November-December 2019
ISSN: 0193-4120 Page No. 3081- 3086

Constructing a Model of Risk Mitigation for Anti Software Ageing During Software Maintenance

Thamaratul Izzah Azman[1], Noraini Che Pa[2], Rozi Nor Haizan Nor[3], Yusmadi Yah Jusoh[4]
[1,2,3,4]*Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, 43400 Seri Kembangan, Selangor, Malaysia;*
[1]*eizahazman@gmail.com,* [2]*norainip@upm.edu.my,* [3]*rozinor@upm.edu.my,* [4]*yusmadi@upm.edu.my*

Article Info
Volume 81
Page Number: 3081- 3086
Publication Issue:
November-December 2019

Article History
Article Received: *5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *14 December 2019*

Abstract:
Software will normally undergo an inevitable ageing process during its lifecycle. Most often, software rejuvenation is mainly proposed by past researchers to delay software ageing caused by errors accumulation from long running software execution. This method addresses software ageing from software dependability perspective. However little attention has been paid to address software ageing from software engineering perspective, which caused by failures to modify the software or from the results of software changes. Changes are vital to enable software accommodating new functions, ensuring its survivability towards new environment but pioneer in software ageing field argued that the results of software changes could influence software ageing. Hence, this motivate the study to develop a method to tackle software ageing from software engineering perspective to reduce risks impact before it become apparent. This paper discusses on the development of conceptual model of risk mitigation for anti software ageing during software maintenance through concepts gained from literature study. The design of the conceptual model illustrates the relationships between related concepts through concept mapping. It is significant to portray preliminary work in the study thus provide direction for further work to examine the relationships between concepts.

Keywords*: Anti Software Ageing, Risk Mitigation, Software Maintenance.*

1. INTRODUCTION

An event where software degrades in term of its performance and quality is known as software ageing. Generally, there are two types of software ageing perspectives described by pioneer in this field: software dependability perspective and software engineering perspective. In software dependability perspective, Huang et. al., (1995) stated that software ageing appears from increasing software failures caused by accumulation of errors from continuously running software. In software engineering perspective, Parnas (1994) highlighted that software ages due to software product's owner failures to modify it or fromthe results of changes made to software.Study in this area concerned with examining and handling the impact of software changes leading to software ageing (Russo, 2014).However, none of related works consider ageing caused from the result of changes (Mahmud, 2017). Therefore, ageing from software engineering perspective are yet to be explored.Thus,the motivation of the study is to gain insight on risks of software changes during maintenancethat influence software ageing and concept of risk mitigation from literature then further develop a model to overview the relationships between the concepts. The structure of this paper begins with background studies and related work, method for the study, discussion on the model developed and lastly, concluded with a summary of the work and further research expectation.

2. BACKGROUND STUDIES AND RELATED WORK

Anti software ageing is software that maintain its high quality and performance andis adaptable to environmental changes, which

enables the software to stay relevant (Abidin et. al., 2018). Existingmethod for anti software ageingvaried from software rejuvenation, software life extension, adopted maintenance tasks, partial computation offloading, software regeneration strategies and recovery actions.Most often, past researchers proposed software rejuvenation to delay the software ageing (Qin et. al., 2018) where software will be restarted or rebooted at a specific interval.However, the methodincreasessoftware overhead and downtime (Li et. al., 2018).Machida et. al., (2017) proposed software life extension which prolongs the software lifespan by allocating extra software resources and workloads control, however, it unable to directly fix errors and bugs manifestation inside the software. Abdullah et. al., (2017) adopted four maintenance tasks (preventive, adaptive, corrective and perfective)that are applied according to ageing factors identified through assessment, however, extensive maintenance increasessoftware complexity and causing software inadaptability to be maintained in future (Mahmud, 2017). Wu and Wolter (2015) suggested partial computation offloading method through determining and counteracting ageing by application-level, which effectively reduces resource consumption andapplication response as well as execution time. However, it involves effort to disintegrate modules of application software to the cloud services (Yiu& Lee, 2014).Jun et. al., (2011) proposed software regeneration strategies throughdetermining the degree of ageing states in each component of software by threshold interval andemployed regeneration strategies based on the results, but the method depends on the recovery time predictionaccuracy.Grottkeet. al., (2016) proposed recovery actions such as restart, reboot, reconfigure and hot-fix (that requires minor change to source code or changes to the software system such as OS), however itonly dealt with aging-related bugs.

