and as new points of correspondence between the model and empirical reality are identified. Maani and Cavana (2007), drawing on the work of Coyle (1983), describe this process as consisting of:

- Verification tests – which focus on the equivalence between the structure and parameters of the real system and the model;
- Validation tests – which are concerned with demonstrating the correspondence between the behaviour of the real system and the model; and
- Legitimation tests – which determine whether the model is in accord with any generally-accepted system rules.

Essentially, the aim of validation is to 'show 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). An excellent example of how much of this can be accomplished through desk checking has been provided by Georgantzas (2003) where statistical measures, such as coefficient of determination and Theil's inequality statistics (TIS) (Theil 1966), were employed to compare the predictive results of an SD model focused on various key measures of the performance of Cyprus hotels against actual data (over a 40 year period). Similarly, we could subject the model developed in this paper to similar tests, concentrating on measures for which data is readily available (such as MRs and maintenance effort*)*. An example of the type of output that results from this type of analysis is presented in Figure 9.

**Figure 9 is inserted here.**

The basis of Theil's approach is that the mean square error (MSE) is divided into three components: i) bias ($U^m$)*;* ii) unequal variation ($U^s$)*;* and iii) unequal co-variation ($U^c$). The sum of all three components equals one and, briefly, a large $U^m$ indicates a potentially serious systemic error and, to a somewhat lesser extent, this applies to $U^s$ as well. If $U^c$ is large though, most of the error is unsystematic and, as noted by Sterman (2000): 'a model should not be faulted for failing to match the random component of the data'. The sample TIS results presented in Figure 9 indicate that, in this case, model behaviour provides a reasonable approximation to reality. Nevertheless, there is significant room for improvement: specifically, the variance in the proposed model is considerably greater than that of the actual data. The TIS results, however, are also useful in that they quantify the extent of the various error types.

## DISCUSSION AND CONCLUSION

This paper has demonstrated the implementation of the systems dynamics method in modelling e-government success factors and their relationships. The demonstration chooses two inter-related sub-domains concerning systems maintenance and HRs. These are two of the more important processes associated with effective and efficient electronic and mobile government service delivery. Within and between these two sub-domains, the modelling method has revealed complex and dynamic relationships among the success factors. Furthermore, the method has explicated feedback relationships of staff effort, productivity, e-government system maintenance performance, and reward.

The next stage in this research will be to further validate the model through a combination of various desk checking and field tests. One of the available validation methods for the system dynamics model has been indicated by this paper. It is planned that the model will be field tested within the context of a developing country. Within this context, e-government system sustainability, which in many cases depends on information system maintenance, is an important problem. This is based on the fact that there is a high percentage of 'sustainability failure' in the developing countries (Heeks 2008b). In addition, most of the developing countries e-government systems are still at the emerging or enhanced level (United Nations 2008), which means system improvement through maintenance is necessary.

Data collected from Indonesian case study will be used to evaluate, and confirm or otherwise, the model and the accuracy of the simulation results. The case study is selected to represent developing countries for two reasons. Firstly, according to the United Nations (2008) the Indonesian e-government system level is within the same category as many other developing countries. Secondly, based on the preliminary observations through Indonesian government websites, there have been various government departments and levels which have developed and implemented e-government systems with disparate levels of system sophistication. By referring to the published results of a yearly program organised by a non-government organisation that provides Indonesian e-Government awards (Majalah Warta Ekonomi 2008), it is indicated that the service level of some of the current e-government implementations can be improve upon. Accordingly, the generalisability of the resulting models is placed in the wider context of developing countries

In conclusion, it is possible and desirable that e-government success factors and their relationships be formally modelled and validated. Such research is valuable in ensuring that those decision-makers responsible for the design, development and maintenance of e-government service delivery do indeed make the “right” decisions.

# References


Abdel-Hamid T & Madnick SE (1991) *Software project dynamics: an integrated approach,* Prentice Hall, Engelwood Cliffs, New Jersey.

