A Model of Feedback Relationships between Software Maintenance and Information Systems Staff Management: A Case of an E-government System

Gunadi*, Geoffrey A. Sandy, and G. Michael McGrath**
* Department of Information Systems, Faculty of Engineering and Informatics, Gajayana University, Malang, Indonesia
** School of Management and Information Systems, Victoria University, Melbourne, Australia

Keywords:
software maintenance,
information systems staff management,
feedback relationships,
causal loop model

ABSTRACT

To ensure systems sustainability in delivering services, software maintenance (SM) is a necessary condition to solve emerging errors and satisfy new requirements during operation. This maintenance involves a great variety of interdependent elements and processes as well as requires competent and motivated staff.This research aims at developing a model capable of explicitly explaining the complexity of causal feedback-relationships between elements and processes of SM and information systems (IS) staff management related factors.

A preliminary causal loop diagram of system dynamics method was developed from literature. A successful e-government system of a Ministry in Indonesia was selected as the case. IS staff and managers of the selected case were deeply interviewed using prepared questionnaires developed based on the preliminary model. Data collected is used to validate and refine the preliminary model.

The resulting empirical causal loop model explicitly describes how various factors, which might not be close in space and time, through chains of causal relationships, influence software availability level over time, hence IS sustainability, and how this level influences back those factors. Therefore, the model helps management with further insight about aligning an important domain of information technology with people to attain system sustainability.

Corresponding Author:
Gunadi,
Department of Information Systems, Faculty of Engineering and Informatics,
Gajayana University, Malang,
Jalan Mertojoyo Blok L, Merjosari, Malang 65144, East Java, Indonesia.
Email: gunadi3374@gmail.com

1. INTRODUCTION

While most of the current knowledge on IS is associated with and dominated by system development, in fact a system is expected to spend most of its lifetime within operation and maintenance stage to deliver services. The importance of the maintenance stage is emphasised by van Vliet [1] as“[c]ompared to software development, software maintenance has more impact on the well-being of an organization”. In addition, literature in IS indicates a large portion of IS budget was allocated for SM [2]. During this stage the emergence of unresolved errors and unsatisfied new users’ requirements can jeopardise the sustainability of a system’s capability in serving users.

In order to achieve the system sustainability an organisation needs to ensure a high software availability level over time (sustainable software) which requires productive and quality SM [3]. Sustainable software can operate consistently to deliver services for a long period, at least as long as its designed lifetime. During this period of sustainability, emerging errors and new functionality requirements can be addressed successfully until it is no longer feasible to maintain the software but rather it must be replaced.

Considering SM process, any SM activity is initiated by a maintenance request (MR) which is caused by software errors and/or new user’s requirements. Conceptually, a maintenance process is similar to a system development life cycle but with much smaller scale [4]. A completed SM task necessarily increases software complexity [5] and any undertaken maintenance might cause recurring maintenance because of possible ripple effects [6]. This, in turn, affects the availability level of the software or systems and triggers further maintenance requests. Therefore, to attain a high level of software availability, it is necessary to understand the existence of feedback relationships between SM elements and activities.

In addition, an organisation cannot ignore the role of IS staff who undertake the maintenance. Managing their performance necessarily determines SM completion and quality levels [7]. In achieving SM success, factors associated with staff motivation are among the most important issues [3, 4, 8]. SM has been associated with being a second-class job compared with software development; and the maintainers are usually within their early careers in information technology [9]. In a more general context of IS management, critical roles of internal IS human resource have been emphasised by many IS literature, for example, by Acuna, Juristo and Moreno [10], and Madachy [11].

Previous research has paid much attention on motivation and performance of human resources [12], especially IS staff [13, 14]. Within the SM context, performance of IS staff might be measured by their productivity in completing SM and quality of the resulting SM. According to the expectancy theory [12], IS staff performance relates dynamically to effort, expected rewards and rewards in feedback fashion [13].On the other hand, the performance in SM is also affected by software complexity [15].In addition, the IS staff performance influences or is influenced by their competence [11], affects or is affected by their experience [16]. Environment surrounding IS staff also has an impact on their performance [13]. Organisational rewards which are normally determined by staff performance, to some degree, relate to absenteeism [17]. In information technology area, competence level naturally degrades over time unless continuous improvement through appropriate training is undertaken [18].

