ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 5 Issue 5 September 2017

International Manuscript ID : 23482001V5I5092017-07

(Approved and Registered with Govt. of India)

Comparative Evaluation of Approaches and Mechanisms for Software Metrics

Ashwani Sethi*[1]*, Surinder Kumar Sharma*[2]*,

Professor, Guru Kashi Unversity[1]

Research Scholar, Guru Kashi University[2]

Abstract

Software metrics refers to the measurement criteria and related dimensions to analyze the overall performance and quality factors in the software. It is directly associated with the software risk that can be measured using software metrics. If there is poor value from software metrics, its outcome and overall performance is degraded. This research manuscript focus on the integration of software metrics and related dimensions on the software quality and risk and by this way there is more need to escalate the processes and paradigms to enhance the metrics for overall quality of the software and with minimum risk.

Keywords ***-*** *Comparative Evaluation of Software Metrics, Software Quality, Software Metrics, Software Quality and Risk Management*

Introduction

Software metric [1] is a measure of software characteristics which are quantifiable or countable. Software metrics are important for many reasons, including measuring software performance, planning work items, measuring productivity, and many other uses. Within the software development process, there are many metrics that are all related to each other. Software metrics are related to the four functions of management: Planning, Organization, Control, or Improvement [2]. The goal of tracking and analyzing software metrics is to

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 5 Issue 5 September 2017

International Manuscript ID : 23482001V5I5092017-07

(Approved and Registered with Govt. of India)

determine the quality of the current product or process, improve that quality and predict the quality once the software development project is complete.

There is no standard or definition of software metrics that have value to software development teams. And software metrics have different value to different teams. It depends on what are the goals for the software development teams.

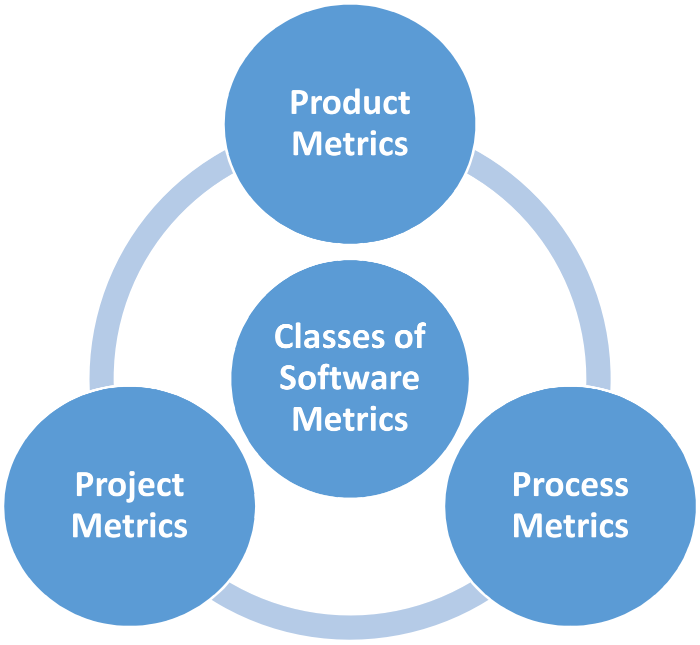

Figure 1. Classes of Software Metrics

Product metrics refers to the integration of characteristics on size, complexity, design feature, quality and performance. Process metrics is based on the risk management aspects with development and maintenance factors. Project metrics is having focus on the overall execution with the software cost, productivity and scheduling factors.

Factors for software metrics include, Balanced scorecard, Bugs per line of code, CISQ automated quality characteristics measures, Code coverage, Cohesion, Comment density, Connascent software components, Constructive Cost Model, Coupling, Cyclomatic complexity (McCabe's complexity), DSQI (design structure quality index), Function Points and Automated Function Points, an Object Management Group standard, Halstead Complexity, Instruction path length, Maintainability index, Number of classes and interfaces, Number of lines of code, Number of lines of customer requirements, Program execution time, Program load time, Program size (binary), Weighted Micro Function Points and many others.

