International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 2
ISSN: 2454-4248
330 – 336

Various Techniques to Detect and Predict Faults in Software System: Survey

Rajdeep Kaur[1], Er Sumit Sharma[2]
M.Tech (Scholar), Assistant Professor
Department of Computer Science Engineering
Chandigarh University
Email: chahal.rajdeep03@gmail.com , cu.sumitsharma@gmail.com

***Abstract* –** Software reliability is frequently estimated as per occurrence of failure and its recurrence, or as far as framework accessibility. Software is often categorized as system software (such as operating system, or program that assists application software etc.), application software (such as spread sheets, CAD / CAM, graphic software, etc.), shareware (generally planned for sale after trial duration), literate (available with few conditioned abilities), freeware, public domain or open source Almost, all frameworks have tendency to have faults however not each of them result in failure. Software fault prediction (SFP) approaches can be used to predict the faults. The objective of the fault prediction is to reduce errors and its influence before failure happens. In particular cases, where failure has just happened, fast recovery is major task. This review provide overview about the software, faults, fault prediction, detection and prevention strategies rehearsed and examined. Benefits and drawbacks of those mechanisms are also discussed. Several previous work is reviewed and compared with respect to used techniques and performances parameters.

***Keywords* –** *Software failure, Fault prediction, detection and prevention method.*

*****

I. INTRODUCTION

The Software is commonly known as computer programs or the set of instructions which allows users to interact with computer and its hardware to perform various tasks. Software is often categorized as system software (such as operating system, or program that assists application software etc.), application software (such as spreadsheets, CAD / CAM, graphic software, etc.), shareware (generally planned for sale after trial duration), literate (available with few conditioned abilities), freeware, public domain or open source [1].

Software failures are the uncertain results produced due to several specific surroundings or condition that creates defaults in an application while executing. All faults aren't results in failures [2]. Failures occur due to many reasons:

- Surrounding conditions like strong magnetic or electronic field, contamination or dirtiness in hardware.
- Human interaction with software might be wrong input values or misread output values.
- Intentionally performing operation to break the system.

Error, Fault and Failures are the different aspects [3]:
Failure: Incapacity of system to perform task required as per specified. It's an incorrect external behaviour.
Fault: A situation influencing software to perform desired operation efficiently. Variation in code causes failures.
Error: Contrast among the expected results and results received. It's a human mistake occurs at developer's end.
As per the IEEE standards, Error is a manual action leads to incorrect results. Fault is wrong conclusion while understanding the problem. Only one error results in fault and many faults results in failure [4]. Fault detection processes are initiated in each phase of software development to avoid failure depending upon their requirement and criticalness.

Software system plays very important role in our daily life. Developing defect free software is the major responsibility of the developers. Due to some fault prone modules or parts of software system, development team may suffer from some extra cost or time problems [2]. Because of its significant role, it should be of higher quality. Software reliability is the major expectation from the high quality software.

Software fault prediction is the process of predicting the faults at initial stages. Several fault prediction approaches are used to predict the faults in software system. These approaches use past data related to faults and software metrics to predict the faulty modules [5].

