UNIVERSITI PUTRA MALAYSIA

A MODEL OF SOFTWARE CHANGE RISK ASSESSMENT MEASURES USING RISK MITIGATION PROCESS IN ANTI-AGEING

THAMARATUL IZZAH BINTI AZMAN

FSKTM 2020 27

A MODEL OF SOFTWARE CHANGE RISK ASSESSMENT MEASURES USING RISK MITIGATION PROCESS IN ANTI-AGEING

By

THAMARATUL IZZAH BINTI AZMAN

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Master of Science

July 2020

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Master of Science

A MODEL OF SOFTWARE CHANGE RISK ASSESSMENT MEASURES USING RISK MITIGATION PROCESS IN ANTI-AGEING

By

THAMARATUL IZZAH BINTI AZMAN

July 2020

Chairman : Associate Professor Noraini binti Che Pa, PhD
Faculty : Computer Science and Information Technology

Continuous changes during maintenance may cause software structure to deteriorate and causing bugs and errors, which reduces software quality leading to software ageing. Software ageing is inevitable, however, the progress of ageing can be delayed to attain software anti-ageing where software maintains its quality. Change analysis involves assessment of risks that monitor, examine and manage the impact of software changes to deal with software ageing, however, it is still unclear how maintainers perform the assessment of risks. Existing risk mitigation is lack of quantifiable approach, which arises ambiguous issues for change analysis. There are also inadequate tools to support maintainers for change analysis during software maintenance for software anti-ageing. The study aims to determine the risks of software changes that influence software ageing from software engineering perspective and further develop a model of software change risk assessment measures using risk mitigation process in anti-ageing as well as building a prototype based on the model. From a comprehensive theoretical study, six risks of software changes such as human, technical, environment, technology, resources and maintenance procedure and process were discovered to influence software ageing during software maintenance. To examine the relationships between those risks and software ageing, a quantitative survey was conducted using a structured questionnaire among 152 software practitioners in Malaysia. The data was analyzed using Structural Equation Modeling (SEM) analysis consists of measurement model and structural model assessment through SmartPLS software. The result shows that human risk, technical risk, environment risk, technology risk and maintenance procedure and process risk have significant effect on software ageing. The study found resources risk has no significant effect on software ageing. The study also discovered that risk mitigation is a moderator for the relationship between software ageing and software anti-ageing, where the interaction term demonstrates a significant path with the p-value of interaction term is 0.0001, which is smaller than recommended p-value of 0.05. Based on the results, a model of software change risk assessment measures using risk mitigation process in anti-ageing is developed that comprises of five components such as change request, risks of software changes, software ageing, risk mitigation process and software anti-ageing. Then, a prototype named Risk Mitigation for Software Anti-Ageing System is built based on the model. The model was validated and prototype was verified using expert or accreditation approach through interview with experts. The findings from this research contributes to assist maintainers to monitor, evaluate and manage risk of software changes that influence software ageing during change analysis in software maintenance to achieve software anti-ageing.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Master Sains

MODEL LANGKAH PENILAIAN RISIKO PERUBAHAN PERISIAN MENGGUNAKAN PROSES PENGURANGAN RISIKO DALAM ANTI PENUAAN

Oleh

THAMARATUL IZZAH BINTI AZMAN

Julai 2020

Pengerusi	**:**	**Profesor Madya Noraini binti Che Pa, PhD**
Fakulti	**:**	**Sains Komputer dan Teknologi Maklumat**

Perubahan secara berterusan semasa penyelenggaraan perisian menyebabkan struktur perisian merosot dan membawa kepada pepijat dan kesilapan yang mengurangkan kualiti perisian menyebabkan penuaan perisian. Penuaan perisian tidak dapat dielakkan, bagaimanapun, perkembangan penuaan boleh ditangguhkan untuk mencapai anti- penuaan perisian di mana perisian boleh mengekalkan kualitinya. Analisis perubahan melibatkan penilaian risiko untuk memantau, memeriksa dan mengurus kesan perubahan perisian bagi menangani penuaan perisian, namun, masih belum jelas bagaimana penyelenggara perisian melaksanakan penilaian risiko. Pengurangan risiko yang sedia ada kekurangan pendekatan kuantitatif menimbulkan masalah kesamaran untuk analisis perubahan. Terdapat juga alat yang tidak mencukupi bagi menyokong penyelenggara untuk analisis perubahan semasa penyelenggaraan perisian untuk anti- penuaan perisian. Kajian ini bertujuan untuk mengenal pasti risiko melaksanakan perubahan perisian yang mempengaruhi perisian untuk menjadi tua dari perspektif kejuruteraan perisian dan seterusnya membangunkan model langkah penilaian risiko perubahan perisian menggunakan proses pengurangan risiko untuk anti penuaan serta membina prototaip berdasarkan model tersebut. Dari kajian teori yang komprehensif, enam risiko perubahan perisian seperti manusia, teknikal, persekitaran, teknologi, sumber dan prosedur dan proses penyelenggaraan telah didapati mempengaruhi penuaan perisian semasa penyelenggaraan perisian. Untuk menkaji hubungan risiko tersebut dengan penuaan perisian, satu tinjauan kuantitatif telah dilakukan dengan menggunakan soal selidik berstruktur dalam kalangan 152 pengamal perisian di Malaysia. Data dianalisis menggunakan analisis Structural Equation Modeling (SEM) yang terdiri dari penilaian model pengukuran dan model struktur melalui perisian SmartPLS. Hasil kajian mendapati bahawa risiko manusia, risiko teknikal, risiko persekitaran, risiko teknologi dan risiko prosedur dan proses penyelenggaraan mempunyai kesan terhadap penuaan perisian. Penemuan kajian mendapati risiko sumber tidak memberikan kesan signifikan terhadap penuaan perisian. Pembolehubah moderator iaitu kaedah pengurangan risiko mempunyai kesan signifikan terhadap hubungan di antara penuaan perisian dan anti-penuaan perisian, di mana istilah interaksi menunjukkan jalan signifikan dengan nilai p dari istilah interaksi adalah 0.0001, lebih kecil daripada nilai p yang disyorkan 0.05. Berdasarkan hasil kajian, model Langkah penilaian risiko perubahan perisian menggunakan proses pengurangan risiko untuk anti penuaan telah dibina daripada lima komponen seperti permintaan perubahan, risiko perubahan perisian, proses pengurangan risiko dan anti-penuaan perisian. Kemudian, prototaip bernama Risk Mitigation for Software Anti-Ageing System dibina berdasarkan model tersebut. Model dan prototaip telah disahkan menggunakan pendekatan pakar atau pentauliahan melalui temu bual bersama pakar. Penemuaan dari penyelidikan ini dapat membantu pengamal perisian untuk memantau, menilai dan mengurus risiko perubahan perisian yang mempengaruhi penuaan perisian semasa analisis perubahan untuk mencapai anti-penuaan perisian.

ACKNOWLEDGEMENTS

Firstly, I would like to be thankful to the Almighty God, the creator of the whole universe, who gave me blessing for health and strength to accomplish and fulfil my master research despite the challenges faced and unwanted circumstances during the years.

Secondly, I would like to give my greatest appreciation to my supervisor Assoc. Prof. Dr. Noraini Che Pa for the help, guidance and kindness throughout the years of doing this research. She had been a great guide for this journey and gave me so much motivation to further my study. I am also grateful to have Dr. Rozi Nor Haizan Nor and Assoc. Prof. Dr. Yusmadi Yah Jusoh as my supervisory committee who have also guided, advised and encouraged me to finish the research.

