Global Journal of Engineering Science and Research Management

A SURVEY ON AUTOMATED TOOL FOR SOFTWARE QUALITY IMPROVEMENT

G Pandiyan[*1] **Dr P Krishnakumari**[2]
[1*]Research Scholar, RVS College of Arts and Science, Coimbatore.
[2]Director, Department of MCA, RVS College of Arts and Science, Coimbatore.
*Correspondence Author: **G Pandiyan**

Keywords: Software Engineering, Data Quality, Fraud Prediction, Security, Software Measurements.

Abstract

Many software organizations today are confronted with challenge of building secure software systems. Traditional software engineering principles place a wide importance on security. Software principles tend to outline security as one of a long list of quality aspect with the hope of high efficiently developed software. As software systems of today have a broad reach, security has become a challenge. An essential need is developed to adopt security into software engineering. Software engineering mainly focused on structure of process, methods and tools which also considers other constraints to enhance and sustains the software process in appropriate manner. The challenge of Software standards lies in high security and maintenances with lengthy list of quality issues and probability of highly improved software. The conventional software engineering ideologies are settled with minimum importance on security. Data quality in software engineering defines attributes with various specification of context to present better software information process. Software fault prediction replicas are utilized to discover the fault levels in various components which lead to high-principled software. The attainment of the fault prediction replicas is connected with existing software measurements and fault information to determine the threads. Security factors turn into a huge precedence process in software engineering and software systems. Moreover, security on online networking process is gradually increasing for protecting the software issues. The necessity of better quality information for the users demands the improvising of software engineering. Experimental evaluation on software quality with parameters like fault prediction rate, software failure rate, software protection cost prove the efficiency.

Introduction

Software engineering is the formation of data which useful in emerge of effectual engineering standards. The formation of data sequentially attains efficient software products with trustworthy and effective technology. Software engineering intends in generation of high quality data with reliable techniques for fast and cost-effective manner. In general, software engineering uses low-priced and long term consumption for effective outcomes with enhanced quality data.

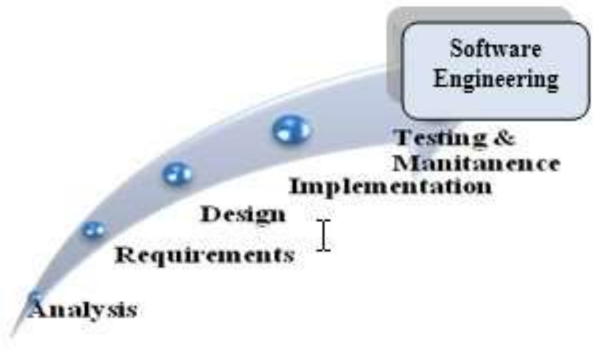

Fig 1.1 Software Engineering Model

Above figure 1.1 describes the standard software engineering models. Initial stage of software engineering involves analysis of products. After which user requirements are gathered and in turn based on needs the design is made. Followed by implementation and checks the coding in testing phase as a final result. Most of the software process is mainly based on the software quality and software maintenances. In [1], the Adaptive Behavior technique is used in composed software scheme for preserving the effective information. Adaptive Behavior technique depends on information in every module where developer faces certain concerns like collision of possible variation on the key performance indicators (KPIs).

In order to tackle problems on collision, a self-adaptive system is presented in [4]. More specifically, self-adaptive system is efficient for constant varying network. And also the technique ensures secure conditions with unidentified data in between transmission rather at destinations. A self-adaptive system helps the methodical and well-ordered data for self-organizing software process.

In [5], reduce some of the quality collision found in Web Engineering development projects with evaluation of quality attributes from initial phases of the software development task. The requirement Meta model concentrates on the addition of the conceptual models used by Web Engineering methodologies with the objective of permitting the clear concern of usability requirements.
The GenProg in [2] is an automatic technique for repairing faults in the software engineering. Mostly used in enlarged structure of genetic training to develop schedule variances that maintains the connection strategies. But the problem is more vulnerable to specified faults which set encryption process for defects and needs functionality methods.

In order to overcome these difficulties, the general tendency for finding the software defects and the failure information are analyzed in [6]. The defects and failure information particularly in data elicitation modifies tracking methods with the real entity software development.