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,* 15: 71-85.

AlShihi H (2006) Critical Factors in the Adoption and Diffusion of E-Government Initiatives in Oman. Unpublished doctoral thesis, School of Information System, Victoria University, Melbourne

Altameem T, Zairi M & Alshawi S (2006) Critical Success Factors of E-Government: A Proposed Model for E-Government Implementation. *Innovations in Information Technology*: 1-5.

Andersson C & Runeson P (2007) A replicated quantitative analysis of fault distributions in complex software systems. *IEEE Transactions on Software Engineering,* 33 (5): 273-286.

Banker RD, Davis GB & Slaughter SA (1998) Software developement practices, software complexity, and software maintenance. *Management Science,* 44 (4): 433-450.

Beynon-Davies P (2007) Models for e-government. *Transforming Government: People, Process and Policy,* 1 (1): 22.

Boehm BW (1981) *Software Engineering Economics,* Prentice-Hall, Inc, Englewood Cliffs, New Jersey 07632.

Burch E & Kung H-J (1997) Modeling software maintenance request: a case study. *13 th International Conference on Software Maintenance (ICSM'97).* Bari, Italy.

Burton-Jones A & Meso PN (2006) Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis. *Information Systems Research,* 17 (1): 38-60.

Coyle RG (1983) The technical elements of the system dynamics approach. *European Journal of Operational Research,* 14 (4): 359-370.

Curtis B, Kellner ML & Over J (1992) Process Modeling. *Communication of the ACM,* 35 (9).

DeLone WH & McLean ER (1992) Information Systems Success: the Quest for the Dependent Variable. *Information Systems Research,* 3 (1): 60-95.

DeLone WH & McLean ER (2003) The DeLone and McLean Model of Information System Success: A Ten-Year Update. *Journal of Management Information Systems,* 19 (4): 9-30.

DuBrin AJ (2001) *Leadership: research findings, practice, and skilss,* Houghton Mifflin, Boston.

Evangelidis A (2005) FRAMES - A Risk Assessment Framework for e-Services. *Electronic Journal of e-Government,* 2 (1): 21-29.

Faraj S & Sproull L (2000) Coordinating Expertise in Software Development Teams. *Management Science,* 46 (12): 1554-1568.

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,* 14: 209-228.

Furuholt B & Wahid F (2008) E-government Challenges and the Role of Political Leadership in Indonesia: the Case of Sragen. *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008).* IEEE.

Gauld R (2007) Public sector information system project failures: Lessons from a New Zealand hospital organization. *Government Information Quarterly,* 24: 102 -114.

Georgantzas NC (2003) Tourism dynamics: Cyprus' hotel value chain and profitability. *System Dynamics Review (Wiley),* 19 (3): 175-212.

Gichoya D (2005) Factors Affecting the Successful Implementation of ICT Projects in Government. *The Electronic Journal of e-Government,* 3 (4): 175-184.

Grabow B, Druke H & Siegfried C (2004) Factors for success for local community e-government. *German Journal of Urban Studies (DfK),* 45 (2): 1-22.

Grubb P & Takang AA (2003) *Software maintenance: concepts and Practice,* 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,* 17: 140-154.

Hallows J (2005) Information Systems Project Management : How to Deliver Function and Value in Information Technology Projects. 2 ed. AMACOM.

Harijadi DA (2004) Developing E-Government, The Case of Indonesia. *E-Government Workshop APECTEL29.* Hong Kong, China.

Harris ML, Collins RW & Hevner AR (2009) Control of Flexible Software Development Under Uncertainty. *Information Systems Research,* 20 (3): 400-419.

Heeks R (2005) Implementing and Managing eGovernment: An International Text SAGE Publications London.