I also would like to express my sincere appreciations to my beloved parents and family members for their support, understanding and encouragement that has emotionally lift my spirit to complete my study. Their sacrifices will not be forgotten.

Finally, I would like to thank to all who supported or guided me directly or indirectly during this research journey.

This thesis was submitted to the Senate of the Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. The members of the Supervisory Committee were as follows:

Noraini binti Che Pa, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Rozi Nor Haizan binti Nor, PhD
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Yusmadi Yah binti Jusoh, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

ZALILAH MOHD SHARIFF, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 10 December 2020

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software

Signature: ________________________ Date: _______________

Name and Matric No: Thamaratul Izzah binti Azman, GS49892

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) were adhered to.

Signature: ______________________

Name of Chairman of Supervisory Committee: Associate Professor Dr. Noraini binti Che Pa

Signature: ______________________

Name of Member of Supervisory Committee: Dr. Rozi Nor Haizan binti Nor

Signature: ______________________

Name of Member of Supervisory Committee: Associate Professor Dr. Yusmadi Yah binti Jusoh

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

3LRM	3 Layer Risk Mitigation Modeling
ARB	Aging-Related Bugs
AVE	Average Variance Extracted
BOHs	Bohrbugs
CA	Cronbach’s Alpha
CIA	Change Impact Analysis
CPU	Central Processing Unit
CR	Composite Reliability
CTHMM	Continuous Time-Hidden Markov Model
FR	Functional Requirement
GUI	Graphic User Interface
HMM	Hidden Markov Model
HTMT	Heterotrait Monotrait
ICT	Information and Communication Technology
IT	Information Technology
LV	Latent Variable
MV	Manifest Variables
NAMs	Non Aging-Related Mandelbugs
NFR	Non-functional Requirement
OS	Operating System
PLS-SEM	Partial-Least Squares – Structural Equation Modelling
Q^2	Predictive Relevance
QoS	Quality of Services

R^2	Coefficient of determination
SDLC	Software Development Life Cycle
SEM	Structural Equation Modelling
SPSS	Statistical Package for Social Sciences
SQuaRE	Software product Quality Requirements and Evaluation
UML	Unified Modelling Language
UPM	Universiti Putra Malaysia
VIF	Variance Inflator Factor
VMM	Virtual Machine Monitors

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter lay out the research background of the study to explain the context of the research, research problems to describe the problem statement of the research, research questions to illustrate the address research problems, research objectives which will highlight the aim of the research, research scope to define the focus of the research and lastly, research significance to present the contribution of the research.

1.2 Research Background

Nowadays, many people use software and computer program and system in their daily life to reduce the manpower. It is a vital and valuable asset for a successful business process as it functions to assist in performing tasks and activities within the organizations. Software that maintains its high quality and possess adaptability feature towards its environmental changes enables the software to stay young, relevant, and anti-ageing (Yahaya and Deraman, 2012). The quality of software is often related to the software technical behavior and end user's perspective towards the software, which measures users' satisfaction and software ability to fulfill the expectation (Abdullah et. al., 2019). Moreover, software adaptability and flexibility features towards dynamic changes allow the software to remain anti-ageing (Yahaya and Deraman, 2012).

The state of software anti-ageing prevents the chances of software errors and failures occurrences to sustain the survivability of software and prolonging its lifecycle. The process for anti-ageing consists of activities that can be applied to prevent software ageing progress before software becomes irrelevant (Abdullah et. al., 2014). Software ageing is referred to the degradation of software performance and quality to deliver and perform its expected function and provide services (Yahaya et. al., 2016; deMelo et. al., 2017). Software maintenance can be implemented to tackle software ageing phenomenon (Singh and Goel, 2007; Abdullah et. al., 2017). It is one of the phases in software development lifecycle where in the event of software errors and failures, actions will be taken to fix errors as well as modifying and updating the software to meet new requirements and demands to improve its performance, quality and services. Cotroneo et. al. (2010) suggested using software maintenance activities as a mean to delay ageing progress in software. Maintaining the software helps tackling ageing issues in software (Singh and Goel, 2007).

Software ageing from software engineering perspective is concerned with monitoring, examining and managing the impact of software changes that gives threats to software structure and its quality to deal with software ageing (Russo, 2014). The activity or task, which assist to discover impact of performing changes to software is done during change analysis in software maintenance (Tripathy and Naik, 2015). Software change analysis is performed during the second phase in software maintenance to assist in determining possible impact or effect of a change or the estimation of what needs to be modified (Bohner & Arnold, 1996). This phase helps managing software changes and inhibits software deterioration that may leads to software ageing once changes are implemented (Isong and Ekabua, 2013; Russo, 2014).

Particularly, it is crucial for experts or maintainers to assess the risks of performing software changes during change analysis in software maintenance to examine the risks level and its impact to software (Williams and Carver, 2006, Rahman et. al., 2019). Risk is defined as an uncertain or unpredictable event that if it occurs, will give a negative effect (KarimiAzari et. al., 2011). Risks during change analysis are crucial to be determined to reduce the impact or unanticipated event and consequences from those risks on the software before performing software changes (Williams and Carver, 2006; Rahman et. al., 2014). Through the early identification of the potential impact and failure from the changes or modification, the risk to deal with unforeseen impact and unpredictable changes on the software will be minimized (Isong and Ekabua, 2013).

There are various risks of software changes during software maintenance discussed by past researchers that is associated to influence software ageing phenomenon such as human, technical, environment, technology, resources and maintenance procedure and process. Few of the risks such as human, technical, environment and technology had been validated in the previous studies (Abdullah et. al., 2019, Abidin et. al., 2018, Mahmud, 2017 and Yahaya et. al., 2016). Nonetheless, those risks were investigated in software dependability perspective where software ageing occur after long running of software execution. For this reason, those studies disregard investigation of those risks on software ageing based on software engineering perspective, where software ages from the result of changes made to the software. Thus, to handle the concerns of software ageing based on software engineering perspective, investigating the relationships between those risks of software changes with software ageing is significant to evaluate its impact to software during change analysis.

Furthermore, failure of the maintainers to manage risks of software changes may leads the risks to be converted into serious issues that will compromise the maintenance success such as promoting increasing number of software failures, declining software performance and its quality and benefits from the software will not be gained by the users thus eventually causing early software retirement

(Salmeron and Lopez, 2012). One of the techniques used for software change analysis during software maintenance is risk mitigation. Knodel and Naab (2014) proposed risk mitigation to mitigate and minimize risk of software changes during change analysis in software maintenance through architecture evaluation.

The scope of mitigating risks in the study however only concentrates on qualitative evaluation on the architecture of the software to predict the impact of risks to software for risk assessment for change analysis. For that case, the interpretation of risks is not scalable as it involves in-depth descriptive understandings of the architecture of the software itself for the evaluation.

According to Ahamad (2016), effects of software ageing may cause difficulties to cope with business operation as it slows down software response time to execute commands in which later causing delayed works. Further, occasional system downtime causes by software ageing also may lead to economic loss (Qin et. al., 2015). Organization or business may need to invest their money to acquire new software after the loss of old software due to ageing (Abidin et. al., 2015). Therefore, it is crucial to counteract and prevent the occurrences of software ageing because it does not only affect the software abilities to function, but also influence users, business and organization experience to utilize the software as well.