From the literature study, there is a lack of discussion on software ageing from software engineeringperspective. Mahmud (2017) addressed software ageing based on software engineering perspective, by developing a simple software rejuvenation framework and argued that manual coding during software changes increases human errors in the software system thus the model aids in code automation to reduce errors made. However, unavailability of a tool to generate codes from the framework in the current marketlimits the method effectiveness.Nonetheless, most past researchers focused on proposing methods to address software ageing from software dependability perspective thus resulting in insufficient method to address ageing from software engineering perspective especially through risk mitigation. Further, the relationship between the risks of making changes during maintenance influencing software ageing is yet to be tested thus the study focusto fill in the literature gap.

3. METHOD

The development of the conceptual model is throughliterature review and concept mapping. Literature review is done in order tosought answers forfollowing questions:

Question 1: What are the existing models for anti software ageing?

Question 2: What are the risks of software changes during maintenance influencing the occurrences of software ageing?

Question 3: What are the activities in the existing model of risk mitigation?

A total of 121 studies had been gathered and reviewed from four digital libraries (IEEExplore, Science Direct, SpringerLink and Google Scholar). To answer question 1, we had distinguish several existing models for anti software ageing, however, the models provide inadequate support to minimize the potential impact of risk of software changes during maintenance.The relationship between the risks of making changes during maintenance influencing software ageing is yet to be tested and method to reduce the risks impact is still inadequate. Thus, this paves the way for thestudy to develop a model to reduce the possible impact of risks for anti software ageing and examine the relationship between risks of software changes during maintenance and

software ageing. To answer question 2, we had identified six dimensions of risk, which includes human, technical, environment, technology, resources and maintenance procedures and process.To answer question 3, we had identified and assessed tenmodels of risk mitigationand distinguish the activities involved during risk mitigation.

Further, model design and development is in line with concept mappingfoundation, portraying networks of concepts and illustrating the relationship between different concepts (Tergan, 2005). The development of the model helps to overview the relationships between components of the model for further work.

4. MODEL DESIGN AND DEVELOPMENT

The model are derived based on the model by Yahaya et. al., (2016) where the researchers uses factors to software ageing as an input, which are then assessed through software ageing index level assessment as the method to examine ageing and later assigned ageing guidelines according to the index level to attain anti software ageing as the output. However, in this study, we investigate on the risks of software changes that influence software ageing as input and risk mitigation as a method to reduce risks to software ageing in order to achieve anti software ageing. From the literature, risk mitigation includes several activities such as risk identification, risk treatment,risk assessment, risk decisionand risk monitoring, but not all of the activities are included in each previous existing risk mitigation model. Thus, in this study, we employed all of the activities in risk mitigation in order to develop an effective risk mitigation method to achieve anti software ageing.Figure 1 illustrates the conceptual model developed for the study. There are four components in this model comprises of risks of software changes during maintenance, software ageing, risk mitigation and anti software ageing. Further, we discussed the components of the model.

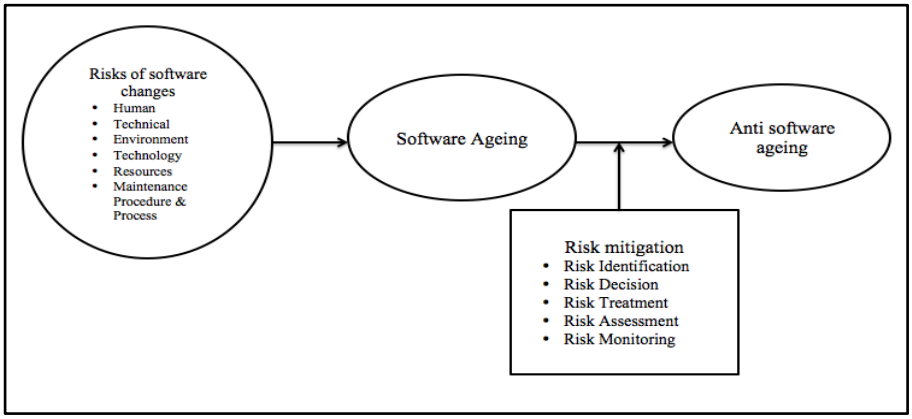

Figure 1.Conceptual Model of Risk Mitigation for Anti Software Ageing during Maintenance