These previous studies indicate the existence of complex factors and feedback relationships affecting the availability level of software over time. Little has been done in previous research that visually models this complexity and accommodates dynamic feedback relationships between those factors. It is important to understand how elements and processes of software maintenance relate to each other and relate to IS staff management factors in order to assist SM management achieving successful SM.

2. RESEARCH METHOD

2.1. Qualitative system dynamics

This research implements qualitative system dynamics (SD) approach [19-21], which is represented visually as a causal loop diagram (CLD), to model the relationships between elements and processes of SM and IS staff management related factors. The approach is chosen because the resulting model is capable of revealing the chain of feedback relationships between variables forming causal loops; therefore indicating the way two or more variables relate to each other which they might not be close in time and distance and providing comprehension of how the effect variable in turn influences the cause variable. Therefore, the CLD can generate insight into the dynamic structure of the system [11, 22]. The method is a particularly appropriate modelling approach where time and feedback loops are important, and where considerable complexity, ambiguity and uncertainty exist [23].

Technically, a CLD consists of words or phrases which are linked by curved arrows, each of which has attached polarity and time delay symbols [22, 24]. The arrow represents a causal relationship between two factors. The polarity is symbolised by '+' indicating the two related variables change in the same direction, or '-' showing the two linked variables vary in two different directions; and the time delay is shown by '//' crossing the arrow. A CLD is formulated by referring to the relevant theories and previous research associated with the formulated problem as well as the researcher's mental model associated with the research problem [22].

2.2. The Case

A unit of a Ministry of the Republic of Indonesia was selected as the Case. The selection is based on the following reasons. The unit plans, develops, operates, and maintains information systems of the Ministry which include software, hardware, people, data, and computer networks, to support the attainment of the Ministry's mission. The unit won e-Government Award in several consecutive years. Currently, it runs a number of application systems and was supported by 16 staff members who have skill and knowledge in computer programming, database management, computer network installation operation and maintenance. Website of the Ministry and most of the application systems currently being operated were developed in-house. The unit runs and maintains successfully many of the application systems to serve and support organisational needs in central office and in other units all over the country for more than five years.

2.3. Data collection, model development and validation

A CLD was constructed as a preliminary model based on previous research and researcher's mental model [25]. Data collected from the Case, which reflects a real world, is used to validate the preliminary model.

Development of the preliminary model from previous research was started by identifying the dynamic behaviour of system software sustainability, and introducing endogenous factors from within SM and IS staff management domains that cause this dynamic. Included within the identification process is the time delay taken by a factor in influencing other factor(s). The identified factor 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.

Data was collected through in depth interviews guided by the preliminary CLD and structured open-ended questionnaires. This CLD along with the relevant existing studies were used to prepare the questionnaires.

The interviewees consisted of a senior manager, an application system manager, and nine IS staff of the unit. The senior manager was asked to evaluate the preliminary model if the factors and dynamic feedback-relationships reflect reality. The model was presented in a stage-wise fashion starting from the simplest model drawn on a single paper to the most complex model on a last single paper. The model is revised accordingly based on the senior manager views if the manager does not confirm any factors and relationships in the preliminary model. In addition, the senior manager was interviewed using a questionnaire focusing on managerial aspects of SM and IS staff. The application system manager was interviewed by using the second questionnaire addressing operational matter of SM and IS staff management under his responsibility. The IS staff was interviewed by using the third questionnaire in order to elicit their actual practice in doing SM, effort, competence, and rewards received.

Findings from the interviews with the senior manager were used to revise and improve the preliminary model, while those elicited from the application systems manager and IS staff were used to corroborate the ones obtained from the senior manager. Further clarification from the interviewees would be sought if discrepancies had been found during interviews.

3. RESULTS AND ANALYSIS

Data collected from the Case indicates that the availability levels of the system software have been very high for a number of consecutive years which can be described as in Figure 1. Management and staff of the Unit have been able to maintain and sustain the software they operate.