Hence, the whole purpose of the study is to provide novel insight and guidance for maintainers in evaluating, monitoring and managing risks of software changes using quantifiable process of risk mitigation to prevent the software from negative consequences of software changes that will compromise its quality and performance. This helps to ensure better function and utilization of the software for it to remain relevant and stay anti-ageing in order to support on-going business and organization function.

1.3 Problem Statement

Software maintenance is one of the methods to deal with software ageing (Singh and Goel, 2007; Abdullah et. al., 2017). However, software maintenance also can promote software ageing from the results of changes performed into the software during software maintenance (Russo, 2014). Continuous changes during maintenance cause degradation of software structure and maintainability by introducing bugs and errors leading to software ageing (Russo, 2014; Mahmud, 2017; Catolino et. al., 2018).

Effective change analysis during software maintenance involves monitoring, examining, and managing the impact of software changes in order to deal with software ageing (Russo, 2014). To ensure effective change analysis, identification of probable risks of software changes is profoundly necessary during change analysis to evaluate the impact or consequences from those risks before changes are performed into the software (Rahman et. al., 2019). Existing risks of software changes discovered were investigated from software dependability perspective (Abdullah et. al., 2019, Abidin et. al., 2018, Mahmud, 2017 and Yahaya et. al., 2016). This causes lack of exploration on risks of software changes from software engineering perspective.

From theoretical study, existing software anti-ageing model provides inadequate support to determine impact from the results of software changes made during software maintenance that could lead to software ageing. Most of the proposed models concentrate on tackling software ageing issues from software dependability perspective (Abdullah et. al., 2019, Abidin et. al., 2018, Mahmud, 2017 and Yahaya et. al., 2016). Thus, these models lack support for software changes risk assessment for change analysis in software maintenance to achieve anti-ageing from software engineering perspective.

In particular, from software engineering perspective, software ageing also need to be dealt by handling results from the impact of changes made to software where it involves the assessment of risks (Russo, 2014). However, it is still yet uncertain how maintainers should analyze the impact of software changes and perform software change risk assessment for change analysis during software maintenance (Rahman et. al., 2019). Past researchers proposed risk mitigation process as change analysis technique, however, existing risk mitigation process is lack of quantifiable which arises ambiguous issues for change analysis (Knodel and Naab, 2014). Mitigation of risk is performed through qualitative evaluation on the architecture of software to discover the impact of risks. Hence, this arises the ambiguous issue on the risk mitigation process for software changes risk assessment for change analysis during software maintenance.

There is also lack of tools or mechanism to support maintainers for change analysis during software maintenance for software anti-ageing (Borg et. al., 2017; Abdullah et. al., 2019). Maintainers usually manage change analysis reports through a simple web interface for repository where it only supports browsing and searching the reports (Borg et. al., 2017). This creates the issue on inadequate platform or tool to support for change analysis during software maintenance.

Therefore, such gaps motivate the study to explore on the risks of software changes that could influence software ageing during software maintenance from software engineering perspective and further proposed a model of software change risk assessment measures using risk mitigation process in anti-ageing that provides scalable measures for quantifiable risk mitigation process for software changes risk assessment during change analysis in software maintenance. The study also will develop a prototype based on the model, which acts as a tool to assist maintainers for change analysis during software maintenance.

1.4 Research Questions

In order to address the research problems, research questions are formulated to drive the research direction. The research questions are described as followed:

1. What are the risks of software changes during software maintenance that influences software ageing from software engineering perspective?
2. How to examine the relationship between the risks of software changes during software maintenance and software ageing, and risk mitigation process moderating effect on the relationship between software ageing and software anti-ageing?
3. How does risk mitigation process help to mitigate risks of software changes for software anti-ageing?

1.5 Research Objectives

The objectives of the research that underline the way for the research are:

1. To determine risks of software changes during software maintenance that influence software ageing from software engineering perspective.
2. To develop a model of software changes risk assessment measures using risk mitigation process to anti-ageing.
3. To build a prototype based on the model of software changes risk assessment measures using risk mitigation process to anti-ageing.

1.6 Research Scope

To achieve the research objectives, the study is concerned with software ageing based on software engineering perspective and method to achieve software anti- ageing using risk mitigation. Hence, the scopes of the research are as followed:

1. The study is limited to be concerned with following point of knowledge: a) risk of software changes during change analysis in software maintenance that influence software ageing, and b) risk mitigation as process to achieve software anti-ageing.
2. The study will be conducted mainly among software practitioners in software industry in Malaysia to determine risks of software changes during software maintenance influencing software ageing and perceived

benefit of risk mitigation in preventing software ageing to achieve software anti-ageing based on their experience and knowledge.

3. Quantitative approach is adopted in this research using survey questionnaire to determine the risks of software changes during maintenance that influence software ageing and determine the moderating effect of risk mitigation process for software anti-ageing from experts' opinions. Meanwhile, to validate the model and verify the prototype, an interview and survey with several informants are performed.
4. All software programs used in this study are free and open source. The required hardware for developing the prototype in this study includes personal computer (PC) of 64bit, 1.3 GHz Intel Core i5, 4GB RAM supported with 1440 x 900 Intel HD Graphics resolution, meanwhile the software components used includes OS X Yosemite (operating system), Eclipse for system's programming, Xampp as web-server that provides MYSQL database and Adobe PDF for downloading report.

1.7 Research Significance

This research presents a new paradigm of software changes risk assessment for software anti-ageing using risk mitigation process. A model of software change risk assessment measures using risk mitigation process in anti-ageing proposed in the study will provide better understanding and insight for maintainers on the potential impact and consequences of performing software changes during maintenance that could affect the quality and performance of existing software. The study is essential to drive a new way to tackle software ageing phenomenon during software maintenance in order to ensure software anti-ageing. The research is also vital to discover new attributes that could contribute to influence software ageing phenomenon from the results of software changes based on software engineering perspective. Overall, the research is beneficial as new knowledge to the literature in the field of software engineering and information system.

The research also presents a new and feasible mechanism for handling and dealing with software ageing issues during software maintenance from the development of a prototype named Risk Mitigation for Software Anti-Ageing system (RMSAAsys). It aids to serve as a tool that support maintainers to conduct software changes risk assessment through a series of computerized risk mitigation process. The prototype will be beneficial for software maintainers as it offers convenient platform for easy management and mitigation of software changes risks in order to ensure software anti- ageing. The results and contributions from the studies may be applied for the use of learning institutions for learning purposes and software industries for effective software maintenance. The limitations and future work from this research also can be used for references and suggestions as a base for future research. The knowledge gained from this research also can be intended to be use for fundamental information in the context of software ageing, anti-ageing and software maintenance in the field of software engineering.

1.8 Thesis Organization

This thesis consists of several research works that have been reported in journal papers and conference papers. Chapter 1 covers research background, research problem, research questions, research objectives, research scope and research significance. Overall, this thesis comprises of 7 chapters, the remaining chapters are listed as follows:

Chapter 2 discussed on literature review which covers the definition and nature of software ageing, causes of software ageing, software anti-ageing, existing methods and model of software anti-ageing, software maintenance on change analysis, risk of software changes during software maintenance as well as risk mitigation process and existing risk mitigation system. In addition, research gap from the literature study is discussed and conceptual model is developed. It also covers on the development of hypotheses for the empirical study based on the components described in the conceptual model.