Heeks R (2008a) A Factor Model for eGovernment Success and Failure. Accessed 1 April 2009 from www.egov4dev.org/success/evaluation/factormodel.shtml

Heeks R (2008b) Success and Failure in eGovernment Projects. Accessed 12 August 2008 from www.egov4dev.org/success/index.shtml

High Performance System (1994) *Getting Started with iThink: A Hands-On Experience,* High Performance System, Inc., Hanover, NH.

Horiuchi C (2006) Creating information systems quality in government setting. IN Duggan, EW & Reichgelt, H (Eds.) *Measuring information systems delivery quality.* Idea Group Publishing, Hershey.

International Organization for Standardization (2006) International Standard ISO/IEC 14764. *Software Engineering -- Software Life Cycle Processes -- Maintenance.* ISO/IEEE.

Jorgensen M (1995) An empirical study of software maintenance tasks. *Software Maintenance: Research and Practice,* 7: 27-48.

Jorgensen M & Molokken K (2002) Combination of software development effort prediction intervals: why, when and how? *Proceedings of the 14th international conference on Software engineering and knowledge engineering.* ACM Ischia, Italy.

Layzell PJ & Macaulay LA (1994) An investigation into software maintenance - Perception and practices. *Journal of Software Maintenance: Research and Practice,* 6 (3): 105-120.

Leenes R (2004) Local e-Government in the Netherlands: From Ambitious Policy Goals to Harsh Reality. *ITA manuscript.* Vienna.

Lucia AD, Pompella E & Stefanucci S (2005) Assessing effort estimation models for corrective mainenance through empirical studies. *Information and Software Technology,* 47: 3-15.

Maani KE & Cavana RY (2007) *Systems Thinking, System Dynamics: Managing Change and Complexity,* Pearson Education New Zealand, North Shore 0362 New Zealand.

Majalah Warta Ekonomi (2008) Warta eGov: Home. Accessed 3 November 2008 from http://jakarta.wartaegov.com/

McGrath GM & More E (1998) The expectancy theory of motivation applied to software engineering process modelling. *IASTED Conference Software Engineering.* Las Vegas, Nevada, USA.

Misuraca GC (2007) *E-Governance in Africa: From Theory to Action - A Handbook on ICTs for Local Governance,* AFRICA WORLD PRESS and International Development Research Centre, Trenton, New Jersey, US and Ottawa, Canada.

Moon MJ (2002) The Evolution of E-Government among Municipalities: Rhetoric or Reality? *Public Administration Review,* 62 (4): 424-433.

Ndou VD (2004) E-government for Developing Countries: Opportunities and Challenges. *The Electronic Journal on Information Systems in Developing Countries,* 18 (1): 1-24.

Ostrand TJ, Weyuker EJ & Bell RM (2005) Predicting the location and number of faults in large software systems. *IEEE Transactions on Software Engineering,* 31 (4): 340-355.

Paulson D & Wand Y (1992) An Automated Approach to Information Systems Decomposition. *IEEE Transaction on Software Engineering,* 18 (3).

Reddick CG (2004) Empirical Models of E-Government Growth in Local Governments. *e-Service Journal,* 3 (2): 59-84.

Richardson GP & Pugh AL (1981) *Introduction to System Dynamics Modeling with Dynamo,* MIT Press.

Rosacker KM & Olson DL (2008) Public sector information system critical success factors. *Transforming Government: People, Process and Policy,* 2 (1): 60-70.

Rose M (2004) Democratizing information and communication by implementing e-government in Indonesia regional government. *The International Information & Library Review,* 36: 219-246.

Sandy GA & McMillan S (2005) A Success Factors Model for M-Government. in Kushchu, I & Kuscu, MH (Eds.) *EURO mGOV 2005.* Mobile Government Consortium International LLC University of Sussex, Brighton, UK.

Schneberger SL (1995) Software maintenance in distributed computer environments: system complexity versus component simplicity. *The 11th International Conference on Software Maintenance.* Opio (Nice), France.

Senge PM (2006) *The Fifth Discipline: The Art and Practice of the Learning Organization,* Random House, London.