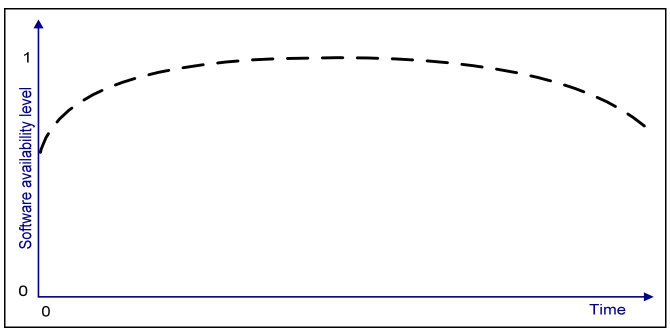

Figure 1. Behaviour over time of the Case system software.

Figure 2 is the resulting CLD showing the model of feedback-relationships of elements and processes of the SM and IS staff management related factors of the e-government system of the Case. The model might be decomposed into a maintenance process, motivational factors, and competence development subsets.

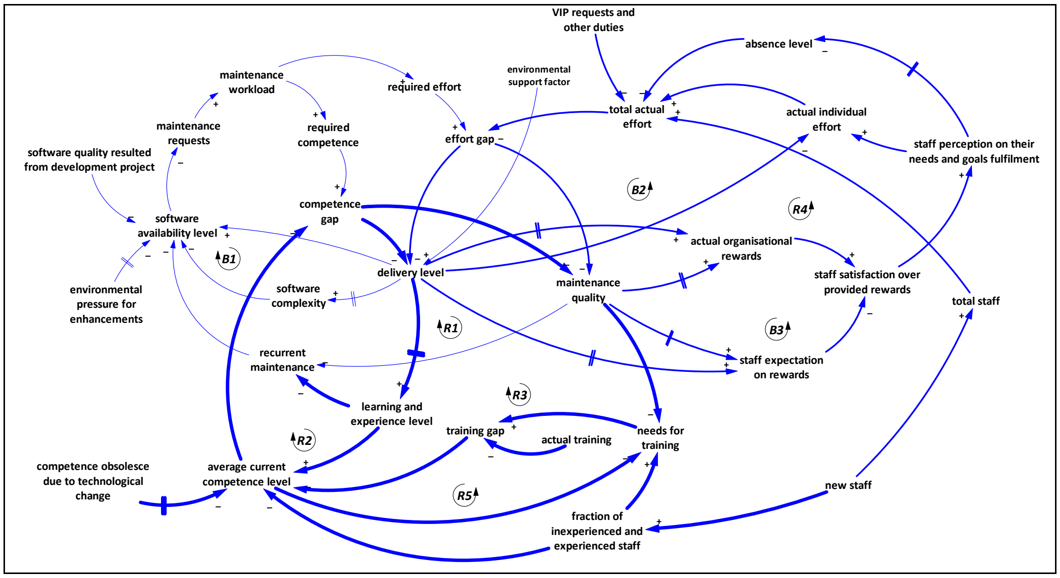

Figure 2. A CLD of feedback relationships between SM and IS staff management of the Case.

Maintenance process. Causal feedback relationships of this subset of the CLD in Figure 2 involve some key factors: software availability level, maintenance requests, required effort and competence, effort gap and competence gap, delivery level and maintenance quality, software complexity and recurrent maintenance. This subset also includes factors which are not part of the causal loop: software quality resulting from development project, environmental pressure for enhancements, and environmental support factor.

System software sustainability can be indicated by a high software availability level over time. This dynamic level is influenced by three factors. First, it is positively influenced by software quality resulting from the development project. Second, it is also affected by environmental pressure for enhancements but in a negative direction. Third, the level of software availability is also negatively influenced by the maintenance itself, through the increasing software complexity and recurrent maintenance requests as well as the maintenance delivery level. In turn, the software availability level will negatively influence the MRs. The maintenance requests occur randomly and a large number of maintenance requests mean 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.

Solving the maintenance workloads need two things simultaneously: staff effort and competence. These two represent two crucial dynamic factors: motivation and ability of the SM staff. However, as the levels of these factors are dynamic, there are always possible gaps between the required and actual levels. These gaps affect completed maintenance delivery level and quality. The delivery level will also be positively affected by environmental support factors, such as smoothness of communication with other competent colleagues and users to obtain their support. Over time, as the number of completed maintenance increases, the software complexity also increases. In turn, this software complexity negatively affects the software availability level forming a balancing closed-loop of dynamics feedback relationships B1. Hence, this results in reduction of the software availability. Much research in software engineering has shown that complex software, especially the one with long SLOC, has higher probability of software error occurrence than the less one.