Chapter 3 explained on the research methodology comprises of five phases: Phase I involves theoretical study, Phase II on empirical study that involves questionnaire design, instrument content validity, pilot study, data collection and analysis, Phase III on model development**,** Phase IV for prototype development and lastly, Phase V involves model validation and prototype verification.

Chapter 4 discussed the results and findings from empirical study.

Chapter 5 presents the proposed model of software anti-ageing and discusses on model evaluation that includes prototype architectural design and implementation.

Chapter 6 discussed on the results of post-study on the model validation and prototype verification. The chapter also explains on the overall study results discussion.

Chapter 7 concludes the research summary and future work, which covers the research theoretical and practical contribution, limitations, and future work in the field.

REFERENCES

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.

Abdullah, Z. H., Yahaya, J., & Deraman, A. (2015). Towards Anti-Ageing Model For The Evergreen Software System. In *Electrical Engineering and Informatics (ICEEI), International Conference*, 388-393. IEEE.

Abdullah, Z. H., Yahaya, J., Ibrahim, S. R. A., Fadzli, S., & Deraman, A. (2019). The implementation of software anti-ageing model towards green and sustainable products. *International Journal of Advanced Computer Science and Applications*, 10(5), 42-50.

Abdullah, Z.H., Yahaya, J.H. and Deraman, A. (2014). The Anti-Ageing Factors for Evergreen Software–A Preliminary Study. *Science International*, 26(4),1615-1618.

Abidin, Z. N. Z., Yahaya, J. H., & Deraman, A. (2015). Software ageing measurement model (SAMM): The conceptual framework. In *Electrical Engineering and Informatics (ICEEI), 2015 International Conference*, 456-461. IEEE.

Abidin, Z. N. Z., Yahaya, J. H., Deraman, A., & Abdullah, Z. H. (2018). Rejuvenation Action Model for Application Software. In *2018 6th International Conference on Information and Communication Technology (ICoICT)*, 38-43. IEEE.

Agepati, R., Gundala, N., & Amari, S. V. (2013). Optimal Software rejuvenation policies. In *Reliability and Maintainability Symposium (RAMS), Proceedings Annual*, 1-7. IEEE.

Agudelo, C. A., Bosua, R., Ahmad, A., & Maynard, S. B. (2016). Mitigating Knowledge Leakage Risk in Organizations through Mobile Devices: A Contextual Approach. In *27th Australasian Conference on Information Systems 2016 – ACIS2016*, 1-12.

Ahamad, S. (2016). Program Aging and Service Crash. International Journal of Computer Applications Technology and Research. 5(7), 422-432.

Ahamad, S. (2016). Study of software aging issues and prevention solutions. *International Journal of Computer Science and Information Security*, 14(8), 307-313.

Ahmad, K., & Kumar, A. (2012). Forecasting risk and risk consequences on ERP maintenance. *International Journal of Soft Computing and Engineering*, 2(5), 13-18.

Ali, H. O., Rozan, M. Z. A., & Sharif, A. M. (2012, May). Identifying challenges of change impact analysis for software projects. In *2012 International Conference on Innovation Management and Technology Research*, 407-411. IEEE.

Aloini, D., Dulmin, R., & Mininno, V. (2012). Risk assessment in ERP projects. *Information Systems*, 37(3), 183-199.

Alonso, J., Matias, R., Vicente, E., Maria, A., & Trivedi, K. S. (2013). A comparative experimental study of software rejuvenation overhead. *Performance Evaluation*, 70(3), 231-250.

Alonso-Ríos, D., Vázquez-García, A., Mosqueira-Rey, E., & Moret-Bonillo, V. (2009). Usability: a critical analysis and a taxonomy. *International Journal of Human-Computer Interaction*, 26(1), 53-74.

April, A., & Abran, A. (2012). Software maintenance management: evaluation and continuous improvement, Vol. 67. John Wiley & Sons.

Araujo, J., Oliveira, F., Matos, R. D. S., Torquato, M., Ferreira, J., & Maciel, P. R. M. (2016). Software Aging Issues in Streaming Video Player. *JSW*, 11(6), pp. 554-568.

Avdoshin, S. M., & Pesotskaya, E. Y. (2011, October). Software risk management. In *2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR)*, 1-6. IEEE.

Avritzer, A., Cole, R. G., & Weyuker, E. J. (2010). Methods and opportunities for rejuvenation in aging distributed software systems. *Journal of Systems and Software*, 83(9), 1568-1578.

Barclay, D., Higgins, C., & Thompson, R. (1995). The partial least squares (PLS) approach to casual modeling: personal computer adoption ans use as an Illustration. Technology Studies, 2(2), 285-309.

Bajpai, V., & Gorthi, R. P. (2012, March). On non-functional requirements: A survey. In *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science*, 1-4. IEEE.

Baron, R. M., & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of personality and social psychology*, 51(6), 1173.

Bayaga, A., & Mtose, X. (2010). Quantitative risk analysis: Determining university risk mitigation and control mechanisms. *The Journal of International Social Research*, 3(12), 55-68.

Bodnar, G. H., (2008) IT Governance. *Internal auditing*. 18(3), 27-32.

Bombardieri, M., & Fontana, F. A. (2009) Software aging assessment through a specialization of the SQuaRE quality model. Software Quality, 2009. WOSQ'09. *ICSE Workshop*, 33-38. IEEE.

Bruneo, D., Distefano, S., Longo, F., Puliafito, A., & Scarpa, M. (2013). Workload- based software rejuvenation in cloud systems. *IEEE Transactions on Computers*, 62(6), 1072-1085.

Brown, T. A., & Moore, M. T. (2012). Confirmatory factor analysis. *Handbook of structural equation modelling*. Retrieved September 18, 361-379.

Bohner, S. A. (2002). Extending software change impact analysis into COTS components. *Proceedings of the 27th Annual NASA Goddard Software Engineering Workshop*, Greenbelt, USA, 175 -182.

Bohner, S. A., & Arnold, R. S. (1996). Software change impact analysis. *IEEE Computer Society Press*, Vol. 6, p. 277.

Borg, M., Wnuk, K., Regnell, B., & Runeson, P. (2017). Supporting change impact analysis using a recommendation system: An industrial case study in a safety- critical context. *IEEE Transactions on Software Engineering*, 43(7), 675-700.

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.

Byrne, B. M. (2010) Structural equation modeling with AMOS: Basic concepts, applications and programming (multivariate applications series). New York: Taylor & Francis Group, 396, 7384.

Canfora, G., & Cimitile, A. (2001). Software maintenance. In *Handbook of Software Engineering and Knowledge Engineering: Volume I: Fundamentals*, 91-120.

Catolino, G., Palomba, F., De Lucia, A., Ferrucci, F., & Zaidman, A. (2018). Enhancing change prediction models using developer-related factors. *Journal of Systems and Software*, 143, 14-28.

Chen, P., Qi, Y., Li, X., Hou, D., & Lyu, M. R. T. (2018). ARF-Predictor: Effective prediction of aging-related failure Using Entropy. *IEEE Transactions on Dependable and Secure Computing*, 15(4), 675-693.

Chin, W. W. (1998). The partial least squares approach to structural equation modeling. *Modern methods for business research*, 295(2), 295-336.

Chowdhury, A. A. M., & Arefeen, S. (2011) Software risk management: importance and practices. *International Journal of Computer and Information Technology*, ISSN, 2078-5828.

Chua, B. B. (2010, August). Rework requirement changes in software maintenance. In *2010 Fifth International Conference on Software Engineering Advances*, 252-258. IEEE.