Halstead Metrics for Evaluation of Complexity

The original performance of a software design is associated with the assessment of complexity measures and metrics. Simply development of code and testing using automation tools are not sufficient because only these perspectives can increase the overall overhead on different resource of the system. The system resources which are directly affected by the software design and code are Memory, Processor, Execution Time, Dependent Libraries and many others. In year 1977, M. H. Halstead devised the metrics for the measurement and evaluation of software complexity using different code components and categories. This metrics is more focused towards the implementation of program code based on the classical components including Operators, Operands and their relative occurrences. That was the time when the Object Oriented Programming (OOP) was not prominent.

Key elements and constituents of the Halstead metrics include the following

Table 1. Indicators and Elements of Halstead Metrics

Element or Indicator	**Description**
n1	Number of unique operators
n2	Number of unique operands
N1	Number of total occurrence of operators
N2	Number of total occurrence of operands

Relation of Software Metrics with Software Quality and Associated Risks

Software Risk Management is directly associated with the software metrics and is one of the prominent domains of research in software engineering which includes the prior identification, processing and management of risks and vulnerabilities associated in the software development. Software Defect Prediction in software engineering used to predict the deformity in the software module using software metrics. Numbers of defect are present during the development or after the delivery of software module. To obtain high quality software the prediction process is followed to predict to the defects. The need of obtaining high quality software with enriched metrics aspects is to gain customer loyalty. Few big organizations are using this prediction process as they release their software and software versions frequently and they have less time so instead of manually predicting the defects they use software deformity process.

Review of Literature

A number of researchers and practitioners have worked on the analysis of similar domain with the suggestive remarks but there is huge scope for the improvement in cases where the deep evaluation of the tools, technologies and paradigms are required to be done. Enormous multi-sources based manuscripts, research papers and articles are analyzed from the time span up to year 2017 so that the latest trends in library and information technology can be evaluated.

M.Zavvar et al. (2017) [3] focus on the classification of metrics based risks associated with the process of software development. The proposed approach is using software metrics with Support Vector Machine (SVM), a soft computing approach for the optimization of results. The parameters of Area Under Curve (AUC) and Classification Accuracy Rate (CAR) are evaluated and presented in this manuscript. The comparisons of the projected results are done with K-Means and Self Organizing Maps (SOM). Edzreena Edza Odzaly et al. (2017) [4] underlines the use of software agents for the agile based risk management in the process of software projects development. The work highlights the integration of data collection from live project environment so that the analytics on collected data can be done effectually. The use of agent programming with rule based analytics can give results with the higher degree of accuracy and performance. An Tool and Model titled ART is presented in this work with the demonstration of effectual results. Fuqiang Lu et al. (2017) [5] integrated the use of nature inspired soft computing approaches of Simulated Annealing (SA) and Genetic Algorithm (GA) for the software risk management in the outsourcing based projects. The work empirically evaluates the performance states of GA, SA and SAGA with the overall

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 5 Issue 5 September 2017

International Manuscript ID : 23482001V5I5092017-07

(Approved and Registered with Govt. of India)