Sterman JD (2000) *Business Dynamics: System Thinking and Modeling for a Complex World,* The McGraw-Hill Companies, Inc, Boston.

Theil H (1966) *Applied economic forecasting,* North-Holland, Amsterdam.

Titah R & Barki H (2008) E-Government Adoption and Acceptance: A Literature Review and Research Framework. IN Norris, D (Ed.) *E-Government Research: Policy and Management.* IGI Publishing, Hershey, New York.

United Nations (2008) UN E-Government Survey 2008, From E-Government to Connected Governance. United Nations.

Vennix JAM (1996) *Group model building : facilitating team learning using system dynamics,* J. Wiley, Chichester ; New York

Whitten JL & Bentley LD (2007) *Systems Analysis and Design Methods* McGraw-Hill/Irwin, New York.

Yourdon E (1989) *Modern structured analysis,* Yourdon Press, Englewood Cliffs, N.J. .

Zhang H (2009) An investigation of the relationships between lines of code and defects. *IEEE International Conference on Software Maintenance 2009. ICSM 2009.* Edmonton, Canada.

**Figure 1: E-Government Success Factors – Functional Decomposition**

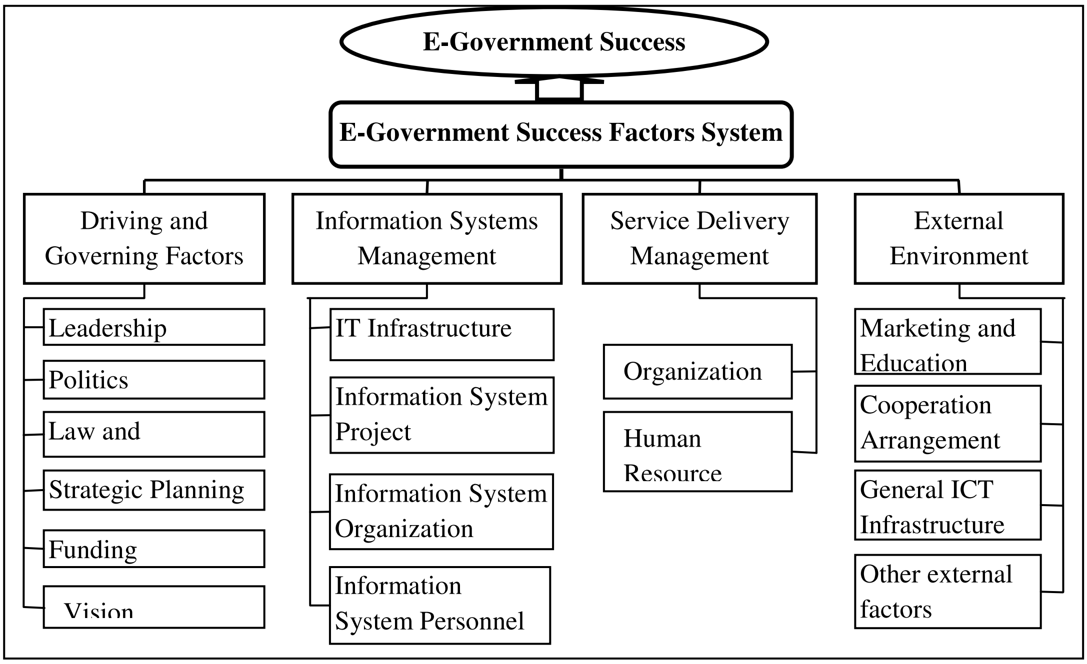