Cotroneo, D., Fucci, F., Iannillo, A. K., Natella, R., & Pietrantuono, R. (2016). Software aging analysis of the android mobile OS. In *Software Reliability Engineering (ISSRE), 2016 IEEE 27th International Symposium*, 478-489.

Cotroneo, D., Natella, R., Pietrantuono, R., & Russo, S. (2010) Software aging analysis of the Linux operating system. In *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium*, 71-80. IEEE.

Cotroneo D. Natella R., Pietrantuono R., and Russo S. (2014). A survey of software aging and rejuvenation studies. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 10(1), 1-34.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. Psychometrika, 16(3), 297-334.

Creswell, J. W. (2013). Steps in conducting a scholarly mixed methods study. DBER Speaker series. University of Nebraska Discipline-Based Education Research Group (online).Retrievedfrom: http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1047&context=dberspea kers . Accessed: 4th January 2018

Danny, L. (2013). Reducing operational risk by improving production software quality. *Software Risk Reduction Rev*, 13, 1-15.

Davide, A., Dulmin, R., & Mininno, V. (2012). Risk Assessment in ERP projects. *Information Systems*, 37, 183-199.

deMelo, M. D. E. T., Araujo, J., Umesh, I. M., & Maciel, P. R. M. (2017). SWARE: An approach to support software aging and rejuvenation experiments. *Journal on Advances in Theoretical and Applied Informatics*, 3(1), 31-38.

de Vaus, D. (2004). Surveys in Social Research (5th ed.). London: Routledge.

DeVon, H. A., Block, M. E., Moyle-Wright, P., Ernst, D. M., Hayden, S. J., Lazzara, D. J. et al. (2007). A psychometric Toolbox for testing Validity and Reliability. *Journal of Nursing scholarship*, 2(2), 155-164.

Diamantopoulos, A., & Siguaw, J. A. (2006). Formative versus reflective indicators in organizational measure development: A comparison and empirical illustration. *British Journal of Management*, 17(4), 263-282.

Ding, W., Liang, P., Tang, A., & Van Vliet, H. (2014). Knowledge-based approaches in software documentation: A systematic literature review. *Information and Software Technology*, 56(6), 545-567.

Dohi, T., Zheng, J., Okamura, H., & Trivedi, K. S. (2018). Optimal periodic software rejuvenation policies based on interval reliability criteria. Reliability Engineering & System Safety, 180, 463-475.

Elzamly, A., & Hussin, B. (2014). An Enhancement Of Framework Software Risk Management Methodology For Successful Software Development. *Journal of Theoretical & Applied Information Technology*, 62(2), 410-423.

Elzamly, A., & Hussin, B. (2014). Mitigating software maintenance project risks with stepwise regression analysis techniques. *Journal of Modern Mathematics Frontier*, 3(2), 34-44.

Faisal, N. M., Banwet, D. K., & Shankar, R. (2007). Information risks management in supply chains: an assessment and mitigation framework. *Journal of Enterprise Information Management*, 20(6), 677-699.

Fakhrolmobasheri, S., Ataie, E., & Movaghar, A. (2018). Modeling and Evaluation of Power-Aware Software Rejuvenation in Cloud Systems. *Algorithms*, 11(10), 160.

Fang, Y., Yin, B. B., Ning, G., Zheng, Z., & Cai, K. Y. (2017). A Rejuvenation Strategy of Two-Granularity Software Based on Adaptive Control. In *Dependable Computing (PRDC), 2017 IEEE 22nd Pacific Rim International Symposium*, 104-109. IEEE.

Ficco, M., Pietrantuono, R., & Russo, S. (2018). Aging-related performance anomalies in the apache storm stream processing system. *Future Generation Computer Systems*, 86, 975-994.

Firdose, S., & Rao, L. M. (2016). 3LRM-3 Layer Risk Mitigation Modelling of ICT Software Development Projects. *International Journal of Electrical and Computer Engineering*, 6(1), 349.

Fornell, C., & Larcker, D. F. (1981). Structural equation models with unobservable variables and measurement error: Algebra and statistics. *Journal of Marketing Research*, 18(3), 382-388.

Ghobadi, S., & Mathiassen, L. (2017). Risks to effective knowledge sharing in agile software teams: a model for assessing and mitigating risks. *Information Systems Journal*, 27(6), 699-731.

Grottke, M., & Schleich, B. (2013). How does testing affect the availability of aging software systems?. *Performance Evaluation*, 70(3), 179-196.

Grottke, M., Kim, D. S., Mansharamani, R., Nambiar, M., Natella, R., & Trivedi, K. S. (2016). Recovery from software failures caused by mandelbugs. *IEEE Transactions on Reliability*, 65(1), 70-87.

Güngör, M. K., Elbaşi, E., & Fawcett, J. W. (2012). New change impact factor estimation in software development. Turkish Journal of Electrical Engineering & Computer Sciences, 20(1), 1-14.

Gupta, A., & Sharma, S. (2015). Software Maintenance: Challenges and Issues. *International Journal of Computer Science Engineering (IJCSE) Issues*, 1(1), 23-25.

Grottke, M., Matias, R., & Trivedi, K. S. (2008, November). The fundamentals of software aging. In *2008 IEEE International Conference on Software Reliability Engineering Workshops (ISSRE Wkspa)*, 1-6. IEEE.

Hallie, P., & Darlene, R. E. (2005). Building Evaluation Capacity Evaluation Models, Approaches, and Designs. SAGE Publications, Inc. City: Thousand Oaks.

Hair, J. F., Sarstedt, M., Hopkins, L., & G. Kuppelwieser, V. (2014). Partial least squares structural equation modeling (PLS-SEM) An emerging tool in business research. *European Business Review*, 26(2), 106-121.

Hair, J. F., Black W. C., Babin B. J., & Anderson, R. E. (2010). Multivariate Data Analysis (7th Ed). New Jersey: Prentice Hall.

Hajjar, S. T. (2018). Statistical analysis: Internal-consistency reliability and construct validity. *International Journal of Quantitative and Qualitative Research Methods*, 6(1), 46-57.

Heckmann, I., Comes, T., & Nickel, S. (2015). A critical review on supply chain risk– Definition, measure and modeling. Omega, 52, 119-132.

Henseler, J., Dijkstra, T. K., Sarstedt, M., Ringle, C. M., Diamantopoulos, A., Straub, D. W., & Calantone, R. J. (2014). Common beliefs and reality about PLS: Comments on Rönkkö and Evermann (2013). *Organizational research methods*, 17(2), 182-209.

Herzig, K., Just, S., Rau, A., & Zeller, A. (2013, November). Predicting defects using change genealogies. In *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE),* 118-127. IEEE.

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, 381-390.

Huo, S., Zhao, D., Liu, X., Xiang, J., Zhong, Y., & Yu, H. (2018, March). Using machine learning for software aging detection in Android system. In *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, 741- 746. IEEE.

Ian, S. (2011). Software Engineering, Addison-Wesley, Inc. Boston, USA.

Ibrahim, K. S. K., Yahaya, J. H., Mansor, Z., & Deraman, A. (2017). Towards the quality factor of software maintenance process: A review. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(3-4), 115- 118.

Isong, B., & Ekabua, O. (2013). Towards Improving Object-Oriented Software Maintenance during Change Impact Analysis. In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, p. 1.