The systematic appraisals on agile software improvements are monitored in recognition of agile software enhancement. In addition agile software requires normal explore schedule for connecting software process with minimum defects and also constructs the general perceptive challenge in [7] agile software.

Even though the faults were defected in software process, security constraints need additional focus. The structure for security constraints are based on creation of various circumstance with some schemes that are evaluated with security requirements in [9]. Additionally, improves the protection process. The system conditions are explained with the problem oriented information, after which they are authenticated with security constraints and the protection process.

By this various discussion, a discussion on software quality improvement with fault prediction techniques. The main complications are analyzed and provide reliable methods to enhance the data quality by fault prevention and secure process.

Fault prediction

Software fault prediction is one of the valuable assertion behaviors in software engineering to formulate the proper authentication, defects, exploration and inspect faults. The software metrics and defects or non-defects data are applied in earlier to software translation. In general, software translations are utilized to generate the prediction replicas. Software fault prediction analyzes products related to fault prediction in order to improve the software quality. Fault prediction is partitioned into four groups namely Self-management of Adaptable Component (SAC), Genetic Program Repair (GenProg) methods, Attack Surface Metric (ASM) methods and Multiparty Access Control for Online Social Networks (MPAC-OSN) models.

Self-management of Adaptable Component (SAC) Models

The SAC models are concern with the trouble of selection in choosing suitable individual modules for adaptation to tackle with variation from the structure of optimal attributes. Here SAC [1], faces various challenges to present better software data quality. These challenges determine the impact of component adaptation in structural behavior. Whereas the technique in [4] provides the state-of-art for identifying the crucial disputes which improvise the task of software development with self-adaptive techniques. Self-adaptive techniques is separated into four methods to present better individual adaptive components with specific assurances, constraints, replicas, and information size to fulfill software development process.

For perfect adaptation and prediction of software development, quality improvements are more essential and needs open source software development (OSSD) model as in [8]. OSSD are emerged with appropriate process for users to provide better results with the combination of agile and plan-determination techniques. The combined techniques improve the data quality by removing the extra information from the software engineering techniques as elaborated in [10].

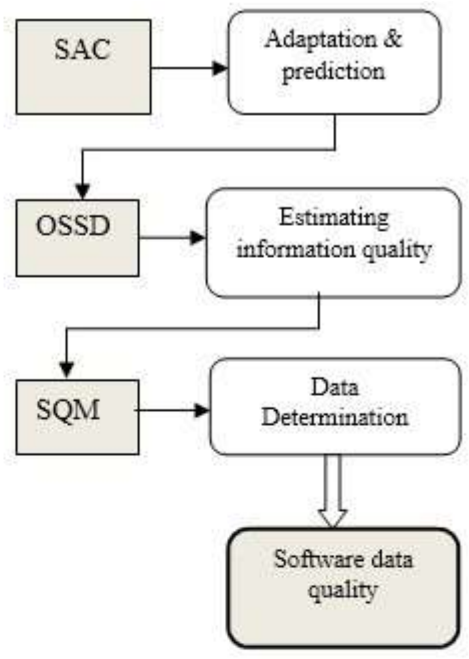

Fig 2.1 Overall Performances of SAC

Above figure 2.1 depicts the process involved in SAC, OSSD and SQM. SAC performs adaptation as well as prediction, OSSD estimates data quality while SQM determinates data. All together provides better software data quality. Specifically, Utilizing data information from code quality determines the effective software quality metrics (SQM). In addition SQM specifies mining schemes to consume automatic quality metrics from the temporal security conditions to prevent the false positive process.

➢ ***Key Performance Indicators (KPIs)***

KPIs permit the advanced information for the system performances which intends in prediction of low-level administration aspects. Additionally evaluates the individual module or complete structure for facilitating the KPI structures. KPI description includes name, kind of predictable value, and significance margin evaluation for combination of global values.

Genetic Program (GenProg) Methods

Genetic Program (GenProg) is a method utilized for obtaining the testing method which automatically produces preservation for real-world bugs in inedible-shelf. Additionally includes legacy appliances. The GenProg method is unbeaten on creation of an alternative passes for encryption with various performance and does not failure with those encrypted bugs.

➢ ***Repair Minimization (RM)***

In GenProg, RM executes minimization by recognizing dissimilarities among the major repair and unique schedules. GenProg removes difference which does not concern the repair's attributes on some of test cases. The minimization method discovers a separation of early repair alterations which does not drop the elements for getting failure in test cases.

