Sci.Int.(Lahore),26(4),1615-1618,2014 ISSN 1013-5316; CODEN: SINTE 8

THE ANTI-AGEING FACTORS FOR EVERGREEN SOFTWARE – A PRELIMINARY STUDY

Zuriani Hayati Abdullah[1], Jamaiah H. Yahaya[2], Aziz Deraman[3]
[1,2]Faculty of Information Science and Technology,
University Kebangsaan Malaysia,Bangi, 43650 Selangor, Malaysia
[3]School of Informatics & Applied Mathematics, Universiti Malaysia Terengganu,
Kuala Terengganu, Terengganu, Malaysia
zha.ukm@gmail.com, jhy@ukm.edu.my, a.d@umt.edu.my

ABSTRACT: *Ageing is a phenomenon in which every creature on earth will definitely bound into it. Ageing is inevitable process, but with an understanding of the ageing factors, it in some way helps to delay the aging process. Software is not spared from being old, but unlike human ageing, software ageing can be delayed by identifying factors that influence the ageing. Previous study indicated that software ageing factors can be formulated by relevant, failure, cost, technology and environment. Our previous works in software quality and certification motivated and led us to the development of software anti-ageing model and its related areas such as software ageing factors and rejuvenation index. This paper presents the background studies in software ageing which includes software quality and certification, and focus further on the theoretical framework of software anti-ageing model.*

KEYWORDS: Software ageing factors, Software rejuvenation, software quality, software anti-ageing, evergreen software

1.0 INTRODUCTION

Software ageing has been introduced for certain time ago and not really considered as a new phenomenon in software engineering. It has been studied for the last eighteen years in various aspects of computer science. A significant body of knowledge has been established in the field of Software Aging and Rejuvenation (SAR)[1]. Earlier definition of software ageing refers to accumulation of errors during the software execution, which are ultimately results in crash or hanging failure [2]. Degradation of software performance also leads to the occurrence of software ageing. According to Parnas there are two types of software ageing, which are caused by the results of the changes that have been made and the failure of the software or system to adapt with dynamic environment[3].

In new technologies demand today, software engineers and practitioners need be a technology savvy in order to cope with the changes required rapidly. Failure to adapt with technology changes will results the relevance and important of software getting lesser to its environment which is called a phenomenon of getting old and age. Software must be built with the nature of modifiability and scalability, thus will give flexibility and enable it to stay young and relevant [7,8,17]. The process of delaying the ageing is called software rejuvenation. It can be done by detecting and classifying the ageing factors that may caused the ageing and implement the reverse action to convey the anti-ageing process.

In this paper, the research background in software ageing and software quality will be discussed and explored. Previous works done by our research group motivate and lead us to the development of the theoretical framework of software anti-ageing and its related areas such as software ageing factors and rejuvenation.

2.0 RESEARCH BACKGROUND AND RELATED WORK

Software ageing refers to the phenomenon of progressive performance degradation of the long running software, which may lead to undesirable hangs or software crashes[4]. Previous studiesindicated that the relevance of the software throughout its life span depended on the quality of the software. Thus, software ageing is closely related to the quality and certification of the software throughout its life cycle in the specific environment [7,17].

2.1 Software Quality

The concern and awareness of the software quality have been increasing in most industrial sectors. Quality by definition is not a specific concept but an abstract measurement. Quality can be viewed as making user to identify about the grade of the product or services, which is related to customer's satisfaction[5]. On the other hand, software quality can also be measured by three categories of measurements which are: internal measures, external measures and quality in use measures [6]. Internal measures is the process of evaluating on static measures of intermediate product, external measures evaluate on the behavior of the code while the quality in used evaluates the basic set of quality in used characteristics which may affect the software in certain operating environment. From the user's perspective, software product is more likely as a black box that must effectively support their business processes. Therefore, business need is one of the factors that influence the development of software quality.

Previous software quality model such are McCall model (1977), Boehm model (1978), FURPS model (1987), ISO9126 (1991), Dromey model (1996), Systematic Quality model (2003), and PQF model (2007) shown that the software quality characteristics that has been found in most models are: efficiency, maintainability, usability, reliability, functionality and portability [8]. Human aspect is a new element of software quality measurement that has been included in PQF model which are not introduced in earlier models. Measuring software product quality by reckoning the human aspect which relates to the user's perspective and

expectations are demanding today [8][18].