 ISSN: 2454-4248
330 – 336

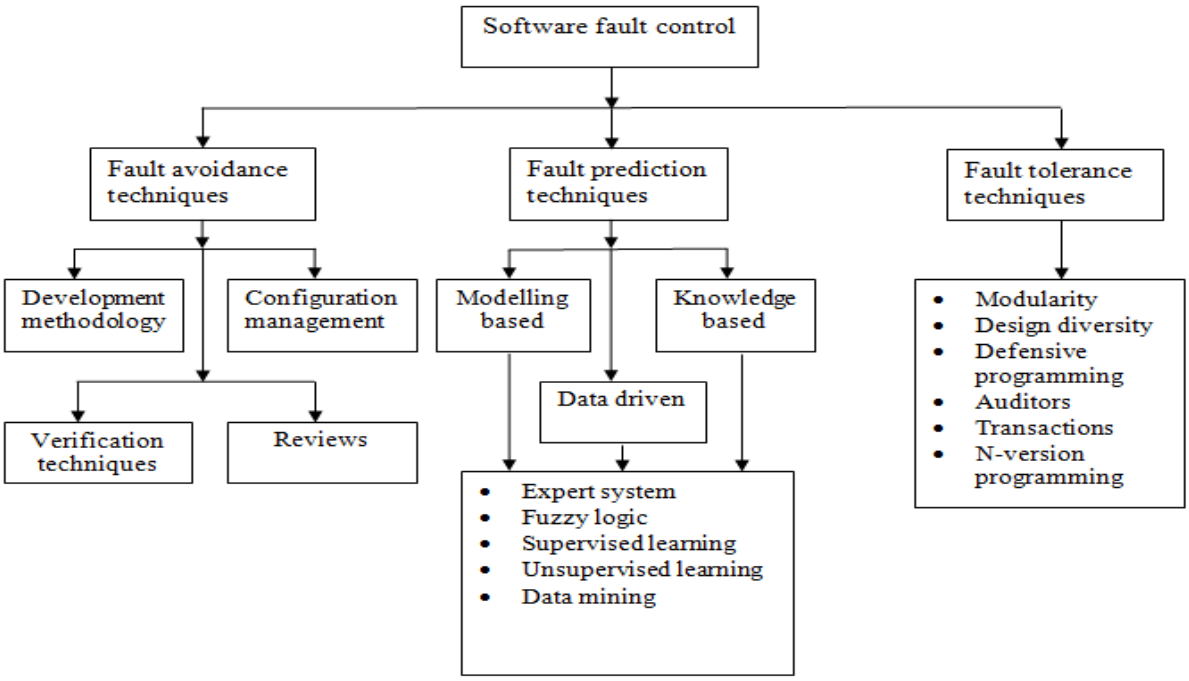

Fig. 1. Software fault control techniques

Fig. 1 shows various fault handling techniques. It includes software fault avoidance, prediction and fault tolerance techniques. It improves the software quality by identifying the fault prone modules of software [6]. Previous data regarding the faulty modules is used by fault prediction approaches. In the large sized software it is very costly and difficult to detect the software faults. Software fault prediction is very beneficial in these cases.

II. LITERATURE REVIEW

Software fault prediction is the important activity in developing fault free software. It is very beneficial in those cases where time and cost of software development is very low. i.e; it saves time and money. It minimizes the efforts of testing team. **Ezgi Erturk et al., (2016) [4]** proposed an interactional prediction model which starts with Fuzzy Inference System while the absence of data for software project and sustain data driven technique when sufficient data available. They focussed to enhance the execution of fault prediction while maintaining significantly. A prediction task was considered to associate the programmer throughout the project. The beginning predictions made by FISs while the following predictions executed by data-driven approaches. ANN and Adaptive Neuro FIS were recruited to prove the applicability of prediction technique. The outcome was evaluated as per the operating features of receiver that specifies the mutual software fault prediction is prosperous and automatically discover the fragile modules. **Shomona Jacob et al., (2017) [5]** presented a technique to solve the problem of neglecting the faults using classification methods and feature selection methods to predicts the issues of aerospace framework accurately. Machine learning and data mining approaches were used for various scientific implementations for bigger space frameworks. Latest researches discover that existing frameworks surrender to enigmatic faults results in serious losses. The core purpose was identification of prominent feature selection and prediction methods which improves the accuracy. The study confirms that hybrid approach reveals the best predictive features, despite of random techniques were used to predict faults and improved accuracy and MCC (Mathew's Correlation Coefficient). **Ezgi Erturk et al., (2015) [6]** presented the implementation of ANFIS for issue of predicting faults in software, additionally, ANN and SVM techniques were created to analyse the presentation of ANFIS. The major assumption of software is to reduce the failures occurred. Software fault prediction is an area to forecast the fault vulnerability of upcoming modules by applying necessary prediction metrics and heuristic fault data. McCabe metrics are considered due to their thorough inscription towards