conclusion that SAGA is quite effectual and performance aware as compared to the other traditional approaches. Daniel Mendez Fernandez et al. (2017) [6] worked on the perspective of evidence based software risk management in software engineering. The process of requirements engineering (RE) is enriched with the evaluation of data from 228 companies with the formation of probabilistic network. The approach of evaluation of datasets in spreadsheets is integrated for the validation and final results. Tobias Rauter et al. (2016) [7] devised Asset-Centric based assessment of risk in the software components. The manuscript evaluates the assorted dimensions of risks and its association with the metric based analysis. The component based model is adopted in this work with the identification of trust architecture for higher degree of security and overall integrity of the software development process. Abdelrafe M. S. Elzamly (2016) [8] proposes and depicts the use of fuzzy regression based modeling approaches for risk management in the software projects. The work presented 49 risk factors associated with the software project development and these can be effectually pushed back using fuzzy based approach with multiple regression. B. Chaitanya Krishna et al. (2016) [9] presented the approach towards software risk management using Principal Component Analysis (PCA) and Halstead Methodology. The work is focusing on the evaluation and identification of software risks using Halstead approach so that inner factors which are prominent for software can be processed with the higher performance and maximum throughput. The work process is divided into four phases or modules with the association of Environment, Organization, Technology and People. The further steps include the estimation of each phase, assessment of inherent risks and integration of PCA. Pradnya Purandare (2016) [10] presented the framework for software risk management using Entropy based approach. This approach is focused towards the evaluation of key risk factors in software project development and the use of Shannon Entropy for analysis of risks. The probability of having risk in the specific phase is done in this work that is associated with every software development phase. The work underlines the effectual and high performance application of Entropy in the management of risks in the software development process. Morakot Choetkiertikul et al. (2015) [11] integrates network based classification approach for the predictions of delays and risks in the software projects which further lays the foundation in the risk assessment architecture. The work evaluates the projected approach on multiple parameters including precision, accuracy, recall, F-measure, Area under Curve (AUC), ROC so that the consistency of the algorithmic approach can be evaluated. The work predicts the tasks and modules in the software projects which can increase the risks of delays and additional time which can be exploited as risk. Abdelrafe Elzamly (2015) [12] uses Linear Stepwise Discriminate Analysis (LSDA) Approach for the prediction of software risks and related threats which can directly or indirectly affect the overall performance of the software project. The key aim and goal of this work is to integrate and implement the LSDA approach for prior prediction of the software risks and further classification as Low, High and Medium classes. The approaches and techniques for the control of risks is also underlined and used so that the integrity of projected approach can be analyzed. Ramakanta Mohanty et al. (2017) [13] presented the work on machine learning based software defect prediction. The approach of classification is used for the categorization of the software defects and final predictive analytics. The approaches used in the implementation include J48, GMDH, CART, TreeNet and Genetic Programming. Riya Singh et al. (2017) [14] worked on the software defect prediction using averaging likelihood ensemble technique that is closely associated with the random forest algorithm. The work is presenting the effectual 6% efficiency as compared to the traditional approaches. The benchmark dataset of PC4 is taken for research analytics and predictive mining. Md. Mohsin Ali et al. (2017) [15] presented the work on software defect prediction and metric selection using a parallel framework in the cloud environment. Two different hybrids have been developed using Fisher and Maximum Relevance (MR) filters with a Artificial Neural Network (ANN) based wrapper in the parallel framework. The evaluations are performed with real defect-prone software datasets for all parallel versions. Experimental results show that the proposed parallel hybrid framework achieves a significant computational speedup on a computer cluster with a higher defect prediction accuracy and smaller number of software metrics compared to the independent filter or wrapper approaches. Haidar Osman et al. (2017) [16] devised the approach of automatic feature selection with the regularization for the improvement in the bug prediction and overall accuracy of the software system. The analysis of ridge, lasso and elasticnet is done with the

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 5 Issue 5 September 2017

International Manuscript ID : 23482001V5I5092017-07

(Approved and Registered with Govt. of India)

implementation of linear and poission regression as the bug prediction approaches on the Java based system. We compare the mean squared error of Poisson regression and linear regression before and after applying three regularization methods: Ridge, Lasso, and ElasticNet. The work show that regularization improves the performance of both regressions and reduces the root mean squared error by up to 50%, while also increasing the stability of the prediction. Satya Srinivas Maddipati et al. (2018) [17] integrated the adaptive neuro fuzzy inference system for software defects prediction. In this paper software defects are predicted using Adaptive Neuro Fuzzy Inference System(ANFIS). Initial Fuzzy Inference System(FIS) was derived using Subtractive Clustering method and then FIS was trained using hybrid learning rule. The performance of the classifier is measured in terms of AuC values for these imbalanced datasets. The work compared the results of ANFIS with cost sensitive neural networks. The Receiver operating characteristics (ROC) curves are generated and presented in Result section. The ROC values of ANFIS are found satisfactory compared to cost sensitive Neural networks. The authors proposed Adaptive Neuro Fuzzy Inference System(ANFIS) for identifying defective prone modules. ANFIS method generates sugeno fuzzy model. Initially FIS was generated by subtractive clustering method and it was trained by ANFIS. The work applied cost sensitive neural networks on SDP datasets with different parameters and the results are compared with ANFIS. The performance is measured in terms of AuC values and found satisfactory results with ANFIS. Jifeng Xuan et al. (2017) [18] underlines the software metrics for defects prediction and analytics using soft computing approach. To determine the order of applying instance selection and feature selection, we extract attributes from historical bug data sets and build a predictive model for a new bug data set. We empirically investigate the performance of data reduction on totally 600,000 bug reports of two large open source projects, namely Eclipse and Mozilla. The results show that our data reduction can effectively reduce the data scale and improve the accuracy of bug triage. The work provides an approach to leveraging techniques on data processing to form reduced and high-quality bug data in software development and maintenance. In this paper, the authors combine feature selection with instance selection to reduce the scale of bug data sets as well as improve the data quality. To determine the order of applying instance selection and feature selection for a new bug data set, the work extract attributes of each bug data set and train a predictive model based on historical data sets. The work empirically investigates the data reduction for bug triage in bug repositories of two large open source projects, namely Eclipse and Mozilla.