Jean, C.L.C. 2005. Are Organisations Too Complex To Be Integrated In Technical Risk Assessment And Current Safety Auditing. *Safety Science*. 43 (5), 613–638.

Jia, S., Hou, C., & Wang, J. (2017). 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*, 1452-1456.

Jin, X. L., Zhou, Z., Lee, M. K., & Cheung, C. M. (2013). Why users keep answering questions in online question answering communities: A theoretical and empirical investigation. *International Journal of Information Management*, 33(1), 93-104.

Jönsson, P., & Lindvall, M. (2005). Impact analysis. In *Engineering and managing software requirements,* 117-142. Springer, Berlin, Heidelberg.

Jun, G., Bo, W., Yunsheng, W., Bin, Z., & Jiaojiao, W. (2011). Research of the software 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, 601-608. Springer, Berlin, Heidelberg.

Kamei, Y., Shihab, E., Adams, B., Hassan, A. E., Mockus, A., Sinha, A., & Ubayashi, N. (2013). A large-scale empirical study of just-in-time quality assurance. *IEEE Transactions on Software Engineering*, 39(6), 757-773.

KarimiAzari, A., Mousavi, N., Mousavi, S. F., & Hosseini, S. (2011). Risk assessment model selection in construction industry. *Expert Systems with Applications*, 38(8), 9105-9111.

Kaur, U., & Singh, G. (2015). A Review on Software Maintenance Issues and How to Reduce Maintenance Efforts. *International Journal of Computer Applications,* 118(1).

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, 230-237. IEEE.

Khatavakhotan, A. S., & Ow, S. H. (2012). An innovative model for optimizing software risk mitigation plan: A case study. In *Modelling Symposium (AMS)*, 2012 Sixth Asia, 220-224. IEEE.

Khatavakhotan, A. S., & Ow, S. H. (2012, May). An innovative model for optimizing software risk mitigation plan: A case study. In *2012 Sixth Asia Modelling Symposium*, 220-224. IEEE.

Kline, R. B. (2015). Principles and practice of structural equation modeling. 2nd Edition.New York: Guilford publications.

Knodel, J., & Naab, M. (2014). Mitigating the Risk of Software Change in Practice. *Interpretation*, 110, p. 01.

Kulkarni, P. (2015). Software Rejuvenation and Workload Distribution in Virtualized System. International Journal of Innovative Research in Computer and Communication Engineering, 3(6), 5966-5973.

Kumar, G., & Kaushik, M. (2016). Maintenance policies for improving the availability of a software-hardware system. In Reliability, Maintainability and Safety (ICRMS), 2016 11th International Conference, 1-5, IEEE.

Lahon, M., & Sharma, U. (2016). Risk assessment and mitigation approach for architecture evaluation in component based software development. In *Computing for Sustainable Global Development (INDIACom),3rd International Conference*, 2801-2804. IEEE.

Lamsweerde, V. A. (2001). Goal-oriented requirements engineering: A guided tour. In *Fifth IEEE International Symposium on Requirements Engineering Proceeding*, 249-262. IEEE.

Langner, F., & Andrzejak, A. (2013). Detecting software aging in a cloud computing framework by comparing development versions. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium,* 896-899.

Lehnert, S. (2011). A review of software change impact analysis. Ilmenau University of Technology, *Technical Report.*

Li, J., Qi, Y., & Cai, L. (2018). A Hybrid Approach for Predicting Aging-Related Failures of Software Systems. In *Service-Oriented System Engineering (SOSE), IEEE Symposium,* 96-105. IEEE.

Li, L., Vaidyanathan, K., & Trivedi, K. S. (2012). An approach for estimation of software aging in a web server. In *Empirical Software Engineering, Proceedings International Symposium*, 91-100. IEEE

Li, B., Sun, X., Leung, H., & Zhang, S. (2013). A survey of code-based change impact analysis techniques. *Software Testing, Verification and Reliability*, 23(8), 613-646.

Li, J., Stålhane, T., Kristiansen, J. M., & Conradi, R. (2010, September). Cost drivers of software corrective maintenance: An empirical study in two companies. In *2010 IEEE International Conference on Software Maintenance*, 1-8. IEEE.

Lightsey, B. (2001). Systems Engineering Fundamentals. Virginia: Defense Acquisition University Press. Retrieved from: http://www.dau.mil/pubs/pdf/SEFGuide%2001- 01.pdf . Accessed 5th December 2017

Liu, J., Zhou, J., & Buyya, R. (2015). Software rejuvenation based fault tolerance scheme for cloud applications. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference*, 1115-1118. IEEE.

Liu, S., Zhang, J., Liu, Y., & Chen, T. (2009, May). Evaluating and mitigating information systems development risk through balanced score card. In *2009 International Symposium on Information Engineering and Electronic Commerce*, 111-115. IEEE.

Lehman, M. M. (1996, October). Laws of software evolution revisited. In *European Workshop on Software Process Technology*, 108-124. Springer, Berlin, Heidelberg.

Lomax, R. G., & Schumacker, R. E. (2004). A Beginner's Guide to Structural Equation Modeling. New York: Psychology Press. 2nd Edition, p. 487.

Macêdo, A., Ferreira, T. B., & Matias, R. (2010). The mechanics of memory-related software aging. In *Software Aging and Rejuvenation (WoSAR), 2010 IEEE Second International Workshop*, 1-5. IEEE.

Machida, F., & Miyoshi, N. (2017). Analysis of an optimal stopping problem for software rejuvenation in a deteriorating job processing system. *Reliability Engineering & System Safety*, 168, 128-135.

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.

Machida, F., Xiang, J., Tadano, K., & Maeno, Y. (2012). Software life-extension: a new countermeasure to software aging. In *Software Reliability Engineering (ISSRE), IEEE 23rd International Symposium*, 131-140. IEEE.

Mahmud, H. (2017). A Simple Software Rejuvenation Framework Based on Model Driven Development. *UHD Journal of Science and Technology*, 1(2), 37-45.

Matias, R., Andrzejak, A., Machida, F., Elias, D., & Trivedi, K. (2014). A systematic differential analysis for fast and robust detection of software aging. In *Reliable Distributed Systems (SRDS), IEEE 33rd International Symposium*, 311-320. IEEE.

Matias, R., Beicker, I., Leitão, B., & Maciel, P. R. (2010). Measuring software aging effects through OS kernel instrumentation. In *Software Aging and Rejuvenation (WoSAR), 2010 IEEE Second International Workshop*, 1-6. IEEE.

Meng, H., Hei, X., Li, Y., Du, Y., & Xie, G. (2015). A Rejuvenation Model for Software System under Normal Attack. In *2015 IEEE Trustcom/BigDataSE/ISPA*, 1, 1160-1164. IEEE.

Meng, H., Hei, X., Zhang, J., Liu, J., & Sui, L. (2016). Software Aging and Rejuvenation In A J2ee Application Server. *Quality and Reliability Engineering International,* 32(1), 89-97.

Meng, H., Liu, J., & Hei, X. (2015). Modeling and optimizing periodically inspected software rejuvenation policy based on geometric sequences. *Reliability Engineering & System Safety*, 133, 184-191.

Meng, H., Zhang, X., Zhu, L., Wang, L., & Yang, Z. (2017). Optimizing software rejuvenation policy based on CDM for cloud system. In *Industrial Electronics and Applications (ICIEA), 2017 12th IEEE Conference*, 1850-1854. IEEE.