 ISSN: 2454-4248
330 – 336

programming attempt. ROC-AUC was utilized as performance parameter for better results. **Ahmet Okutan et al., (2014) [7]** proposed an approach by utilising Bayesian grid to traverse the link among software metrics and default vulnerability. They used several datasets to prove that LOC, LOCQ and RFC are effective on default vulnerability and influence of DIT and NOC on imperfection is limited. They introduced a metric namely LOCQ (Lack of Coding Quality) utilized to forecast imperfection and efficiency of object oriented parameters such as WMC and CBO. Bayesian networks were utilized to regulate the probabilistic authoritative relation between defect vulnerability and software metrics. They concluded that considering effective software metric instead of bigger ones will enhance the success rate of default forecast methods. **Shuo Wang, et al., (2013) [8]** studied the challenge that class imbalance learning techniques could be beneficial for software fault prediction with a purpose to discover better solutions. They explored various methods comprising of threshold moving, ensemble algorithm and resampling approaches. Numerous amounts of machine learning techniques were examined to forecast faults in software to reduce testing costs, etc. Class imbalance learning particularize in handling classification issues with imbalanced conveyance, i.e., helpful for fault prediction. In addition they explored AdaBoost NC to show the best performance as per the parameters such as: balance, G – mean, AUC, etc. **Tracy Hall, et al., (2012) [9]** investigated the manner in which the individual variables and context of modals were used and application of modelling approaches effect the fault prediction models' execution. They analyse and arranged the results of previous studies to create a significant criterion. The models based on simple modelling methods like Logistic Regression or Naïve Bayes performed well. In table 1 described that the literature review with various methods used in previous papers and evaluated the performance parameters like precision, recall, ROC acc, MCC , accuracy, balance and G-mean and compared with existing performance parameters. Fig. 2 shows the software fault scenario. Fault detection, fault diagnosis and fault prediction activities plays very important role in the development of fault free software [10]. These practices include various techniques.

Table 1.

Author	Year	Techniques Applied	Parameters Considered						
			Precision	Recall	ROC_{AUC}	MCC	Accuracy	Balance	G-Mean
Ezgi Erturk	2016	FIS (Fuzzy Inferences System), ANN, ANFIS	✓	✓	✓	X	X	X	X
Shomona Jacob	2017	Hybrid Feature Selection	X	X	X	✓	✓	X	X
Ezgi Erturk	2015	ANN, SVM, ANFIS	X	X	✓	X	X	X	X
Ahmet Okutan	2014	Bayesian Networks	X	X	X	X	✓	X	X
Shuo Wang	2013	Naïve Bayes, Random Forest	X	X	✓	X	X	✓	✓