➢ ***Repair Descriptions (RD)***

The RD authenticates the automatic repair of real-world faults which is found more on various buggy schedules and patterns of each patch seen in GenProg. The repairs are executed in generic programs that produce the better quality of data without bugs in software engineering.

A common structure for predicting the faults and to improve the data quality of software process is discussed in [13] with development of rival prediction process. The software defect prediction maintains the balanced and complete evaluation among the rival prediction process. The defect analyst constructs the replicas according to the estimated learning method and forecast software faults with information for constructed models.

Moving with reverse engineering strategy, they are merged with genetic exploration of static and dynamic scrutiny to predict the fault appliances. The genetic applications utilizes the recreation process for attributes replicas from observed data, runtime byte code calculation and static byte code scrutiny as in [14].

In [6], [7] explores the amount of records modified for avoiding the malfunctions from redistribution process. The redistribution process separates the various records for preventing the bug's software techniques. In [15], the method is useful to discover the predictions of bugs known as change classification. The change classification forecasts the subsistence and the faults in software modifications.

Attack Surface Metric (ASM) methods

The attack surface metrics is more useful in the software system's attack dimension as a pointer for the classification protection. The awareness of a system attack surfaces are initiated with the attack surface metrics to calculate the attacks in the efficient approach as in [8].

Attack Surface Reduction Schemes

The attack surface reduction methods are balanced with their software conventional code quality development approach for defense risk improvements [8]. The safeguard metrics and measurements are necessary for protecting the software improvement. The evaluation of cryptographic algorithms separates the present fault injection attacks with minimum outlays and extremely accomplished attackers with a huge budget. After emerge of failures to some extent, the attacks decreases by using common ciphers which indicate the fault prevention process [16]. But in [17], particular attacks arise by liabilities in packet dispensation software which commence with the overwhelming denial-of-service (DoS) attacks. DoS attack is prevented by certain different techniques which offer the better quality improvement.

The system consistency and system protection approaches are processed under the balanced attacks. In particular, we examine how protection and attack approaches are might impact with system consistency when both the protector and aggressor are specified with a fixed quantity of resources. The cyber securities are given by protector or by generally choosing the division of modules to attack by attackers [18].

Multiparty Access Control for Online Social Networks (MPAC-OSN) models

An online social network (OSNs) holds proper knowledgeable with remarkable growth and becomes effective gateway for huge Internet users. These OSNs present attractive methods for digital social connections and data distribution which also increase an amount of security and isolation concerns.

The security issues for information management are crucial to safe guard their intellectual resources. So, only approved persons are accepted to perform the several procedure and purpose in an organization. In [19], supports an secured information management with the spotlighting of privacy, expectation and isolation.

Since the stimulation of OSNs provides protection and confidentiality for the users with the best effective manner. The attractive aspects of conventional OSNs are presented with the possible information to the users with the specified contacts or any social interactions among the various users [20].

Even though OSNs provides better safe guard process, recognition of attacks is not possible. Some of the detection dimension process for identifying the attacks needs further investigation. On the other hand, OSNs only utilized malice-aware routing for attack detection which is less effective facing various attacks [21]. Using the key-based routing (KBR) methods also slightly reduce the malicious attacks but needs extra attention in unknown attacks activities especially in the online social networks.

Performance evaluation

The performance evaluations are conduct with various metrics on the examination of data quality in software process. Analyze the software data quality in terms of fault prediction, software failure rate and software protection cost.

Fault Prediction Rate

The fault predictions are utilized to diminish the defects in software engineering to improve the software quality. The fault prediction rate is defined as

$$\text{Fault prediction rate} = \frac{\sum MSE + MAE + SR}{\sum False\ positive\ rate}$$

Where MSE denotes mean Square error (MSE), MAE indicates Mean absolute error, and SR represents Software regression.