Meyers, L. S., Gamst, G., & Guarino, A. J. (2016). Applied multivariate research: Design and interpretation. Sage publications.

Mukhlash, I., Maulidiyah, R., & Setiyono, B. (2017, September). Web-based decision support system to predict risk level of long-term rice production. In *Journal of Physics: Conference Series,* 890(1), p. 012143. IOP Publishing.

Naeem, M. R., Zhu, W., Memon, A. A., & Khalid, A. (2014, December). Using V-Model methodology, UML process-based risk assessment of software and visualization. In *Proceedings of 2014 International Conference on Cloud Computing and Internet of Things*, 197-202. IEEE.

Narania, S., Eshahawi, T., Gindy, N., Tang, Y. K., Stoyanov, S., Ridout, S., & Bailey, C. (2008). Risk mitigation framework for a robust design process. In *Electronics System-Integration Technology Conference*, 2008. ESTC 2008. 2nd, 1075-1080.

Nasir, Z., & Abbasi, A. Z. (2010, June). A framework for software maintenance and support phase. In *2010 International Conference on Information and Emerging Technologies*, 1-6. IEEE.

Nguyen, V., Boehm, B., & Danphitsanuphan, P. (2011). A controlled experiment in assessing and estimating software maintenance tasks. *Information and software technology*, 53(6), 682-691.

Ning, G., Zhao, J., Lou, Y., Alonso, J., Matias, R., Trivedi, K. S., & Cai, K. Y. (2016). Optimization of two-granularity software rejuvenation policy based on the Markov regenerative process. *IEEE Transactions on Reliability,* 65(4), 1630-1646.

Nurmuliani, N., Zowghi, D., & Williams, S. (2006). Requirements volatility & its impact on change effort: Evidence based research n software development projects. In Verified OK. University of South Australia.

O'Connor, H., & Gibson, N. (2003). A step-by-step guide to qualitative data analysis. Pimatisiwin: A *Journal of Indigenous and Aboriginal Community Health*, 1(1), 63-90.

Ogheneovo, E. E. (2014). On the relationship between software complexity and maintenance costs. *Journal of Computer and Communications*, 2(14), p. 1.

Okamura, H., & Dohi, T. (2013). Dynamic software rejuvenation policies in a transaction-based system under Markovian arrival processes. *Performance Evaluation*, 70(3), 197-211.

Okamura, H., Zheng, J., & Dohi, T. (2017). Statistical Framework on Software Aging Modeling with Continuous-Time Hidden Markov Model. In *Reliable Distributed Systems (SRDS), 2017 IEEE 36th Symposium,* 114-123. IEEE.

Orlando, L. A., Wu, R. R., Myers, R. A., Buchanan, A. H., Henrich, V. C., Hauser, E. R., & Ginsburg, G. S. (2016). Clinical utility of a Web-enabled risk-assessment and clinical decision support program. *Genetics in Medicine*, 18(10), 1020-1028.

Pa, N. C., Jnr, B. A., Jusoh, Y. Y., Haizan, R. N., & Nor, T. N. M. A. (2017). A risk mitigation decision framework for information technology organizations. *Journal of theoretical and applied information technology,* 95(10), 2102-2113.

Pa, N. C., & Anthony, B. (2015, August). A Model Of Mitigating Risk For IT Organisations. In *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, 49-54. IEEE.

Paetsch, F., Eberlein, A., & Maurer, F. (2003, June). Requirements engineering and agile software development. In WET ICE 2003. *Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises,* 308-313. IEEE.

Pallant, J. (2016). SPSS survival manual: A step by step guide to data analysis using IBM SPSS. Routledge.

Parnas, D. L. (1994). Software aging. *Proceedings of 16th International Conference on Software Engineering,* 279-287. IEEE.

Porter, A. A. (1997). Fundamental laws and assumptions of software maintenance. *Empirical Software Engineering*, 2(2), 119-131.

Qin, F., Zheng, Z., Bai, C., Qiao, Y., Zhang, Z., & Chen, C. (2015). Cross-Project Aging Related Bug Prediction. In *Software Quality, Reliability and Security (QRS), IEEE International Conference*, 43-48. IEEE.

Qin, F., Zheng, Z., Qiao, Y., & Trivedi, K. S. (2018). Studying Aging-Related Bug Prediction Using Cross-Project Models. *IEEE Transactions on Reliability*, 68(3), 1134-1153.

Rahman, M. A., Razali, R., & Ismail, F. F. (2019). Risk Factors for Software Requirements Change Implementation. *International Journal of Advanced Computer Science and Applications*, 10(3), 133-139.

Rahman, M. A., Razali, R., & Singh, D. (2014). A Risk Model of Requirements Change Impact Analysis. *Journal of Software*, 9(1), 76-81.

Rahme, J., & Xu, H. (2017). Preventive maintenance for cloud-based software systems subject to non-constant failure rates. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI),* 1-6. IEEE.

Rejab, M. M., Chuprat, S., & Azmi, N. F. M. (2018). A Review for Improving Software Change using Traceability Model with Test Effort Estimation. *International Journal of Academic Research in Business and Social Sciences,* 8(4), 1198-1208.

Russo, S. (2014). The Dual Nature of Software Aging: Twenty Years of Software Aging Research. In *2014 IEEE International Symposium on Software Reliability Engineering Workshops*. 431-432. IEEE.

Robillard, M. P., & Dagenais, B. (2008, October). Retrieving task-related clusters from change history. In *2008 15th Working Conference on Reverse Engineering* ,17-26. IEEE.

Roya, O., Aye, Z. C., & Jaboyedoff, M. (2013). Development Of A Prototype For Spatial Decision Support System In *Risk, Reduction Based On Opensource Web- Based Platform*. In GEOMUNDUS 2013, Castellon de la Plana Spain, 1-6.

Salfner, F., & Wolter, K. (2010). Analysis of service availability for time-triggered rejuvenation policies. *Journal of Systems and Software*, 83(9), 1579-1590.

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.

Sekaran, U., & Bougie, R. (2003). Research Methods For Business, A Skill Building Approach, John Willey & Sons. Inc. New York.

Shahid, M., & Ibrahim, S. (2016, January). Change impact analysis with a software traceability approach to support software maintenance. In *2016 13th International Bhurban conference on applied sciences and technology (IBCAST),* 391-396.

Shahzad, B., Al-Ohali, Y., & Abdullah, A. (2011). Trivial model for mitigation of risks in software development life cycle. *International Journal of Physical Sciences*, 6(8), 2072-2082.

Sharma, V. S., Ramnani, R. R., & Sengupta, S. (2014, June). A framework for identifying and analyzing non-functional requirements from text. In *Proceedings of the 4th international workshop on twin peaks of requirements and architecture*, 1-8. ACM.

Shihab, E., Hassan, A. E., Adams, B., & Jiang, Z. M. (2012, November). An industrial study on the risk of software changes. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, 1-11. ACM.

Singh, Y., & Goel, B. (2007). A step towards software preventive maintenance. *ACM SIGSOFT Software Engineering Notes*, 32(4), p. 10.

Singh, B., Sharma, K. D., & Chandra, S. (2012) A new model for software risk management. *International Journal of Computer Technology and Applications*, 3(3), 953-956.

Sinha, R. P., Whitman, L. E., & Malzahn, D. (2004). Methodology to mitigate supplier risk in an aerospace supply chain. *Supply Chain Management: an international journal*, 9(2), 154-168.