**Figure 2: The E-Government Success Factor Subsystems and Their Relationships**

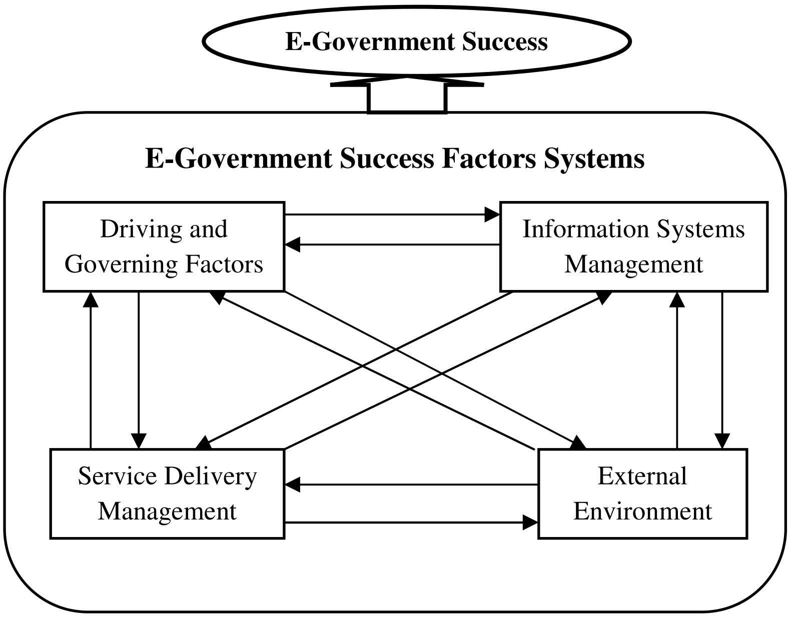

**Figure 3: A System Dynamics Model of Expectancy Theory**

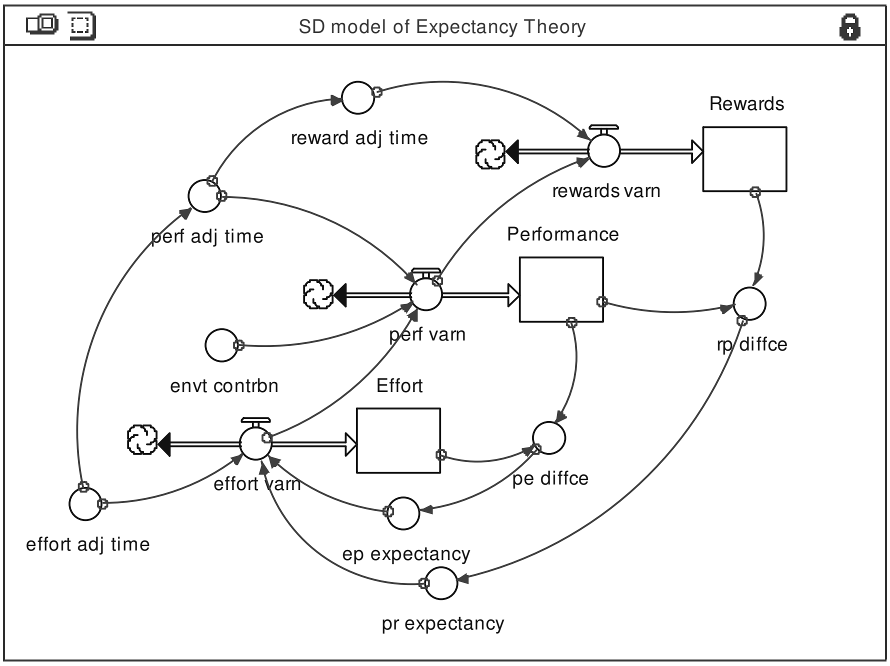

**Figure 4: Detailed Model – System Maintenance**

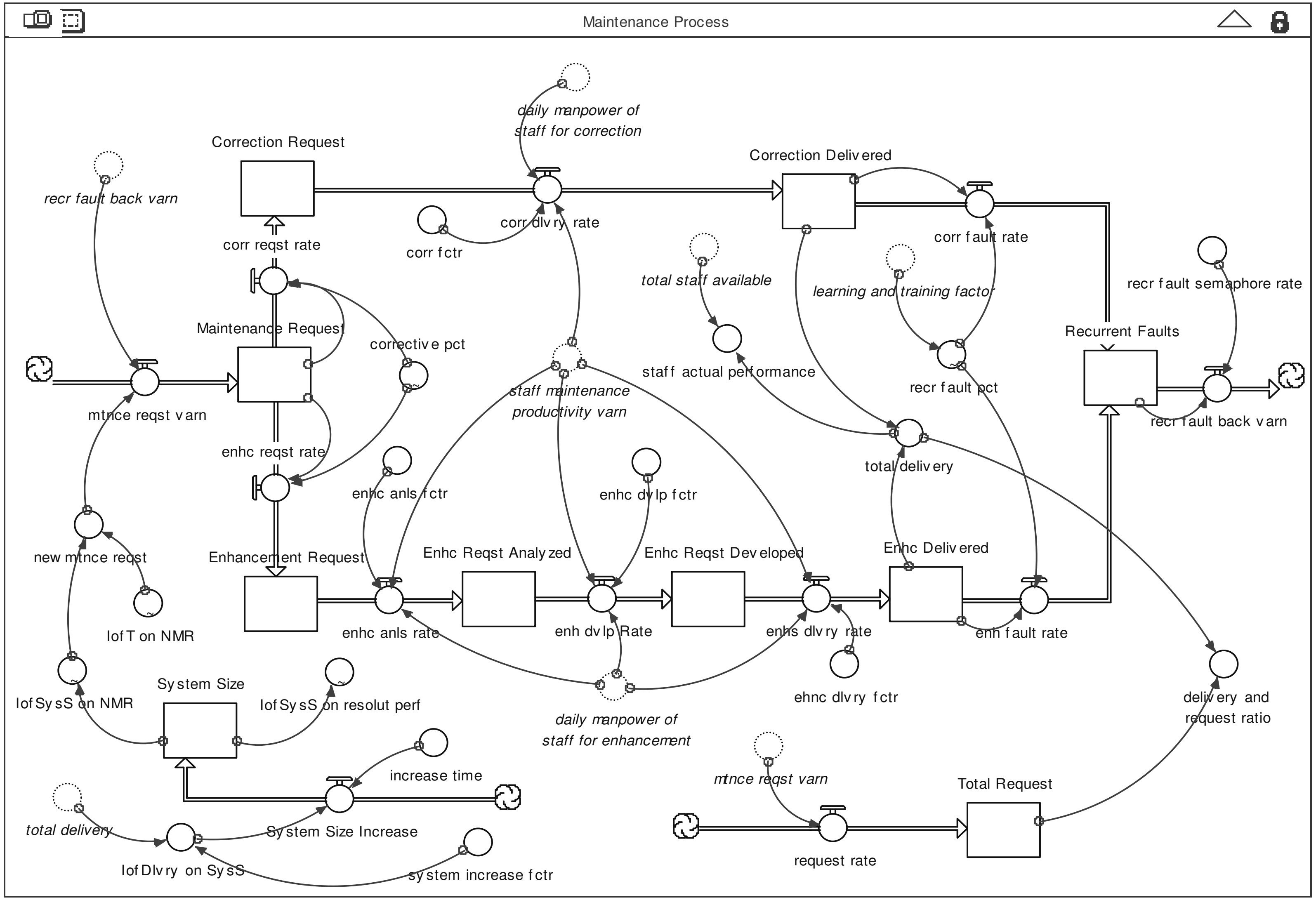

**Figure 5: Relationship between System Size (SLOC) and MRs**

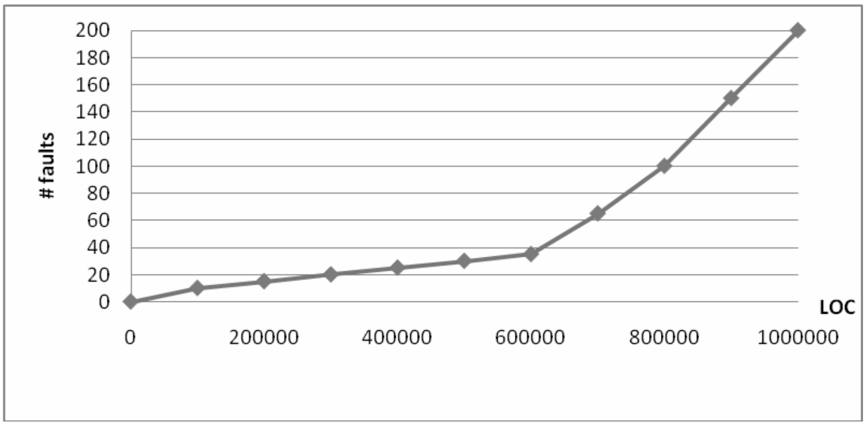