2.2 Software Ageing

Software ageing is a phenomenon in long-running software system that shows an increasing of failure rate in which the occurrence of a progressive degradation in software performance. The accumulation of software errors and failure to perform as user intended such as hang or crush also considered as ageing process [9][10]. Software ageing characteristics were described and identified as follows: memory bloating/leaks, shared- memory-pool latching, unreleased file locks, accumulation of un-terminated threads, file-space fragmentation, data corruption/round off accrual, thread stacks bloating and overruns[2][9][11]. Most of these studies focused on the ageing factors of a software systems.

Software ageing can also be understood as similarto biological system of human [3][12]. By using two remarkable examples such as ageing and software life cycle, software systems can be implied, and view as a category of organism. This analogy is appropriate because it creates certain realization about the software. First, the system exists (software as a community of intercommunicating agents) inside a given environment. Furthermore, much like their biological counterparts, they evolve (adapt to their environment) and they grow old. Finally, the life cycle is a series of stages which a living thing passes from the beginning of its life until its death. So that, from the software life cycle, we can imply that software system eventually dies [13].

However, the causes of software ageing are different from the biological organism (such as humans). The human will get older when the time passes by which can be measured by year. Opposite with the nature and human, software will not subject to fatigue or physical deterioration[13]. Software is not getting old along with time. Some of the software experiences ageing approximately two to four years after being used. Is that can be considered as software ageing? Numbers cannot determine the age of a software. Does not matter how long the system has been used as long as the software in the good quality and dynamic with environment changes, the software will stay young and healthy[17]. Though, software ageing is inevitable but by understanding the factors of software ageing, in some ways may help to prevent the occurrence of ageing. Software ageing factors can be defined in two categories which are internal factors and external factors. Those factors will be explained in more detail the next section.

2.3 Software Ageing Factors And Software Rejuvenation

The literature study has revealed that there are some factors which may lead to software ageing: internal factors and external factors. Certain actions are needed in order to ensure and to keep maintaining the high quality of the software which is via rejuvenation process. The rejuvenation process includes the maintaining activities (adaptive, corrective, preventive, and perfective), redesign, realignment, restructuring and etc. [7][14]. These actions should be carried out continuously until the software ends and terminates and replaces by a new software.

Several factors have been identified through literature study as the software ageing factors as shown in Table 1. While Table 2 shows the action in which the possible activity to prevent the software from getting old earlier than expected [7]. These actions are considered as rejuvenation activities of a software and will resulteda young and evergreen software in the operating environment of the software.

Table 1: Software Ageing Factors

Ageing Factors	
Internal factors	Memory bloating Residual defects Memory leak Unreleased file locks Debug Failure Declining quality Increasing complexity
External Factors	Environment dynamic Technology challenges (Hardware and Software) Competition Business compatibility & stability Requirement evolution High maintenance cost Ontology change

Table 2: Prevention action

Prevention Action
Addaptive Corrective Preventive Perfective Restructuring Redesign Realignment Redeployment

2.4 Anti-Ageing Actions and Factors

Anti-ageing is a process to delay, prevent, stop and retard the ageing from occur. Klatz [15] indicates that anti-ageing factors for human are by practicing a healthy lifestyle, such as avoid eating unhealthy food, avoid drinking alcohol, stop smoking, stay slim, regular exercises and stress reduction management. Unlike human ageing, software anti-ageing might differ than anti-ageing that has been defined by [15]. Prevention action as well as rejuvenation action that has been listed in Table 2 (adaptive, corrective, preventive, perfective and etc.) could be the anti-ageing action for application software.

Sci.Int.(Lahore),26(4),1615-1618,2014 ISSN 1013-5316; CODEN: SINTE 8

More work needs to be conducted in order to identify the factors that might influence or contribute to the occurrence of software ageing and anti-ageing. Correlation between the factors might be needed to identify the significance of anti-ageing factors for application software. In work done by [19] identifies ageing factors and their classifications as functional, environment, human and product contour or profile. For each classifications, there are associated metrics to measure the age of the software.

3.0 RESEARCH METHODOLOGY

Reseach methodology is a set of procedures to be performed in conducting a research. This research is conducted in five main phases. The first phase is the theoretical study, where the literature review of the existing related research is reviewed from journals, proceedings and technical reports. Apart from that review on the topic related to software ageing, the topics in software quality and software certificationneed to be reviewed and studied as well.

The second phase of this research is the empirical study in which a questionnaire is designed and distributed to software practitioners in Malaysia.The aim of this study is to identify current practices, methodologies and metrics used by the users in maintaining software performance throughout its life cycle. The survey is transmitted via mail, online survey and also interviews. Data from the survey will be analyzed by using statistical software.