Literature Review with various techniques and performance parameters

 ISSN: 2454-4248
330 – 336

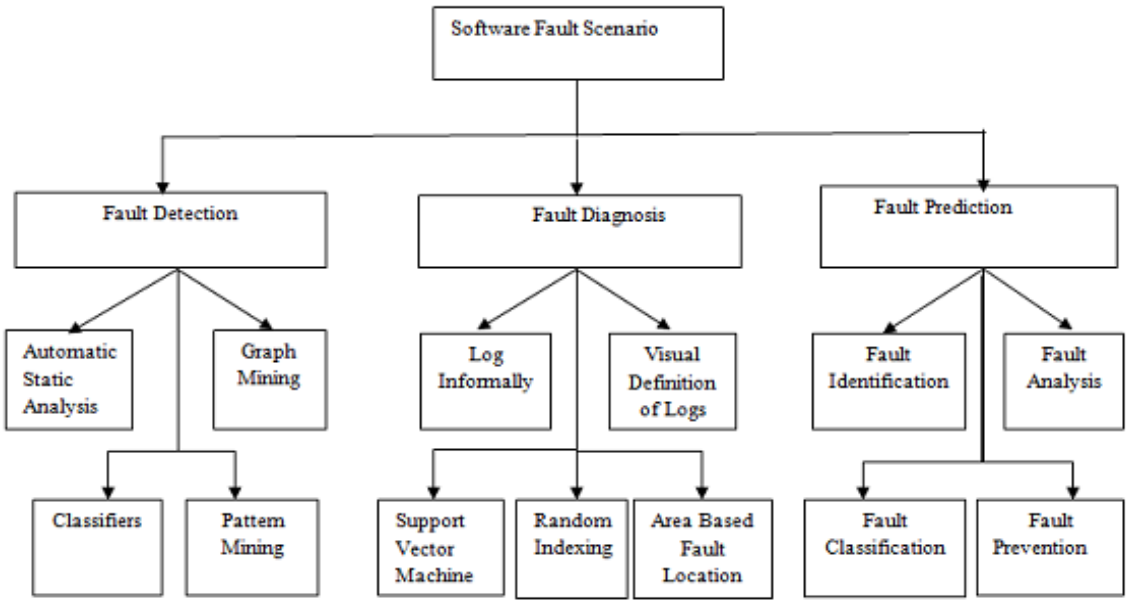

Fig. 2. Software fault scenario

III. SOFTWARE FAULT DETECTION MECHANISM

A simulation model was proposed by utilizing results of fault prediction to calculate the cost effectiveness of strategies used [10]. The proposed model estimates the qualified faults related to allocation strategy and set of segments and fault prediction results, based on which 25 percent of test effort has been reduced while examining faults in testing. Static and dynamic are 2 detection strategies applied.

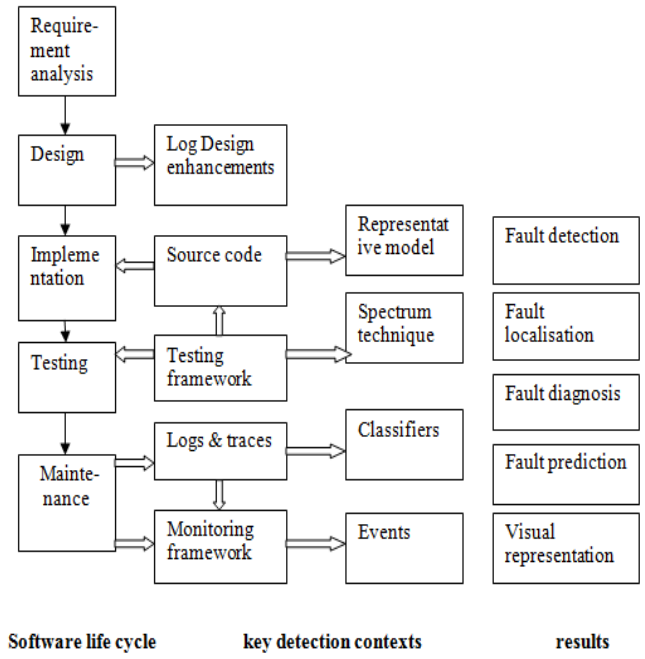

Fig 1: Project phases correlation with detection activities [17]

Static includes automated tool processing through code and data to discover defects, based on pre-set rules and specification on desired output. Dynamic approach includes pattern matching and studying log files.

The detection tasks took place at various phases of life cycle. Above figure represents the simple view of project activities. The span of solutions utilizes any one combination depending upon their requirements and criticalness. Several other faults detection approaches are:

- ***Automated Static Analysis:*** ASA is a commonly done for analysing code. It's an oldest technique still in practice but automated tools are rapidly used for standard issues like memory loss, using variables, etc. the efficiency of ASA tools for open source code discovers 3% faults.
- ***Graph Mining:*** It is an approach based on control flows help to locate non-crashing faults. It simplifies the processing calls and represents functions.
- ***Classifiers:*** A neural network, decision tree identifies unusual events. Bagging and Naïve Bayes are generally used classifiers. A latest research [11] proposes an unsupervised model represent the distribution of each code region' behaviour.
- ***Pattern Mining:*** Pattern based detection classifier that distinguishes iterative forms of classification by utilising software traverse analysis. [12] Shows

 ISSN: 2454-4248
330 – 336

the conceptual pattern to discover the relation among normal process and handing functions.

IV. SOFTWARE FAULT PREDICTION MECHANISM

SFP is the most common approach used to predict the defects or faults in the software system. It improves the cost and time required to deliver the fault free software. Several faults unfold while development sequence. Injecting faults at beginning and removing through the process of development isn't possible. The faults occurred within the process of development, therefore, the prevention became crucial part to enhance the software's quality and minimizes the overall time, costs and resources [13].