Software Failure Rate
The software failure rate calculates the defects in software data quality for reducing the failures. Software failure rate is used to reduce the failure rates in order to enhance the software data quality. The software failure rate is estimated by Length of exposure in a particular fault (LEF) and dispersion of faults (DF) and also failure time. The software failure rate are illustrated below

$$\text{Software Failure Rate} = \frac{\sum LEF + DF}{\sum Failure\ time}$$

Software Protection Cost
The software protection cost estimates the computation time, product size and also cost estimation for enhancing the better software products.

$$software\ protection\ cost = \frac{\sum SCT + PS + CE}{\sum\ No.of\ software\ products} \times 100$$

Here, SCT represents the software computation time, PS denotes Product size and CE is cost estimation

Experimental results
The software data quality improvement for software engineering process is analyzed with Self-management of Adaptable Component (SAC), Genetic Program Repair (GenProg) methods, Attack Surface Metric (ASM) methods and Multiparty Access Control for Online Social Networks (MPAC) models. The performances of fault prediction are compared with above all models and techniques.

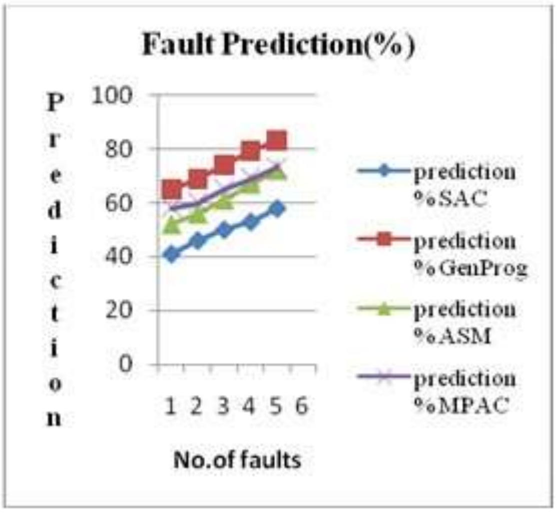

Fig 4.1 Fault Prediction in Software Data Quality

Fig 4.1 defines about the fault prediction based on the software data quality improvements which is calculated in terms of percentage. As per the graph in figure 4.1 GenProg increases the prediction rate. The experimental evaluation shows that the GenProg are gradually increased when compared to other techniques like SAC, ASM and MPAC. Similarly the fault prediction is 15-20% higher when compared to SAC, ASM 10-14 % is higher in MPAC.

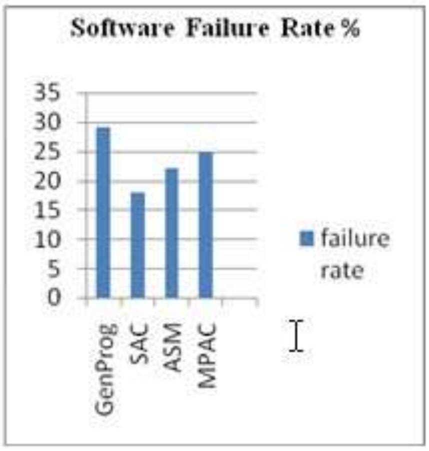

Fig 4.2 Software Failure Rate

Above Fig 4.2 describes the failure rate based on the software data quality developments and estimated in the terms of percentage. The failure rate is decreased to a great extent in SAC. The experimental evaluation illustrates that the SAC are gradually decreased when compared to GenProg, ASM and MPAC. Similarly the failure rate is 10-13% higher when compared to ASM, whereas MPAC 10-14 % is higher in GenProg.

Conclusion

The software engineering needs a better enhancement more specifically in improving data quality. Hence, the data quality is possibly increased with more effective process of time minimization as well as faults reduction. Both the time minimization and fault information reduction highly improves the software quality. Here we discussed about various parameter metrics like software failure rate, fault prediction and software protection cost to improve the software data quality methods.

This paper conveys the essential needs of data quality with emerge of data module techniques for providing effective software information quality. Even though, present techniques provide sufficient data quality, the fault prediction process is unnoticed or less focused. In real time applications, a social network requires more secures information for users in software engineering. By this observation of various techniques and methods, the demands on high quality fault-free software engineering is in great extent.