The third phase is the data analysiswhich involvesanalyzing and identifying factors that influence software ageing based on the data collected from the empirical study.

The fourth phase is the formulation of the anti-ageing formula.The formula is derived from the software ageing factors that have been identified in the previous phase and will reverse the factors using reverse engine. The anti-ageing formula subsequently is tested in the final phase by evaluating the level of ageing for the selected application software.

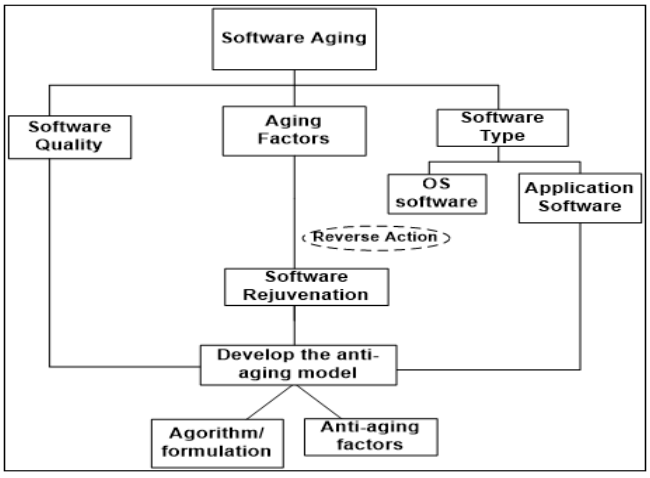

Figure 1: The Theoretical framework of Anti-Ageing Factors

4.0 THEORETICAL FRAMEWORK

The theoretical framework is an overview of definition in which both theory and concept in this research are described. The design of theoretical framework is based on study done in software ageing and software quality for application software.

A previous study has revealed that software ageing is closely related to software product quality. The effective approach for managing ageing of the software is through software quality and certification[7]. Figure 1 shows that several underlying theories in literature review might be helpful and valuableto develop the anti-ageing model for software application. The main theoretical aspects that needs to be considered are the software quality, ageing factors and software types.

4.1 Software Ageing and Factors

The main focus of this research is to study the occurrence of ageing in application software. Several questions need to be answered such as how the ageing occurs, how it affects the performances of the software, does it influence the working performance of software practitioners, what are the consequences of software ageing to an organization, software practitioners and the environment, anddo software practitioners aware of the ageing phenomenon? Literature study on software ageing is done to identify the factors that contribute to the software ageing process which also may be used formulate the anti-ageing factors.

4.2 Software Quality

Software quality is closely related to the occurrence of software ageing. The more quality of the software the less of software failure or software ageing will occur. Software quality can be measured by a number of variable which can be categorized by external variables and internal variable. Software quality practices and certification in the software operating environment is essential to ensure the software stays young and healthy.

4.4 Software Type

This research will study on application software ageing. Application software is a program that is designed to perform specific task for end users. Application software can be used as a business tool that supports and assists the business process. Therefore, it is crucial to study the phenomenon of software ageing towards application software. It is different from previous works where their focusesare on software systems [1][[2][13][16].

4.5 Software Rejuvenation

Software rejuvenation is the action or a technique of proactive fault that designs the system for periodic reboots [16]. It has been used to manage the occurrence of software ageing. The rejuvenation process includes the maintaining activities such as adaptive, corrective and etc. It prevents and delays the ageing by applying the reverse action of the ageing process.

4.6 Developing the Anti-Ageing Model

Application software anti-ageing model is still a new innovative topic. But, by understanding the human ageing might useful to explore it into a new application software domain. Based on the initial survey that has been distributed to small sample of software practitioners in Malaysia [17], the anti-ageing function of the application software has been identified and can be formulate as shown in Figure 2. However, further works need to be carried out in order to

verify the relevance of the formula as well as the practicality of the measurements in real environment.

$$\text{Application Software Anti-Aging} = f(\text{adaptive, corrective, preventive, perfective, redesign, realignment, restructuring…})$$

Figure 2: Application Software Anti-aging Formula

5.0 CONCLUSION

The initial works related to software ageing and software quality has been presented in this paper. The symptoms of degradation in term of software quality in the application software is observed as the indicator of ageing factors in application software. We have identify some initial factors of software ageing and thus requires further study and exploration to confirm the correlation between factors and formulate the anti-ageing formula and model. This anti-ageing formula and model is beneficial to prevent ageing and to ensure the software young and evergreen in the environment.