Fault prevention is crucial in software development cycle where a team focuses on fault identification and correction. Time required for fault analysis at initial stage minimize the resources and cost. Fault injecting techniques enables prevention knowledge to enhance quality and productivity.

Fault Prevention Tasks:

Several activities performed in prevention of fault:

- ***Fault Identification:*** It's a pre-set campaign focuses to highlight the particular faults. Mostly Faults are recognized in design review, code and GUI review and unit testing at several stages.
- ***Fault Classification:*** An ODC (Orthogonal Defect Classification) approach used to classify faults at the time of occurrence. Bigger projects are required to be analysed and classified in detail. Whereas small projects classified in first level to save effort and time.
- ***Fault Analysis:*** It is a simultaneous process to improve quality. RCA (Root Cause Analysis) software plays vital role in analysis. RCA focus to identify main cause of flaws to be initiated.
- ***Fault Prevention:*** This is important process in software. Initially discovering the cause and later objective is to prevent their occurrence [14].

V. FAULT DIAGNOSIS

The normal input techniques used for responsibility diagnosis are core dumps and log-files. Core a dump are created only, when there is a crash and contains the memory picture through crash and given dissimilar to study. Important work in future will do to enhance the log-files to present more applicable info and also read automatically and alert consumers for any abnormalities [15] .

- *Log Informally regarded data*

Individual point of logging is verified and the control flow and data flow is tracked till that point within the method. The attributes involved in the situation and each memory position added are the information of interest to the support engineer who is added is studying the fault. In such information is defined to be included to the similar log-point.

- *Visual definition of logs*

Log file info is normally disconnected and it takes large interval of time for developers to understand associating the related. Easy method is to relate messages based-on the message –type. There is a fixed relation between message-type and the set of variable groups referenced by message of that kind. The graph of this relation is to identifier-graph. All identifier class and a message type if the messages of the type add variables of the class.

Edges used here are un-directed. In this graph help developers spot various regarded groups of deficiencies in logs as-well-as variables that aren't present.

- *Support Vector Machine*

The machine learning algorithm is a normally used in data mining method which could be used to identify the software faults form the log-files.

- *Random Indexing*

It is used to define sequences of operations extracted from the log files, SVM is used to classify them as either success and fail. In this technique is good hold due to the sequential nature of system messages. To enhance the accuracy rate of non-failure messages SVMs are used.

- *Area based fault localization*

Area based fault localization methods could be used to locate a fault in an erroneous segment of code , when it is defined that some specific program is fading. This is automated approach which could be applied on a

 ISSN: 2454-4248
330 – 336

program with-out much knowledge about the execute program. Hence suits best for analysis the software methods , but could be used in other steps as well [16].

VI. BENEFITS OF SOFTWARE FAULT PREDICTION

SFP systems are used to predict the software faults. They use the past data to predict the faults in the software. Some benefits of software fault predict are:

- Software fault prediction improves the quality of software by predicting the faulty modules.
- It also helpful in the development of the highly reliable software systems.
- It also reduces the cost of the software by predicting the faulty modules before the testing process begins.
- It improves the time required to deliver the software. If we have idea about the faulty or fault free modules, then testing team can provide more focus on the faulty modules and testing efforts on fault free modules can be reduced.
- It can improve the testing process by reducing the test efforts.
- It can be helpful to increase the stability of the software system.

VII. CONCLUSION

Today, requirement of software quality is expanded exceptionally fault tolerant framework. In this review paper, study on software fault prevention, fault detection, fault prediction structure in relation to the recent trend of the new technologies have been explained. Some detection of flaw and software system used to identify the various approaches and methods used, but not every suitable in every system. The way to verify the tendency to high levels in crossbreed methods and models are in inclined towards more system oriented solutions for prevention and detection. Software issue handling in new day applications are in the early phases of research and the solution model try to construct the tolerance phase as much as possible. It improves the quality of software by predicting the faulty modules. The proposed work in the software fault prediction system can be implemented using FIS (Fuzzy Interference System) and Optimize the predicted faults using PSO (Particle Swarm Optimization) Approach. It will calculate the performance parameters like precision and recall.

