VOL 1 (2017) NO 4 - 2

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Initial Review on ICTS Governance for Software Anti-Aging

Mohamad Khairudin Morshidi[#], Rozi Nor Haizan Binti Nor[#], Hairulnizam Mahdin[*]

[#] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia*
[*] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn, Johor, Malaysia*

E-mail: mohamadkhairudin1994@gmail.com, rozinor@upm.edu.my, hairuln@uthm.edu.my

Abstract— **For the past 20 years various researches regarding software aging have been conducted. Software aging is the situation in which the accumulation of errors occurring in operational software system that has run for a long time that may lead to performance degradation, resource depletion and eventually causing the software to crash or hang [1]. David Parnas divided software aging into two categories: 1) the failure of the software to adapt with environment that is dynamic and 2) the result of the changes itself [2]. Factors that can affects software aging can be classified into several categories: 1) functional, 2) human, 3) product and 4) environment [3]. In general, the factors that affect software aging can be divided into internal and external factors. The main objectives of this paper are to briefly describe the definition of software aging and also ICTS governance. In addition, to that, this paper also compiles the software aging factors that are being investigated by previous researchers. The need for future research regarding ICTS governance and Software aging also determined at the end of this paper.**

Keywords— **governance, ICTS, Anti-aging.**

I. Introduction

For the past 20 years various researches regarding software aging have been conducted. Software aging is the situation in which the accumulation of errors occurring in operational software system that has run for a long time that may lead to performance degradation, resource depletion and eventually causing the software to crash or hang [1]. David Parnas divided software aging into two categories: 1) the failure of the software to adapt with environment that is dynamic and 2) the result of the changes itself [2].

Factors that can affects software aging can be classified into several categories: 1) functional, 2) human, 3) product and 4) environment [3]. In general, the factors that affect software aging can be divided into internal and external factors.

The main objectives of this paper are to briefly describe the definition of software aging and also ICTS governance. In addition to that, this paper also compiles the software aging factors that are being investigated by previous researchers. The need for future research regarding ICTS governance and Software aging also determined at the end of this paper.

II. Research Background

Software aging can be defined as the progressive performance degradation or the state of the software degrades in time [3]. While internal factors of the software that lead to aging has been investigated extensively, there are not much attention given to the human factors that may lead to software aging [2]. This section will briefly define ICTS Governance that been used or defined by previous researchers and describe the definition of software aging.

A. ICTS Governance.

There are many definitions that defined ICT governance but all are highlighting the idea that a business must be govern by a well-organized activity carried out by professional people who will take full responsibilities for every action they take [4]. ICT play important role in the information era that we live on this day. This is proved by the fact that, the largest components on capital expenses of most company for the last decades have been devoted to ICT [1]. Besides, based on the data gathered from Department of Statistics Malaysia's Official Portal, it shows that in Malaysia, ICT contribution to economy increase 17.8% in 2015.

In addition, Kendall, 2006 stated that the role of ICT is considered critical in helping promoting the development of sustainable development effort in developing countries [5]. This proves ICT play a major part in the world we are living now. ICT can be divided into five major domains: 1) IT principles, 2) IT architecture, 3) IT infrastructure, 4) Business application and 5) Investment decision [6].

If governance is considered as a whole, ICT governance is a subset of a very important subset inside that whole [4]. There are 8 component of ICTS governance listed by [7]. Within 8 of these factors, there are 12 quality factors that may influence the quality of ICTS governance. All the components and quality factors in regard for ICTS governance that are listed by [7] are shown in Table 1.

TABLE 1
ICTS GOVERNANCE COMPONENT AND QUALITY FACTORS BY[7].