On the other hand, the quality level of maintenance has a negative effect on recurrent maintenance. This recurrent maintenance especially takes place when a fault caused by delivered maintenance becomes evident at some point of time in the future (ripple effect). It also happens when the requesters repeat their request due to cancellation, incomplete delivery or rescheduling of maintenance. The more recurrent maintenance occurs the lower the software availability. This forms a reinforcing loop R1.

Observing this subset of the CLD, SM management could comprehend the importance of motivation and competence of the IS staff in ensuring the software availability to deliver services. The relationships between these factors are masked by chain of causal relationships.

Motivational factors. Another subset of the CLD in Figure 2 presents feedback-relationships between factors related to IS staff motivation. This subset is a representation and implementation of the expectancy theory and other related factors in an e-government SM context which involves: maintenance quality and delivery level, actual organisational rewards and staff expectation on rewards, staff satisfaction over provided rewards, staff perception of their needs and goals fulfilment, absence level and actual individual effort, effort gap, and total actual effort. This subset also includes total staff and VIP requests and other duty factors which are not involved in a causal loop.

The delivery level and maintenance quality factors, which represent maintenance performance, positively influence both actual organisational rewards and staff expectation of rewards. These influences may take time. 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 on their needs and goal fulfilment. A high level of staff perception on 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, over time, 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. A chain of relationships involving actual organisational rewards factor forms a reinforcing loop R4, while the one containing staff expectation on rewards creates a balancing loop B3.

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, the total actual effort also increases, assuming the total staff is constant, which in turn reduces the effort gap.

The expectancy theory also mentions that staff members 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 members tend to raise their effort – for example, by increasing the number 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 B2 of delivery level → actual individual effort → total actual effort → effort gap → delivery level.

Competence development. In addition to motivation, another subset of the CLD in Figure 2 presents feedback causal relationships of IS staff competence-related factors. In this subset, some key success factors are average current competence level, maintenance quality and delivery level, learning and experience level, needs for training, training gap, fraction of less competent and competent staff. Factors included in the subset which are not part of a causal loop are actual training, new staff, and competence obsolescence due to technological change.

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. That is when the staff is fully motivated and the supporting facilities to perform maintenance are perfect. 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 R2 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.

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. This chain of relationships forms a reinforcing loop R3. On the other hand, the need for training is negatively influenced by the average current competence level.

Additionally, the fractions of less competent and competent staff, which is an exogenous factor, negatively influence the average current competence level and positively affect the need for training. Newly recruited staff determines the fractions.

4. CONCLUSION

This study has developed an empirical CLD, based on an e-government system Case, which explicitly presents the way elements and processes of SM and IS staff management factors, which are not close in space and time, relate to each other influencing the software availability level over time. The model describes how the factors influence the availability level and the level influences back the factors through chains of causal relationships. The CLD facilitates exploration and explanation of the dynamic feedback relationships between factors. For example, the model shows how dynamic rewards eventually influence the software availability level and vice versa. Therefore, the model can assist management to establish fruitful policies in aligning an important domain of information technology with people to attain system sustainability.

5. ACKNOWLEDGMENT

Draft of this paper was proofread by Sri Wahyuni, PhD.