4.1 Risk of software changes during maintenance

Risk is described as uncertain event or condition of making changes during software maintenance if it occurs has an impact to introduce software ageing. Six risks dimensionidentified from the literature areclassified as in Table 1.

Table 1. Risk of software changes during maintenance contributing to software ageing phenomenon

Risk	Description	References
Human (Risk stems from the deficiency or failure from human)	Lack of management commitment	Salmeron& Lopez, 2012
	Insufficient expertise, inexperience maintainers.	Kamei et. al., 2013; Salmeron& Lopez, 2012;
	Inadequate trained maintainers	Elzamly&Hussin, 2014; Ahmad and Kumar, 2014; Salmeron& Lopez, 2012;
	Inadequate skilled maintainers	
	Lack of maintainers' knowledge	
	Lack of support services	Elzamly&Hussin, 2014

Technical (Risk stems from the deficiency or failure of software component)	Increase in software faults	Ogheneovo, 2014
	Software inadaptability feature	
	Software code degradation	Salmeron& Lopez, 2012; Gupta and Sharma, 2015; Oghoneovo, 2014; Ahmad and Kumar, 2014;Kamei et. al., 2013; Bettenburg et. al., 2012; Trumper et. al., 2012
	Increase in software complexity	Tripathy&Naik, 2015;Ogheneovo, 2014; Tombe et. al., 2014; Trumper et. al., 2012
	Interface/design defects	Tripathy&Naik, 2015; Gupta and Sharma, 2015; Kamei et. al., 2013; Oghoneovo, 2014; Bettenburg et. al., 2012;
Environment (Risk stems from the deficiency or failure of perceiving environmental changes)	Misunderstanding/conflicting business process	Ahmad and Kumar, 2014
	Misunderstanding/conflicting business requirement	Knodel and Naab, 2014; Ahmad and Kumar, 2014; Salmeron& Lopez, 2012;
	Frequent changes in users' need	Oghoneovo, 2014; Salmeron& Lopez, 2012;
Technology (Risk stems from the deficiency or failure from technological challenges)	Hardware compatibility	Khan & Mattson, 2012
	Software platform compatibility	
	Wrongly fit of upgraded with its existing application	Elzamly&Hussin, 2014; Salmeron& Lopez, 2012;
Resources (Risk stems from the deficiency or failure from management of resources for software changes)	Inadequate cost	Tripathy&Naik, 2015;Gupta and Sharma, 2015; Ogheneovo, 2014; Tombe et. al., 2014; Elzamly&Hussin, 2014; Trumper et. al., 2012
	Lack of time allocated	Gupta and Sharma, 2015; Knodel and Naab, 2014; Oghoneovo, 2014; Salmeron& Lopez, 2012;
	Wrong estimation of resources	
	Lack/poor/null documentation	Gupta and Sharma, 2015; Oghoneovo, 2014; Elzamly&Hussin, 2014; Ahmad and Kumar, 2014; Salmeron& Lopez, 2012;
Maintenance procedure/process (Risk stems from the deficiency or failure maintenance procedure and process)	Complex procedure	Ahmad and Kumar, 2014; Salmeron& Lopez, 2012
	Inadequate procedure	Salmeron& Lopez, 2012
	Poor establishment of standard procedures, processes or methods	Ahmad & Kumar, 2014
	Lack of maintenance technique	Ahmad and Kumar, 2014; Salmeron& Lopez, 2012;

4.2Software ageing

Specifically, ageing in software is not measured physically, but rather is measured through its relevancy and importance to its environment. Software is described to be old once it gives less benefit to the users (Abdullah et. al., 2017).Software ageing occurs due to failure of performing software changes or from the impact of changes made to software (Russo, 2014) consequently causing software performance and quality degradation (Abidin et. al., 2018). For this reason, it is significant to restrain ageing occurrences in software as it may affect the users' satisfaction to use it.