ICTS Governance Component	Quality Factors
ICTS Organization Structure	Structure
	Role
	Position
ICTS Policy	Policy implementation and enforcement
ICTS Quality Culture	Quality culture
ICTS Delivery	Marketing implementation
ICT Resource Management	Staff competencies
ICT Finance Management	Finance allocation efficiency
ICT Outsource Management	Outsource implementation
ICTS Plans	Contingency planning
	Capacity planning
	Planning for new ICTS development

With the high investment in ICTS, organizations are expecting the return on the investment. Thus in order to meet the high expectation of ICTS, every aspect of ICTS governance needs to be taken into consideration. Therefore, incompetence in the governance of ICT needed to be tackled and monitored efficiently. The inefficiency in ICT governance will affect the organization directly. This research will develop a measurement model that can be used to measure the effectiveness of the plan in preventing software aging.

B. Software Aging Factors.

Software aging is the situation in which the accumulation of errors occurring in operational software system that has run for a long time that may lead to performance degradation, resource depletion and eventually causing the software to crash or hang [1].

Ghoneim, 2003 stated that it will be more cost effective in preventive maintenance of software aging rather than repair actions when the software has aged [8]. Taking that into consideration, it is very important to identify each and every one of the symptoms and factors of software aging. The factors gathered through this research are compiled in the results section.

III. METHODOLOGY

This paper will mainly focus on defining the definition of software aging and ICTS Governance by previous researchers. This paper will implement the methodology proposed by [9] which comprised of three major steps. Figure 1 below shows the graphical representation of the methodology applied for this research.

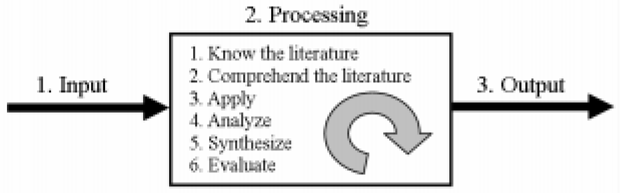

Fig 1. The three stages of effective literature review process by [9]

The first step in this methodology is input. Input is the process of finding the previous research that may be related to achieving the objective of this paper. After completing the choosing of papers, then start the second phase for the research which is processing all the papers. The main objective is to gathered as much factors of software aging that been highlighted by previous researchers as possible.

Finally, all the data gathered are compiled in this paper and based on the data gathered, a decision on whether a further research on this topic are needed or not are determined.

IV. RESULTS

Based on the study of the previous research, the definition of ICTS Governance and software aging are described in the research background earlier. For this section, list of factors on software aging highlighted by previous researchers are compiled

TABLE 2
FACTORS OF SOFTWARE AGING HIGHLIGHTED BY PREVIOUS RESEARCHERS

Author(Year)	Factors				
	Technology	Functionality	Cost	Human	Environment
Zuriani et.al (2006)[10]		•Time responsiveness •Software are unable to meet the user's needs •Failure to functions us user's intended •Software is no longer relevant •High frequency of software error •Failure to get support •User interface		• Lack of expertise in upgrading and maintaining software • Weak software quality practices among practicioner • Inefficient software management by management team • Software in not user friendly	• Dynamics to environment changes • Lack of cost for software maintenance • Software not compatible with current hardware technology • Hardware changes • Business changes • Business need • Software technology changes

Author(Year)	Factors				
	Technology	Functionality	Cost	Human	Environment
Yujuan et.al (2005)[11]	- Resource leak				
Castelli et. Al (2001)[12]	- Exhaustion of system resources - Data corruption - Numerical error accumulation				
Edward (1984)[13]		- Complex software system			
Ghonem and Fahmy (2003)[8]		- Memory bloating or leaking - Unreleased file locks - Data corruption - Numerical error accumulation			
Parnas(1994)[2]		- Failure of the product to meet the owner needs		- Failure of owner to modify the products	- Result of changes in the products
Jamaiyah et. al (2012)[14]	- Technology challenges - Declining quality	- Requirement evolution - Business compatibility and stability - Increasing complexity	High Cost	- Failure	- Environment dynamics - Competition - Ontology change

The factors in Table 2 are compiled in 5 categories which is technology, functionality, cost, human and environment. These categories are based on the research conducted by [], and concluded that aging factors can be classified into 5 categories. In [8], stated that there are three symptoms of software aging: 1) hard to keep up with the market, 2) aging software degrades in its performance due to gradually deteriorating structure and 3) software is buggy when changes are made. All the decision that made by the governance personnel contribute a very great impact on the definition of the software aging itself [7]. The highest contributor to software aging is software error (40%), followed by human errors (40%) and hardware error (15%), finally other (5%) [9]. Data has shown that human errors contribute to software failure just as much software error. Thus a further study on the human factors needed to be conducted.