**Figure 6: Core of the Human Resource Sub-model**

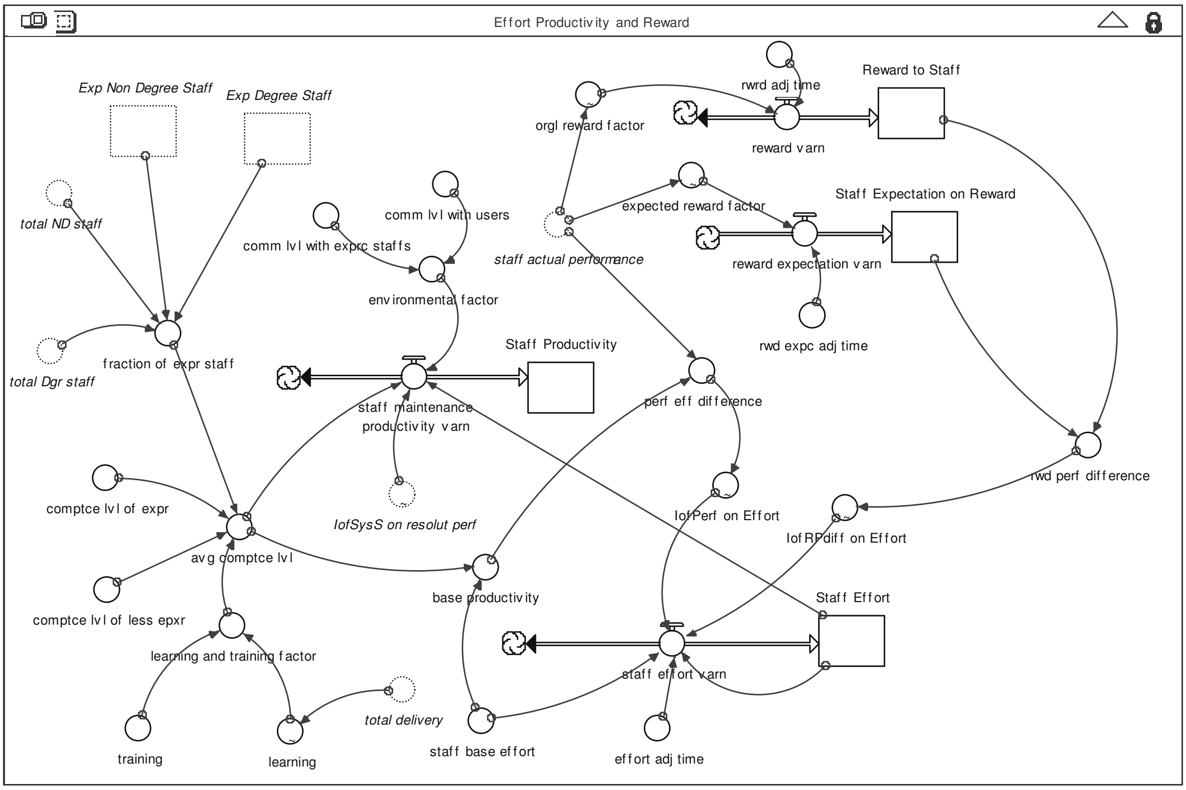

**Figure 7: Performance-Effort Relationship – Used to Instantiate the Converter, *IofPerf on Effort***

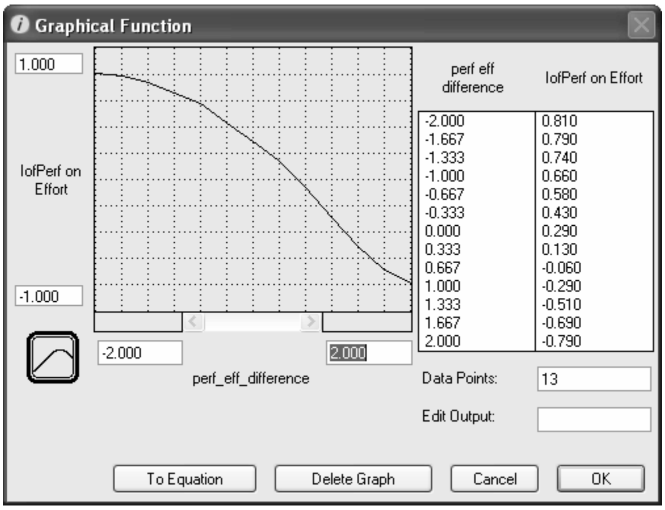


**Figure 8: Maintenance Requests Delivered**

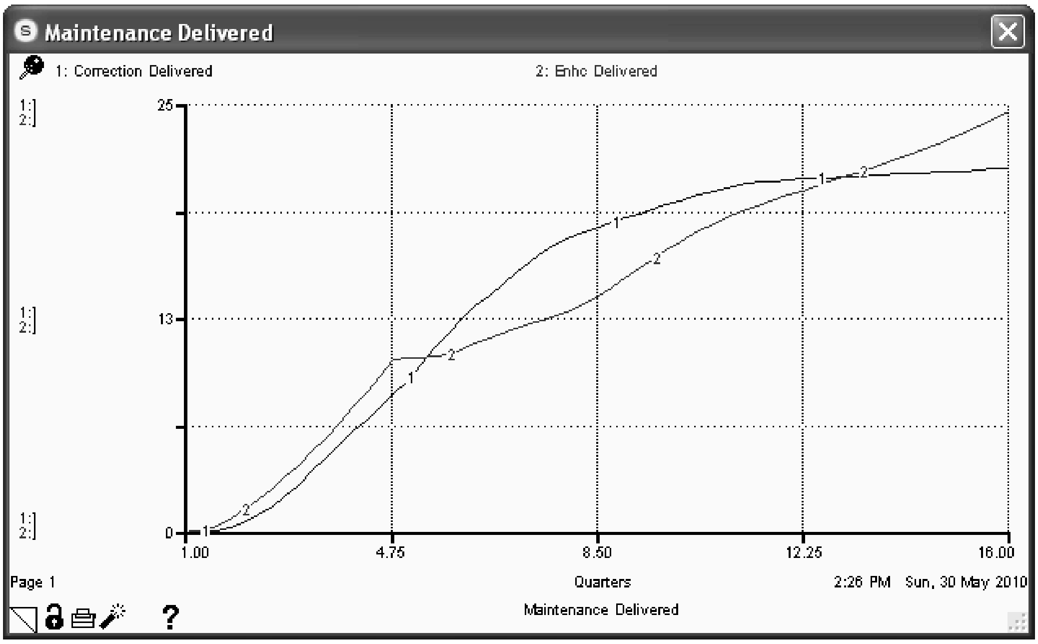

**Figure 9: Model Validation – Actual Versus Simulation Results for Maintenance Requests**