Sipayung, J. J. P., & Sembiring, J. (2015, November). Risk assessment model of application development using Bayesian Network and Boehm's Software Risk Principles. In *2015 International Conference on Information Technology Systems and Innovation (ICITSI),* 1-5. IEEE.

Sommerville, I. (2009). Software engineering (Ninth Edition). Pearson education. Straub, D. W. (1989). Validating instruments in MIS research. *MIS quarterly*,147-169.

Sun, X., Huang, M., & Wang, X. (2011, May). Web and multi-agent based virtual enterprise risk management system. In *2011 Chinese Control and Decision Conference (CCDC)*, 902-906. IEEE.

Sun, X., & Li, B. (2011, November). Using Formal Concept Analysis to support change analysis. In *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011),* 641-645. IEEE.

Spaulding, D. T. (2013). Program Evaluation in practice: Core concepts and examples for discussion and analysis. John Wiley & Sons. San Francisco, CA: Jossey-Bass.

Stojanov, Z., Dobrilovic, D., & Stojanov, J. (2018). Extending data-driven model of software with software change request service. *Enterprise Information Systems*, 12(8-9), 982-1006.

Tamjidyamcholo, A., Baba, M. S. B., Shuib, N. L. M., & Rohani, V. A. (2014). Evaluation model for knowledge sharing in information security professional virtual community. Computers & Security, 43, 19-34.

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. *International Journal of Medical Education*, 2, 53-55.

Tao, Y., Dang, Y., Xie, T., Zhang, D., & Kim, S. (2012, November). How Do Software Engineers Understand Code Changes?: An Exploratory Study In Industry. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, 1-11. ACM.

Thaheem, M. J., De Marco, A., & Barlish, K. (2012, June). A review of quantitative analysis techniques for construction project risk management. In *Proceedings of the Creative Construct Conference*, 656-667.

Tombe, R., Okeyo, G., & Kimani, S. (2014). Cyclomatic Complexity Metrics for Software Architecture Maintenance Risk Assessment. *International Journal of Computer Science and Mobile Computing*, 3(11), 89-101.

Tonny, B., Pa, N., Khalefa, M., Alasadi, H., & Zmezm, H. (2016). A Proposed Risk Assessment Model For Decision Making In Software Management. *Journal of Soft Computing and Decision Support Systems (JSCDSS),* 3(5), 31-43.

Tripathy, P., & Naik, K. (2014). Software Evolution and Maintenance: A Practitioner's Approach. John Wiley & Sons.

Trümper, J., Beck, M., & Döllner, J. (2012). A visual analysis approach to support perfective software maintenance. In *2012 16th International Conference on Information Visualisation*, 308-315.

Umesh, I. M., & Srinivasan, G. N. (2016). Optimum Software Aging Prediction and Rejuvenation Model for Virtualized Environment. *Indonesian Journal of Electrical Engineering and Computer Science,* 3(3), 572-578.

Umesh, I.M., Srinivasan, G.N. and Torquato, M. (2017). Software Rejuvenation Model for Cloud Computing Platform. *International Journal of Applied Engineering Research,* 12(19), 8332-8337.

Urbach, N., & Ahlemann, F. (2010). Structural equation modeling in information systems research using partial least squares. *Journal of Information technology theory and application*, 11(2), 5-40.

Verner, J., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2012). Risk Mitigation Advice for Global Software Development from Systematic Literature Reviews. School of Computing and Mathematics, Keele University, Keele, Staffordshire, UK.

Wong, K. K. K. (2013). Partial Least Squares Structural Equation Modeling (PLS-SEM) techniques using SmartPLS. *Marketing Bulletin*, 24(1), 1-32.

Williams, B. J., Carver, J., & Vaughn, R. B. (2006). Change Risk Assessment: Understanding Risks Involved in Changing Software Requirements. In *Software Engineering Research and Practice*, 966-971.

Williams, B. J., & Carver, J. C. (2010). Characterizing software architecture changes: A systematic review. *Information and Software Technology*, 52(1), 31-51.

Wu, H., & Wolter, K. (2015). Software aging in mobile devices: Partial computation offloading as a solution. In *Software Reliability Engineering Workshops (ISSREW), 2015 IEEE International Symposium*, 125-131. IEEE.

Xiang, J., Weng, C., Zhao, D., Tian, J., Xiong, S., Li, L., & Andrzejakb, A. (2018). A New Software Rejuvenation Model for Android. In *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 293-299. IEEE.

Xiaoping, W., Fang, Z., & Yi, S. (2014, July). An optimal software maintenance policy based on reliability and risk. In *2014 International Joint Conference on Neural Networks (IJCNN)*, 3043-3048. IEEE.

Yahaya, J. H., Abidin, Z. N. Z., Ali, N. M., & Deraman, A. (2013). Software ageing measurement and classification using Goal Question Metric (GQM) approach. In *Science and Information Conference (SAI),* 160-165.

Yahaya, J. H., Deraman, A., & Abdullah, Z, H. (2016). Evergreen software preservation: The anti ageing model. *Proceedings of the International Conference on Internet of things and Cloud Computing,* p.51. ACM.

Yahaya, J. H., Abidin, Z. N. Z., & Deraman, A. (2015, July). Perspective and perception on software ageing: The empirical study. In *2015 10th International Conference on Computer Science & Education (ICCSE),* 365-370. IEEE.

Yahaya, J., & Deraman, A. (2012). Towards the anti-ageing Model for Application software. In *Proceedings of the World Congress on Engineering,* 2, 388-393.

Yiu, L. and Lee, M. J. (2014). An effective dynamic programming offloading algorithm in mobile cloud computing system. In *Wireless Communications and Networking Conference (WCNC),* 1868–1873, IEEE.

Yan, Y., & Guo, P. (2016). A practice guide of software aging prediction in a web server based on machine learning. *China Communications,* 13(6), 225-235.

Zambon, E., Bolzoni, D., Etalle, S., & Salvato, M. (2007, May). Model-based mitigation of availability risks. In *2007 2nd IEEE/IFIP International Workshop on Business- Driven IT Management*, 75-83. IEEE.

Zeng, S. X., Tam, C. M., & Tam, V. W. (2010). Integrating safety, environmental and quality risks for project management using a FMEA method. *Engineering Economics*, 66(1), 44-52.

Zhai, Y. Z., Li, Q. Y., & You, H. C. (2017). Software Health Measurement Method Based on Aging-related Bugs. In *2017 2nd International Conference on Computational Modeling, Simulation and Applied Mathematics (CMSAM 2017),* 475-480.

Zhao, J. F. (2016). Modeling of Software Aging Based on Non-stationary Time Series. *In Information System and Artificial Intelligence (ISAI), 2016 International Conference*, 176-180. IEEE.

Zhao, J., Wang, Y., Ning, G., Trivedi, K. S., Matias Jr, R., & Cai, K. Y. (2013). A comprehensive approach to optimal software rejuvenation. *Performance Evaluation*, 70(11), 917-933.

Zheng, J., Okamura, H., Li, L., & Dohi, T. (2017). A Comprehensive Evaluation of Software Rejuvenation Policies for Transaction Systems With Markovian Arrivals. *IEEE Transactions on Reliability*, 66(4),1157-1177.

Zheng, P., Xu, Q., & Qi, Y. (2012). An advanced methodology for measuring and characterizing software aging. In *Software Reliability Engineering Workshops (ISSREW), 2012 IEEE 23rd International Symposium*, 253-258. IEEE.