

Developing an Assessment Model of E-Government Software Assets for Maintenance Recommendations: A Case Study in BKD Bandung

Rizqia Lestika Atimi¹, Bayu Hendradjaya², Wikan Danar Sunindyo³
{rizqia.lestika@gmail.com¹, bayu@informatika.org², wikan@informatika.org³}

State Polytechnic of Ketapang, Jalan Rangge Sentap, Dalong, Sukaharja, Delta Pawan, Kabupaten Ketapang, Kalimantan Barat 78112, Indonesia¹, School of Electrical Engineering and Informatics, Bandung Technology Institute, Jalan Ganesha 10, Bandung 40132, Indonesia^{2,3}

Abstract. This study developed a model of e-Government software asset assessment for its maintenance recommendations. A case study conducted in Badan Kepegawaian Daerah (BKD) Bandung. The steps consist of the identification of software asset management, developing asset assessment, and proposing maintenance recommendations that were carried out in this investigation. The Likert questionnaire and task analysis were used to assess the e-Government software based on two essential aspects regarding the business value and technical value. It is found that the business values and the averages of the technical values for the software of SIM Gaji, MPEG, and SIAP are in high category (the entire average score between 3.00 – 4.00). The recommendations are ordinary maintenance and feasible to use. Thus, the model developed in this study is appropriate to use for assessing e-Government software. To provide comprehensive maintenance recommendations, future studies might be used for additional aspects besides the business and technical value.

Keywords: Assessment model, Business and technical value, Maintenance recommendations, Software evolution

1 Introduction

Nowadays, many organizations are highly depending on the software system for managing their business. The systems are critical and considered as an important thing. The organization has to invest a large amount of money not only for the availability of the software but also for maintaining and changing the value of the software [1]. The efforts comply with Lehman's first law that stated system maintenance is an inevitable process. A program or software used in the organization will unwell operated anymore in the future time. Therefore, to achieve the longlasting usefulness of the software, their maintenance should be done by the organization. According to INPRES No.3 Tahun 2003, the software should be used or developed by the government and their periodical maintenance as well [2].

Software maintenance is the process of changing a system after it has been delivered to the user [1, 3]. Maintenance aims to extend the life of assets, ensure the availability of equipment installed for production or services and get the return value of the investment, ensure the operational readiness of all equipments needed in an emergency, and ensure the

safety of people who use the facility [4]. The changes made to the software may be a simple change to correct coding errors, more extensive changes to correct design errors or significant enhancements to correct specification errors or triggered to accommodate new requirements [5].

In the context of software maintenance, the different approaches have been proposed for determining the most appropriate recommendation for the system software. Ramson [6], for instance, using three measurement aspect i.e., technical quality, business value, and organization characteristic. De Lucia [7], Cimitile [8], and Alkazemi [9] used two measurement aspect i.e. business value and technical value. This study will asses the software by using two essential aspects of measurement, i.e. business aspect and technical aspect. After assessing the software, there are some maintenance recommendations launched. The term "ordinary maintenance, adaptive maintenance, and elimination" is proposed by De Lucia [7]. Cimitile [8] introduced the terms "extraordinary maintenance and reengineering", and Alkazemi [9] used the term "re-architecting and replacement".

Although various software assessing techniques for its maintenance recommendations have been conducted for business organizations, the relevant study is far from investigated for the government institutions, especially in Badan Kepegawaian Daerah (BKD) Bandung. This organization had not had a model and instruments for assessing its e-Government software assets. It is not surprising that this condition might affect inefficiency in the utilization of government funds and lack of software sustainability. This study focuses on developing a model of e-government software asset assessment for maintenance recommendations in Badan Kepegawaian Daerah Bandung.