REFERENCES

[1] H. van Vliet. *Software engineering: principles and practice*. Chichester: John Wiley & Sons; 2008.
[2] P. Klint, T. van der Storm and J. Vinju, editors. "DSL Tools: Less Maintenance?", *Preliminary Proceedings of the Tenth Workshop on Language Descriptions Tools and Applications LDTA 2010*; 2010, 27-28 March 2010; Cyprus.
[3] K. Ketler and E. Turban. "Productivity improvements in software maintenance", *International journal of information management*. vol. 12 (1): 70-82. 1992
[4] P. Grubb and A.A. Takang. *Software maintenance: concepts and Practice*, 2nd ed. New Jersey: World Scientific; 2003.
[5] M.M. Lehman and J.F. Ramil. "Software evolution--Background, theory, practice", *Information Processing Letters*. vol. 88 (1-2): 33-44. 2003
[6] S. Black. "The Role of Ripple Effect in Software Evolution", In: Nazim H. Madhavji, Juan C. Fernández-Ramil, Perry DE, editors. *Software Evolution and Feedback: Theory and Practice*. Chichester, UK: John Wiley & Sons, Ltd; 2006. p. 249-68.
[7] X. Junchao, W. Qing, L. Mingshu, Y. Ye, Z. Fan and X. Lizi, editors. "A constraint-driven human resource scheduling method in software development and maintenance process", *Proceeding of the IEEE International Conference on Software Maintenance, 2008 (ICSM 2008)*; 2008, 28 September - 4 October 2008 Beijing.
[8] T.M. Pigoski. *Practical Software Maintenance: Best practices for managing your software investment*. New York: John Wiley & Sons, Inc; 1997.
[9] W.-G. Tan and G.G. Gable. "Attitudes of maintenance personnel towards maintenance work: a comparative analysis", *Journal of Software Maintenance: Research and Practice*. vol. 10 (1): 59-74. 1998
[10] S.T. Acuna, N. Juristo and A.M. Moreno. "Emphasizing human capabilities in software development", *Software, IEEE*. vol. 23 (2): 94-101. 2006
[11] R.J. Madachy. *Software process dynamics*. Hoboken, New Jersey: IEEE / John Wiley & Sons; 2008.
[12] V.H. Vroom. *Work and Motivation*. San Fransisco: Jossey-Bass Inc.; 1995.
[13] G.M. McGrath and E. More, editors. "The expectancy theory of motivation applied to software engineering process modelling", *Proceedings of the IASTED International Conference on Software Engineering* 1998, 28-31 October 1998; Las Vegas, Nevada, USA.
[14] R.H. Rasch and H.L. Tosi. "Factors affecting software developers' performance: an integrated approach", *MIS Quarterly*. vol. 16 (3): 395-413. 1992
[15] Y. Ahn, J. Suh, S. Kim and H. Kim. "The software maintenance project effort estimation model based on functions points", *Journal of Software Maintenance and Evolution: Research and Practice*. vol. 15: 71-85. 2003
[16] S. Narayanan, S. Balasubramanian and J.M. Swaminathan. "A matter of balance: Specialization, task variety, and individual learning in a software maintenance environment", *Management Science* vol. 55 (11): 1861-76. 2009
[17] R.H. Griep, L. Rotenberg, D. Chor, S. Toivanen and P. Landsbergis. "Beyond simple approaches to studying the association between work characteristics and absenteeism: Combining the DCS and ERI models", *Work and Stress*. vol. 24 (2): 179-95. 2010
[18] K. Lee and D. Mirchandani, editors. "Analyzing the dynamics of skill sets for the U.S. information systems workforce using latent growth curve modeling", *Proceedings of the SIGMIS 47th Annual Conference on Computer Personnel Research*; 2009, 28 - 30 May 2009; Limerick, Ireland: ACM.
[19] G. Coyle. "Qualitative and quantitative modelling in system dynamics: some research questions", *System Dynamics Review*. vol. 16 (3): 225-44. 2000
[20] K.E. Maani and R.Y. Cavana. *Introduction to Systems Thinking*. Auckland: Pearson Education New Zealand; 2009.
[21] E.F. Wolstenholme. "Qualitative vs Quantitative Modelling: The Evolving Balance", *The Journal of the Operational Research Society*. vol. 50 (4): 422-8. 1999
[22] J.D. Sterman. *Business Dynamics: System Thinking and Modeling for a Complex World*. Boston: The McGraw-Hill Companies, Inc; 2000.
[23] J.A.M. Vennix. *Group model building : facilitating team learning using system dynamics*. Chichester ; New York J. Wiley; 1996.

[24] J.D.W. Morecroft. *Strategic Modelling and Business Dynamics: A Feedback System Approach*. West Sussex PO19 8SQ, England: John Wiley & Sons, Ltd; 2007.

[25] M. Fryling. "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 ". New York: The University at Albany, State University of New York; 2010.

BIBLIOGRAPHY OF AUTHORS

	Gunadi is a lecturer in Information Systems at the Department of Information Systems, Gajayana University, Malang. He received a PhD degree in Information Systems from Victoria University, Melbourne, Australia. His research interests are e-government, information systems management, information systems human resource management, system dynamics modelling, etc.
	Geoffrey A. Sandy was an Associate Professor of Information Systems, School of Information Systems, Victoria University, Melbourne, Australia.
	G. Michael McGrath is a Professor of Information Systems at the School of Management and Information Systems, Victoria University, Melbourne, Australia.