ACKNOWLEDGMENT

This research is funded by the Fundamental Research Grant Scheme, Malaysia Ministry of Higher Education.

REERENCES

[1] D. Cotroneo, R. Natella, R. Pietrantuono, and S. Russo, "Software Aging and Rejuvenation: Where We Are and Where We Are Going," 2011 IEEE Third Int. Work. Softw. Aging Rejuvenation, no. 30, pp. 1–6, Nov. 2011.

[2] K. J. Cassidy, K. C. Gross, and a. Malekpour, "Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers," *Proc. Int. Conf. DependableSyst. Networks*, pp. 478–482, 2002.

[3] D. L. Parnas, "Software Aging Invited," IEEE, pp. 279–287, 1994.

[4] J. Zhao, K. S. Trivedi, Y. Wang, and X. Chen, "Evaluation of software performance affected by aging," 2010 IEEE Second Int. Work. Softw. Aging Rejuvenation, vol. 3, pp. 1–6, Nov. 2010.

[5] H. Jin and F. Zeng, *Research on the definition and model of software testing quality*, Proc. 2011 9th Int. Conf. Reliab. Maintainab. Saf., pp. 639–644, Jun. 2011.

[6] W. Suryn, P. Bourque, a. Abran, and C. Laporte, "Software product quality practices - quality measurement and evaluation using TL9000 and ISO/IEC 9126," Intermag Eur. 2002 Dig. Tech. Pap.2002 IEEE Int. Magn. Conf., pp. 156–160, 2003.

[7] J.H. Yahaya and A. Deraman, "Towards the Anti-Ageing Model for Application Software," *Proc. Of the World Congress onEnggineering 2012*, London, UK, vol. II, 2012.

[8] J. H. Yahaya, A. Deraman, F. Baharom, and A. R. Hamdan, "Software Certification from Process and Product Perspectives," IJCSNS International Journal of Computer Science and Network Security, Vol. 9 No. 3, March2009.

[9] K. S. T. Sachin Garg, AadvanMoorsel, KalyanaramanVaidyanathan, "A Methodology for Detection and Estimation of Software Aging." Software Reliability Engineering, Proceedings. The Ninth International Symposium ,1998.

[10] M. Grottke, L. Li, K. Vaidyanathan, and K. S. Trivedi, "Analysis of Software Aging in a Web Server," IEEE Trans. Reliab., vol. 55, no. 3, pp. 411–420, Sep. 2006.

[11] P. Zheng, Q. Xu, and Y. Qi, "An Advanced Methodology for Measuring and Characterizing Software Aging," *2012 IEEE 23rd Int. Symp. Softw. Reliab. Eng. Work.*, pp. 253–258, Nov. 2012.

[12] S. Sustainment, "Geriatric Issues of Aging Software," December, pp. 4–8, 2007. The Journal of Defense Software Engineering. December, 2007

[13] C. Constantinides and V. Arnaoudova. (2009). " Prolonging the aging of aoftware systems," Encyclopedia of Information Science and Technology [Online]. Second Edition (8 Volumes).

[14] H. Van Vliet, "Software Engineering : Principles and Practice," 3rd edition, John Wiley &Sons, West Sussex, England.

[15] R. Klatz, "Definition of Anti-Aging Medicine," Academic Journal Article, *Generations* , Vol. 25, No. 4 , Winter 2002.

[16] Hanmer, Robert. "Software rejuvenation." *Proceedings of the 17th Conference on Pattern Languages of Programs*. ACM, 2010.

[17] J.H. Yahaya & A. Deraman. 2012. "Towards a Study on Software Ageing for Application Software: The Influential Factors", IJACT: International Journal of Advancements in Computing Technology, Vol. 4, No. 14, pp. 51 ~ 59, 2012.

[18] J. H. Yahaya, A.Deraman, S. R. Ahmad Ibrahim, A.R.Hamdan& Y. Y.Jusoh. 2013. User-Centric Approach of Software Certification: The Conceptual Model. *Proceedings of the 21st International Business Information Management Association Conference, Vienna, Austria*, June 2013, pp. 77-85.

[19] J.H. Yahaya, Z. N. ZainalAbidin, N. Mohd Ali & A. Deraman. "Software Ageing Measurement and Classification Using Goal Question Metric (GQM) Approach", *Proceeding of 2013 Science and Information Conference, 7-9 OCT 2013, London, UK*, pp. 160-165.