2 Research Method

In developing the model, there are four stages was carried out. *First*, determine factors that affect the business values and the technical values by the characteristic of local government organizations and e-Government software. In this stage, observation, interview, and kinds of literature study were used for determining the factors. *Second*, analyze indicators to each factor based on the literature review using metrics mapping. The instruments were used are the Likert questionnaire that raised using Goal Question Metrics (GQM) method for measuring the business values and task analysis as performing as black-box testing for measuring the technical values. The scoring of each assessment using score range, following; (1) score 4, means the software very effective; (2) score 3, means the software effective; (3) score 2, means the software less effective; (4) score 1, means the software ineffective. And then the assessment result of business value and technical value were classified into two qualification categories, high and low; high category, if average score 3,00 – 4,00 and low category, if 1,00 – 2,99. *Third*, classifying the software maintenance recommendation based on their terminology using metrics mapping. The recommendation refers to the *Quadrant Mapping* model using categories of the business values and the technical values, high and low. *Fourth*, develop the model with integrating software asset management and software legacy concepts. The model consists of three steps i.e., *the identification of software asset management, developing software of asset assessment, and proposing maintenance recommendations*. To validate the developed model, a case study employed in Badan Kepegawaian Daerah (BKD) Bandung i.e. for the software asset of SIM Gaji, MPEG, and SIAP.

3 Results

An assessing model of e-Government software assets for maintenance recommendation developed in this research is depicted in Figure 1.

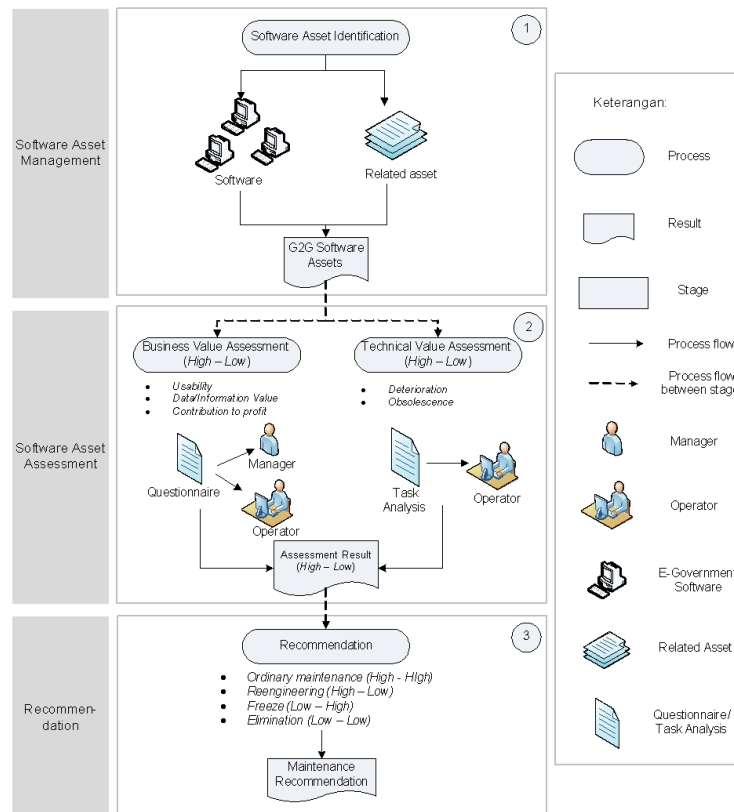


Fig. 1. An Assessment Model of Software Assets for Maintenance Recommendation.

The operationalization of the model is described below.

1. Stage of Software Asset Management

At this stage, information on the e-Government software asset generated as well as other things related to the software. Assets of e-Government software are identified and then classified based on the type of service, they are Government to Government (G2G), Government to Citizen (G2C) and Government to Business (G2B). The assessment asset of e-Government software in this study only focuses on G2G services. A detail description of the basic information attributes of software assets can be shown in Table 1.

Table 1. Basic Information Attributes of Software Asset.

No.	Information Attributes	Description
1	Software Name	Information the name of the software
2	Owner/Custodian of Software	Information the owner/custodian of software
3	Location	Information the location of software used
4	Status	Information the status of software usage
5	Version	Information the versions of the software
6	Vendor	Information the developers of software
7	Documentation	Information the documentation support owned by software
8	Information	Information activities carried out against software

2. Stage of Software Assessment

The set of the process at this stage generates the assessment of e-Government software asset. The assessment result determines the type of software maintenance recommendation.