REFERENCES

[1]. Abaei, Golnoush, and Ali Selamat. "A survey on software fault detection based on different prediction approaches." *Vietnam Journal of Computer Science* 1, no. 2 (2014): 79-95.

[2]. Catal, Cagatay. "Software fault prediction: A literature review and current trends." *Expert systems with applications* 38, no. 4 (2011): 4626-4636.

[3]. Evett, Matthew, Taghi Khoshgoftar, Pei-der Chien, and Edward Allen. "GP-based software quality prediction." In *Proceedings of the Third Annual Conference Genetic Programming, volume*, pp. 60-65. 1998.

[4]. A. Okutan and O. Yıldız, "Software defect prediction using Bayesian networks", *Empirical Software Engineering*, vol. 19, no. 1, pp. 154-181, 2012.

[5]. C. catal and B. Diri, Investigating the Effect of Dataset Size, Metrics Sets, and Feature Selection Techniques on Software Fault Prediction Problem. elsevier, 2009, pp. 1040-1058

[6]. Erturk, Ezgi, and Ebru Akcapinar Sezer. "Iterative software fault prediction with a hybrid approach." Applied Soft Computing 49 (2016): 1020-1033.

[7]. Jacob, Shomona, and Geetha Raju. "Software defect prediction in large space systems through hybrid feature selection and classification." Int. Arab J. Inf. Technol. 14, no. 2 (2017): 208-214.

[8]. Erturk, Ezgi, and Ebru Akcapinar Sezer. "A comparison of some soft computing methods for software fault prediction." Expert Systems with Applications 42, no. 4 (2015): 1872-1879.

[9]. Okutan, Ahmet, and Olcay Taner Yıldız. "Software defect prediction using Bayesian networks." Empirical Software Engineering 19, no. 1 (2014): 154-181.

[10]. Wang, Shuo, and Xin Yao. "Using class imbalance learning for software defect prediction." IEEE Transactions on Reliability 62, no. 2 (2013): 434-443.

[11]. Hall, Tracy, Sarah Beecham, David Bowes, David Gray, and Steve Counsell. "A systematic literature review on fault prediction performance in software engineering." IEEE Transactions on Software Engineering 38, no. 6 (2012): 1276-1304.

[12]. Dhanalaxmi, B., G. Apparao Naidu, and K. Anuradha. "A Review on Software Fault Detection and Prevention Mechanism in Software Development Activities."

[13]. Monden, Akito, Takuma Hayashi, Shoji Shinoda, Kumiko Shirai, Junichi Yoshida, Mike Barker, and Kenichi Matsumoto. "Assessing the cost effectiveness of fault prediction in acceptance testing." *IEEE Transactions on Software Engineering* 39, no. 10 (2013): 1345-1357.

[14]. Bronevetsky, Greg, Ignacio Laguna, Bronis R. de Supinski, and Saurabh Bagchi. "Automatic fault characterization via abnormality-enhanced classification." In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*, pp. 1-12. IEEE, 2012.

 ISSN: 2454-4248
330 – 336

[15]. Lerner, Barbara Staudt, Stefan Christov, Leon J. Osterweil, Reda Bendraou, Udo Kannengiesser, and Alexander Wise. "Exception handling patterns for process modeling." *IEEE Transactions on Software Engineering* 36, no. 2 (2010): 162-183.

[16]. Lo, David, Hong Cheng, Jiawei Han, Siau-Cheng Khoo, and Chengnian Sun. "Classification of software behaviors for failure detection: a discriminative pattern mining approach." In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 557-566. ACM, 2009.

[17]. J Zheng, L Williams, N Nagappan, W Snipes, J P. Hudepohl and M A. Vouk, "On the Value of Static Analysis for Fault Detection in Software", *IEEE Transactions on Software Engineering*, Vol. 32, No. 4, April 2006.

[18]. T. Khoshgoftaar and E. Allen, "Predicting the Order of FaultProne Modules in Legacy Software*", Proc. Int'l Symp. Software Reliability Eng*., pp. 344-353, 1998.

[19]. Scholar, P. G. "Software Fault Detection and Diagnostic Techniques: A Review and Current Trends.", *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)* ISSN: 0976-1353 Volume 13 Issue 1 –MARCH 2015.