References

1. *Liliana Rosa., Lu´ıs Rodrigues., Ant´onia Lopes., Matti Hiltunen., Richard Schlichting ., "Self-management of Adaptable Component-based Applications," TRANSACTIONS ON SOFTWARE ENGINEERING., 2011*
2. *Claire Le Goues., ThanhVu Nguyen., Stephanie Forrest., and Westley Weimer., "GenProg: A Generic Method for Automatic Software Repair," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 1, JANUARY/FEBRUARY 2012*
3. *Hongxin Hu., Gail-Joon Ahn., and Jan Jorgensen., "Multiparty Access Control for Online Social Networks: Model and Mechanisms," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 2013*
4. *Betty H.C. Cheng, Rog´erio de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee "Software Engineering for Self-Adaptive Systems: A Research Roadmap" Springer-Verlag Berlin Heidelberg 2009 pp. 1–26,*
5. *Fernando Molina., Ambrosio Toval., "Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems," Advances in Engineering Software., Elsevier Journal., 2009*
6. *Maggie Hamill., Katerina Goseva-Popstojanova., "Common Trends in Software Fault and Failure Data," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. 4, JULY/AUGUST 2009*
7. *Tore Dyba., Torgeir Dingsoyr., "Empirical studies of agile software development: A systematic review," Information and Software Technology, Elsevier Journal., (2008)*
8. *Chitu Okoli, Kevin Carillo "The best of adaptive and predictive methodologies: Open source software development, a balance between agility and discipline" International Journal of Information Technology and Management Volume 11, Number 1-2/2012, Pages 153-166*

9. *Charles B. Haley., Robin Laney., Jonathan D. Moffett., and Bashar Nuseibeh., "Security Requirements Engineering: A Framework for Representation and Analysis," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 34, NO. 1, JANUARY/FEBRUARY 2008*
10. *Claire Le Goues, Westley Weimer "Measuring Code Quality to Improve Specification Mining" Software Engineering, IEEE Transactions on (Volume:38 , Issue: 1) 2012 Page(s): 175 – 190*
11. *Charles B. Haley., Robin Laney., Jonathan D. Moffett., and Bashar Nuseibeh., "Security Requirements Engineering: A Framework for Representation and Analysis," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 34, NO. 1, JANUARY/FEBRUARY 2008*
12. *Yuriy, Burn., Giovanna Di Marzo Serugendo., Cristina Gacek., Holger Giese., Holger kienle., Marin Litoiu., Hausi M'uller., Mauro Pezze., and Mary Shaw., "Engineering Self-Adaptive Systems through Feedback Loops," Springer-Verlag Berlin Heidelberg 2009*
13. *Qinbao Song, Zihan Jia, Martin Shepperd, Shi Ying and Jin Liu "A General Software Defect-Proneness Prediction Framework" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. X, NO. X, 2010*
14. *Klaus Krogmann, Michael Kuperberg, and Ralf Reussner, "Using Genetic Search for Reverse Engineering of Parametric Behaviour Models for Performance Prediction" Software Engineering, IEEE Transactions on (Volume:36 , Issue: 6) 2010 Page(s): 865 – 877*
15. *Sunghun Kim; Whitehead, E.J.; Yi Zhang "Classifying Software Changes: Clean or Buggy?" Software Engineering, IEEE Transactions on (Volume:34 , Issue: 2) March-April 2008 Page(s): 181 – 196*
16. *Barenghi, A. ; Breveglieri, L. ; Koren, I. ; Naccache, D. "Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures" IEEE international conferences (Volume:100 , Issue: 11) Nov. 2012 Page(s):3056 – 3076*
17. *Danai Chasaki, and Tilman Wolf "Attacks and Defenses in the Data Plane of Networks" IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), VOL. XX, NO. YY, MONTH 2012*
18. *Li Wang, Shangping Ren, Bogdan Korel, Kevin Kwiat, and Eric Salerno "Improving System Reliability against Rational Attacks under Given Resources" Systems, Man, and Cybernetics: Systems, IEEE Transactions on (Volume:44 , Issue: 4) April 2014 Page(s): 446 – 456*
19. *Elisa Bertino, Latifur R. Khan, Ravi Sandhu, and Bhavani Thuraisingham "Secure Knowledge Management: Confidentiality, Trust, and Privacy" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 36, NO. 3,2007*
20. *Chi Zhang ; Jinyuan Sun ; Xiaoyan Zhu ; Yuguang Fang "Privacy and Security for Online Social Networks: Challenges and Opportunities" Network, IEEE (Volume:24 , Issue: 4) July-August 2010 Page(s): 13 – 18*
21. *Yan Sun ; Yuhong Liu "Security of online reputation systems: The evolution of attacks and defenses" Signal Processing Magazine, IEEE (Volume:29 , Issue: 2) March 2012 Page(s): 87 – 97*