This assessment was based on two essential aspects i.e., business value and technical value. Assessment of each aspect validated by using metrics mapping that refers to ISO/IEC 9126-4 standard [10] and other relevant literature supports [3, 5, 6, 11, 12]. Moreover, Likert's questionnaire, which is intended for stakeholders (managers) and operator officers in the institution, was used to assess the business value of e-Government software assets. The questionnaire is raised by using the GQM (*Goal Question Metric*).

To assess business value, three variables must be measured i.e., *utility*, *data/information value*, and *contribution to profit*.

1. *Utility*: indicate the importance of using the e-Government software asset in supporting the organization's business process.
2. *Data/information value*: indicate the relevance of data or information that managed by e-Government software asset for organization business process.
3. *Contribution to profit*: indicate beneficial effects earned by individuals as well and the organization as a whole through the utilization of e-Government software assets.

A detail description of variables and their indicators of the business value aspect can be shown in Table 2.

And to assess the technical value, the measurement of the software performed as black-box testing to see the functionality of the software. Two variables must be measured i.e., *deterioration* and *obsolescence*.

1. *Deterioration*: represents the aging of an e-Government software asset caused by the failure to meet changing requirements.
2. *Obsolescence*: represents the aging of an e-Government software asset as a result of continuing changes that are made.

A detail description of variables and their indicators of technical value aspect can be shown in Table 3.

Table 2. Variables and Indicators of Business Value.

N o.	Variables	Indicators	Purpose	Proposed Metrics
1	Utility	Effectivity	Assessing the effectiveness of software duty and support the achievement of organization goals.	Task Completion, Error Frequency, Subjective valuation from the manager
		Efficiency	Assess the efficiency of the software in performing work tasks.	Task Time, Response Time, Subjective valuation from a manager perspective
		Frequency of use	Assess the frequency of the use of the application in support of the work task.	Frequency of use, Subjective valuation from an operator perspective
2	Data/ Information Value	Accurate	Assessing the accuracy of data/information that is managed by the software.	Accuracy of data/information
		Completeness	Assess the completeness of the data/information that is managed by the application.	Completeness of data/information
		Timeliness	Assess the timeliness of updates data/information that is managed by the software.	Timeliness of data/information
		Integration	Assess the needs of data/information that is managed by the software, both internally and externally.	Integration of data/information
		Dependency	Assessing the organization's dependence on the data/information that is managed by the software, both internally and externally.	The dependency of data/information
3	Contribution to profit	Benefit impact: a. Productivity b. Effectivity c. Efficiency	Assessing the impact of the software implementation, either by individuals as well or organizations as a whole to see the improvement of productivity, effectiveness, and efficiency of work.	Subjective valuation of productivity, effectivity, and efficiency from the manager and operator perspective.

Table 3. Variables and Indicators of Technical Value.

N o.	Variables	Indicators	Purpose	Proposed Metrics
1	Deterioration	Error frequency	Measuring the frequency of occurrence of software errors.	Error frequency
		Task completion	Measuring tasks that can be completed correctly by the software.	Task completion
		Respon time	Measuring how long the response time required in response action by the software.	Respon time
2	Obsolescence	Age	Measuring the age of the software.	Age of software
		Documentation	Knowing the supporting documentation which is owned by the software.	Documentation Support
		Programming language	Knowing the programming language used to build software.	Generation of Programming language

Assessment of the technical value of e-Government software assets validated by using observation method and task analysis which is intended for stakeholders (operators) in local government organizations. Both assessment results of the business value and technical value were classified into two qualifications i.e., *high* and *low*.

3. Stage of Recommendation

This stage aims to produce the appropriate maintenance recommendations of e-Government software assets based on the result of the assessment from the previous process. The maintenance recommendations of e-Government software assets are determined by integrating the result of the valuation of the business value and technical value as shown in Table 4.

Table 4. Maintenance Recommendation Metric of e-Government Software Asset.

Business Value	Technical Value	Maintenance Recommendation	The Sustainability
High	High	<i>Ordinary maintenance</i>	Worthy to use
High	Low	<i>Reengineering</i>	Worthy to use
Low	High	<i>Freeze</i>	Worthy to use
Low	Low	<i>Elimination</i>	Can't be used

A case study was applied to implement the model at *Badan Kepegawaian Daerah* (BKD) in Bandung. The assessing results of three assets (*SIM Gaji*, *MPEG*, and *SIAP*) on business value and technical values were shown in Table 5 and Table 6.