Conclusion

This paper has presented the initial work that related to ICTS governance and software aging. The lack in papers that focus on external factors that affect software aging has become the main motivation to further investigate and explore the relation between ICTS governance and software aging. Based on both the factors of software aging and the type of ICTS plan that are listed by previous researchers, a more extensive research needed to be done in order to find the relation between both factors.

References

[1] Cotroneo, D., Natella, R., Pietrantuono, R., & Russo, S. (2011). Software aging and rejuvenation: Where we are and where we are going. In Proceedings - 2011 3rd International Workshop on Software Aging and Rejuvenation, WoSAR 2011. https://doi.org/10.1109/WoSAR.2011.15

[2] Parnas, D. L. (1994). Software aging. In Proceedings of 16th International Conference on Software Engineering (pp. 279–287). https://doi.org/10.1109/ICSE.1994.296790

[3] Yahaya, J. H. A. D. ; Z. H. A. (2016). Evergreen Software Preservation : The Anti-Ageing Model, 1–6. https://doi.org/http://dx.doi.org/10.1145/2896387.2896436

[4] Kan, B. A. H. G. R. (2004). IT Governance and Corporate Governance at ING. ISACA Journal, 2.

[5] Kendall, K. E., Kendall, J. E., & Kah, M. M. O. (2006). Formulating information and communication technology (ICT) policy through discourse: How Internet discussions shape policies on ICTs for developing countries. Information Technology for Development, 12(1), 25–43. https://doi.org/10.1002/itdj.20019

[6] Weill, P., & Ross, J. (2005). A Matrixed Approach to Designing IT Governance. In MIT Sloan Management Review (Vol. 46, pp. 26–34). https://doi.org/10.1177/0275074007310556

[7] Rozi Nor Haizan Binti Nor. (2013). ICT Service Quality Measurement Framework for Malaysian Universities Context. Universiti Teknologi Malaysia.

[8] Ghoneim S.A., F. H. M. a, Ghoneim, S. a, & Fahmy, H. M. a. (2003). Evaluation of the DRM and the time for preventive maintenance for aging software. Software Quality Journal, 11(1), 57–75. https://doi.org/http://dx.doi.org/10.1023/A:1023636112886

[9] Levy, Y., & Ellis, T. J. (2006). A systems approach to conduct an effective literature review in support of information systems research. Informing Science, 9, 181–211. https://doi.org/10.1049/cp.2009.0961

[10] Abdullah, Z. H., Yahaya, J., & Deraman, A. (2015). Towards anti-Ageing model for the evergreen software system. Proceedings - 5th International Conference on Electrical Engineering and Informatics: Bridging the Knowledge between Academic, Industry, and Community, ICEEI 2015, 388–393. https://doi.org/10.1109/ICEEI.2015.7352532

[11] Bao, Y., Sun, X., & Trivedi, K. S. (2005). A workload-based analysis of software aging, and rejuvenation. IEEE Transactions on Reliability, 54(3), 541–548. https://doi.org/10.1109/TR.2005.853442

[12] Castelli, V., Harper, R. E., Heidelberger, P., Hunter, S. W., Trivedi, K. S., Vaidyanathan, K., & Zeggert, W. P. (2001). Proactive management of software aging. IBM Journal of Research and Development, 45(2), 311–332. https://doi.org/10.1147/rd.452.0311

[13] Adams, E. N. (1984). Optimizing Preventive Service of Software Products. IBM Journal of Research and Development, 28(January), 2–14. https://doi.org/10.1147/rd.281.0002

[14] Jamaiah Yahaya, & Aziz Deraman. (2012). Towards the Anti-Ageing Model for Application Software. Proceedings of the World Congress on Engineering 2012, II, 5.