4.3Risk mitigation

As software ageing is inevitable, alternatively, we may slow down the process of ageing. In this study, risk mitigation will assist to measure potential risks impact to software ageing during maintenance impact analysis phase in order to achieve anti software ageing.Risk mitigation isa method of reducing risk probability and its consequences. It assists in determining risks, providing convenient decision-making support and control the emergence of risk (Xiao et. al., 2013). Mitigation activities includes identifying potential risks, prioritizing risks, selecting best risk action solutions, assessing the successfulness of solution applied and lastly, monitoring treated and residual risks.

4.4 Anti Software Ageing

Anti software ageing is software that maintain its high quality and possess adaptability feature towards its environmental changes enabling the software to stay relevant (Abidin et. al., 2018).The attribute of software to be modified and updated enables the software to stay relevant and flexible towards dynamic environment (Abdullah et. al., 2017). Anti software ageing could be attained when the impact of making changes is lessen. The study adopts Bombardieri and Fontana (2009) usability characteristics to evaluate the software

November-December 2019
ISSN: 0193-4120 Page No. 3081- 3086

capability and examine the relevancy state of the software to meet users' need and expectation.

5. CONCLUSION

In brief, this paper interest is to overview software ageing from software engineering perspective, which had received little attention from previous researches and had presented the development of risk mitigation for anti software ageing model using concept mapping foundation. Risks of software changes during maintenance are identified and risk mitigation activities from existing risk mitigation models are assessed, both gathered through literature review to assist in the model development. Further, the model consists of four components including (1) risks of software changes as an input which contributes to(2) software ageing, (3) risk mitigation as a method to reduce impact of risks to software ageing and lastly (4) anti software ageing, as an output. The model proposed is a preliminary work,which will behelpful to provide general overview ofthe relationships between its components. We believe this study contributes to fill in the research gap in software ageing field as studies that discusses on software ageing from the results of software changes had been paid a little attention. Further work will be performed to examinethe relationships between the components in the model as well as validating themodel.

6. ACKNOWLEDGMENT

This work was supported by Malaysian Ministry of Education under Putra Graduate Initiative Grant, Universiti Putra Malaysia (GP-IPS/2018/9644600).

7. REFERENCES

1. Abdullah, Z. H., Yahaya, J. H., Mansor, Z., &Deraman, A. (2017).Software Ageing Prevention from Software Maintenance Perspective–A Review. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *9*(3-4), 93-96.
2. Abidin, Z. N. Z., Yahaya, J. H., Deraman, A., & Abdullah, Z. H., Rejuvenation Action Model for Application Software.In 2018 6th International Conference on Information and Communication Technology (ICoICT), 2018, pp. 38-43.IEEE.
3. Ahamad, S. (2016).Program Aging and Service Crash.International Journal of Computer Applications Technology and Research 5(7).
4. Ahmad, K., & Kumar, A. (2012). Forecasting risk and risk consequences on ERP maintenance.*International Journal of Soft Computing and Engineering*,*2*(5),13-18.
5. Bettenburg, N., Shang, W., Ibrahim, W. M., Adams, B., Zou, Y., & Hassan, A. E. (2012). An empirical study on inconsistent changes to code clones at the release level. *Science of Computer Programming*, *77*(6), 760-776.
6. Chen, P., Qi, Y., Li, X., Hou, D., &Lyu, M. R. T., ARF-Predictor: Effective prediction of aging-related failure Using Entropy. IEEE Transactions on Dependable and Secure Computing, **15**(4), 2018, 675-693.
7. Elzamly, A., &Hussin, B. (2014).Evaluation of quantitative and mining techniques for reducing software maintenance risks. *Appl. Math. Sci*, *8*(111), 5533-5542.
8. Grottke, M., Kim, D. S., Mansharamani, R., Nambiar, M., Natella, R., & Trivedi, K. S., Recovery from software failures caused by mandelbugs. IEEE Transactions on Reliability, **65**(1), 2016, 70-87.
9. Gupta, A., & Sharma, S. (2015). Software Maintenance: Challenges and Issues. *International Journal of Computer Science Engineering (IJCSE) Issues*,*1*(1),23-25.
10. Huang, Y., Kintala, C., Kolettis, N., & Fulton, N. D. (1995, June). Software rejuvenation: Analysis, module and applications. In *Twenty-Fifth International Symposium on Fault-Tolerant Computing. Digest of Papers* IEEE. (pp. 381-390).
11. Jia, S., Hou, C., & Wang, J., Software aging analysis and prediction in a web server based on multiple linear regression algorithm. In Communication Software and Networks (ICCSN),IEEE 9th International Conference, 2017 IEEE.pp.1452-1456.
12. Jun, G., Bo, W., Yunsheng, W., Bin, Z., &Jiaojiao, W., Research of the software