Table 5. The Assessing Results of Business Value.

Variables	Indicators	Asset Score		
		SIM Gaji	MPEG	SIAP
Utility	Effectivity	4	4	4
		4	4	4
		4	4	4
	Efficiency	2	4	4
		4	4	4
	Frequency of use	4	4	4
Data/Information Value		3	3	3
	Accuration	3	4	3
	Completeness	3	4	4
	Timeliness	4	3	3
		4	4	4
	Integration	3	3	2
		4	4	2
	Dependency	4	4	1
Contribution to profit		4	4	4
	Benefit impact:			
	a. Productivity	3	3	3
		3	3	3
		4	4	4
	b. Effectivity	4	3	3
		4	3	3
	c. Efficiency	4	3	3
		4	4	3
		82	84	76
Average	3.56	3.65	3.30	
Classification	High	High	High	

Based on Table 5 and Table 6, showed that the software *SIM Gaji*, *MPEG*, and *SIAP* are in the high category because of the entire average score of the business value and the technical value between 3.00 - 4.00. Therefore, the maintenance recommendations are ordinary maintenance and the software is feasible to use.

Table 6. The Assessing Results of Technical Value.

Variables	Indicators	Asset Score		
		SIM Gaji	SIMPEG	SIAP
Deterioration	Error frequency	4	4	4
	Task completion	3	4	4
	Respons time	4	4	4
Obsolescence	Age	3	4	4
	Documentation	2	3	3
	Programming language	3	3	3
	Total	19	22	22
	Average	3.17	3.67	3.67
	Qualification	High	High	High

4 Discussion

Software maintenance is the process of changing a system software after delivered [1, 3]. According to Lehman's first law, system maintenance is an inevitable process. Every program or software that is used in the environment needs to change, or become progressively less useful in that environment (software evolution) [1]. The changes made to the software may be a simple change to correct coding errors, more extensive changes to correct design errors or significant enhancements to correct specification errors or triggered to accommodate new requirements [5].

There are some types of software maintenance in different terminology such as, *fault repairs, environmental adaptation, functionality addition, corrective maintenance, adaptive maintenance, perfective maintenance, emergency maintenance, ordinary maintenance, extraordinary maintenance, discard, wrap, freezing, outsource, and reverse engineering* [1, 5] [7, 13]. But, refer to the general understanding of the terminology, an inference can be drawn that software maintenance is about *ordinary maintenance, reengineering, freeze, and elimination*. *Ordinary maintenance* is a modification of a software product to correct discovered faults, to keep a computer program usable in a changed or changing the environment, and the modification to improve performance or maintainability after it was

delivered. *Reengineering* is the process of re-structure a legacy system to a new system or updating software with better functionalities. *Freeze* is doing nothing to the software. And, *elimination* is to remove the software and start again from defining new requirements to develop a new one.

Based on the kinds of literature review, the different approaches for deciding the most appropriate strategy for software maintenance have been proposed [6, 7, 8, 9]. It was found that two essential aspects are used to decide the software maintenance recommendation i.e., business value and technical value. The business value represents the critical of the software for an organization [1, 3, 6, 7, 8, 9]. Also, the technical value represents the condition or system quality included hardware, application software and software support [1, 3, 6, 7, 9].

4.1 Business Value

Sommerville [1] stated four basic issues need to discuss to assess the business value of the software asset. First, *the use of the system*. If systems are only used occasionally or by a small number of people, they may have a low business value. Second, *the business processes that are supported*. A system may have a low business value because it forces the use of inefficient business processes. Third, *the system dependability*. If a system is not dependable and the problems directly affect the business customers or mean that people in the business are diverted from other tasks to solve these problems, the system has a low business value. Fourth, *the system output*. If the successful business depends on these outputs, then the system has a high business value.

According to De Lucia [7], the business value of the software might be expressed by (1) The *economic value* has been included in order to take into account the economic dimension of software value: the greater the contribution to profit and cost savings provided by the asset to its owner, the greater its economic value; (2) The *data value* represents the relevance of information managed by a software asset: if data are essential for business processes survival, their value is critical. (3) The *utility* variable takes into account how effectively software is used and how suitably it supports the organization's business goal achievement, and (4) the *specialization* variable represents the degree of specialization of the software concerning the business domain.

For assessing the business value of a software system, Cimitille [8] stated that the data sources to be considered might include (1) users and managers from the business organization; (2) user documentation, like user guides and similar; and (3) descriptions of the persistent data of the application.

4.2 Technical Value

The term technical value might be expressed by *maintainability* [7], *decomposability* [7] [8], *deterioration* [7, 8], and *obsolescence* [7, 8]. *Maintainability* represents an aggregated measure of various software product attributes such as its complexity, testability, and analyzability. *Decomposability* expresses how easily the main components of a software system are identifiable and independent from each other. The decomposability tries to capture the goodness of the software architecture. *Obsolescence* expresses the aging of a software system caused by the failure to meet changing needs. The *deterioration* represents the aging of

a software system as a result of continuing changes that are made. The continuous progress of hardware/software platforms, programming languages and development practices causes software technology to become outdated in a short time.

For assessing technical value can be performed through three approaches [3], they are:

1. System-level: see the system as BlackBox.
2. Component level: assessment involves looking inside the system and identifying its processing components. This assessment tends to applications and utility programs; including support software programs for tasks such as printing and sorting.
3. Detail (constituent) level: At this level, the software application assessed in terms of program and data schema.

Sources of information that necessary to assess the technical quality may be obtained from the expert staff of experienced maintenance or learn the system artifacts such as source code, documentation or documentation change history of software development.

5 Conclusion

A model of e-Government software asset assessment for its maintenance recommendations developed in this study is appropriate and worthy to use for certain governments. The software asset of SIM Gaji, MPEG, and SIAP are in high category regarding business and technical values. Their recommendations are ordinary maintenance and feasible to use. Most stakeholders and operator officers who actively participated in this study declared that the model is not applying several complicated technical computerized procedures. Due to this study only based on two essential aspects i.e. business value and technical value, for forthcoming studies, the use of another or additional aspects of measurement might be developed.

Acknowledgments.The authors due thanks stakeholders and operator officers in Badan Kepegawaian Daerah (BKD) Kota Bandung for kind helpful and cooperation.

References

- [1] Sommerville, I.: Software Engineering 9th Edition. United States of America: Addison-Wesley (2011)
- [2] Inpres No.3. Retrieved January 28, 2016, from <http://jatim.kemenag.go.id/> (2003)
- [3] Warren, I.: The Renaissance of Legacy System Method Support for Software Evolution, London: Springer, (1999)
- [4] Corder, A.: Teknik Manajemen Pemeliharaan. Jakarta: Erlangga (1996)
- [5] Bennett, K., Ramage, M., & Munro, M.: Decision Model for Legacy System, IEE Proceedings – Software, pp. 153-159 (1999)
- [6] Ransom, J., Sommerville, I., & Warren, I.: A Method for Assessing Legacy Systems for Evolution. Software Maintenance and Reengineering, Proceedings of the Second Euromicro Conference, pp. 128-134 (1998)
- [7] DeLucia, A., Fasolino, A. R., & Pompella, E.: A Decisional Framework for Legacy System Management. IEEE International Conference, pp. 642-651 (2001)
- [8] Cimitile, A., Fasolino, A. R., & Lanubile, F.: Legacy System Assessment to Support Decision Making (2001)

- [9] Alkazemi, B. Y., Nour, M. K., & Meelud, A. Q.: Towards a Framework to Assess Legacy Systems. 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 924-928 (2013)
- [10] ISO/IEC. ISO/IEC 9126-4: Software Engineering - Software product quality. Geneva.
- [11] Iribarren, M., Concha, G., Valdes, G., Solar, M., Villaruel, M. T., Gutiérrez, P., et al.: Capability Maturity Framework for e-Government: A Multi-dimensional Model and Assessing Tool (2008)
- [12] De Lone, W. H., & McLean, E. R.: The DeLone and McLean Model of Information Systems Success: A Ten-Year Update, Journal of Management Information Systems, pp. 9-30 (2003)
- [13] IEEE: Standard for Software Maintenance, USA: Institute of Electrical and Electronics Engineers, Inc. (1998)