November-December 2019
ISSN: 0193-4120 Page No. 3081- 3086

aging regeneration strategy based on components.In Proceedings of the 2011, International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011) November 19–20, 2011, Melbourne, Australia, Springer. pp. 601-608.

13. Khan, A. S., &Kajko-Mattsson, M. (2012, April).Evaluating the role of maintenance environment activities in software handover. In *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*(Vol. 1, pp. 230-237). IEEE.
14. Knodel, J., &Naab, M. (2014, February).Mitigating the Risk of software change in practice: Retrospective on more than 50 architecture evaluations in industry (Keynote paper). In *2014 Software Evolution Week-IEEE Conference on Software Maintenance,Reengineering and Reverse Engineering(CSMR-WCRE)*.IEEE. (pp. 2-17).
15. Machida, F., Xiang, J., Tadano, K., &Maeno, Y. (2017). Lifetime extension of software execution subject to aging.*IEEE Transactions on Reliability*,*66*(1),123-134.
16. Mahmud, H. (2017). A Simple Software Rejuvenation Framework Based on Model Driven Development. *UHD Journal of Science and Technology*, *1*(2), 37-45.
17. Ogheneovo, E. E. (2014). On the relationship between software complexity and maintenance costs. *Journal of Computer and Communications*, *2*(14), 1.
18. Parnas, D. L. (1994, May).Software aging.In *Proceedings of 16th International Conference on Software Engineering* (pp. 279-287).IEEE.
19. Qin, F., Zheng, Z., Qiao, Y., & Trivedi, K. S. (2018).Studying Aging-Related Bug Prediction Using Cross-ProjectModels.*IEEE Transactions on Reliability*,(99),1-20.
20. Russo, S. (2014, November). The Dual Nature of Software Aging: Twenty Years of Software Aging Research. In *2014 IEEE International Symposium on Software Reliability Engineering Workshops*.IEEE.(pp. 431-432).
21. Salmeron, J. L., & Lopez, C. (2012). Forecasting risk impact on ERP maintenance with augmented fuzzy cognitive maps. *IEEE Transactions on software engineering*, *38*(2), 439-452.
22. Tergan, S. O. (2005). Digital concept maps for managing knowledge and information. In *Knowledge and information visualization.*Springer.(pp. 185-204).
23. Tripathy, P., &Naik, K. (2014). *Software Evolution and Maintenance: A Practitioner's Approach*. John Wiley & Sons.
24. Wu, H., &Wolter, K., Software aging in mobile devices: Partial computation offloading as a solution. In Software Reliability Engineering Workshops (ISSREW), 2015 IEEE International Symposium on (pp. 125-131). IEEE.
25. Xiao, J., Osterweil, L. J., Chen, J., Wang, Q., & Li, M. (2013, May). Search based risk mitigation planning in project portfolio management. In *Proceedings of the 2013 international conference on software and system process* (pp. 146-155).ACM.
26. Yiu, L. and Lee, M. J., "An effective dynamic programming offloading algorithm in mobile cloud computing system," in Wireless Communica- tions and Networking Conference (WCNC), 2014 IEEE, pp. 1868–1873, IEEE, 2014.