Conclusion

Software metrics evaluates the performance of any software on assorted parameters and criteria by which the cumulative effectiveness of the software is analyzed. Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. These are more closely associated with process and product metrics than with project metrics. This work focuses on the different software metrics with the relation to software quality as well as the risk management. A number of approach are devised so far for the escalation of metrics and overall quality, still the soft computing approaches are quite prominent and can be used to enhance the metrics parameters and overall effectiveness with higher degree of accuracy.

References

[1] Fenton, N. and Bieman, J., 2014. Software metrics: a rigorous and practical approach. CRC press.

[2] Basili VR. Software modeling and measurement: the Goal/Question/Metric paradigm. 1992.

[3] M. Zavvar, A. Yavari, S. M. Mirhassannia, M. R. Nehi, A. Yanpi, and M. H. Zavvar, "Classification of Risk in Software Development Projects using Support Vector Machine," J. Telecommun. Electron. Comput. Eng., Vol. 9, No. 1, pp. 1–5, 2017.

[4] E. E. Odzaly, D. Greer, and D. Stewart, "Agile risk management using software agents," J. Ambient Intell. Humaniz. Computer, 2017.

[5] F. Lu, H. Bi, M. Huang, and S. Duan, "Simulated Annealing Genetic Algorithm Based Schedule Risk Management of IT Outsourcing Project," Vol. 2017, 2017.

[6] D. M. Fernández, M. Tießler, M. Kalinowski, M. Felderer, and M. Kuhrmann, "On Evidence-based Risk Management in Requirements Engineering," pp. 1–20, 2017.

[7] T. Rauter, N. Kajtazovic, and C. Kreiner, "Asset-Centric Security Risk Assessment of Software Components," 2nd Int. Work. MILS Archit. Assur. Secur. Syst., 2016.

[8] Abdelrafe MS. Managing Software Project Risks Using Stepwise And Fuzzy Regression Analysis Modelling Techniques, 2016.

[9] B. C. Krishna and K. Subrahmanyam, "A Decision Support System for Assessing risk using Halstead approach and Principal Component Analysis," Vol. 9, No. 4, pp. 3383–3387, 2016.

[10] P. Purandare, "An entropy based approach for risk factor analysis in a software development project," Int. J. Appl. Eng. Res., Vol. 11, No. 4, pp. 2258–2262, 2016.

[11] M. Choetkiertikul, H. K. Dam, T. Tran, and A. Ghose, "Predicting delays in software projects using networked classification," Proc. - 2015 30th IEEE/ACM Int. Conf. Autom. Softw. Eng. ASE 2015, No. June, pp. 353–364, 2016.

[12] Elzamly, B. Hussin, S. S. A. Naser, and M. Doheir, "Predicting Software Analysis Process Risks Using Linear Stepwise Discriminant Analysis : Statistical Methods," Int. J. Adv. Inf. Sci. Technol., Vol. 38, No. 38, pp. 108–115, 2015.

[13] Mohanty, R., & Ravi, V.. Machine Learning Techniques to Predict Software Defect. In Artificial Intelligence: Concepts, Methodologies, Tools, and Applications (pp. 1473-1487). IGI Global. 2017

[14] Singh, R., Raja, R., & Chopra, J.. Software Defect Prediction Using Averaging Likelihood Ensemble Technique. 2017

[15] Ali, M. M., Huda, S., Abawajy, J., Alyahya, S., Al-Dossari, H., & Yearwood, J. A parallel framework for software defect detection and metric selection on cloud computing. Cluster Computing, 20(3), 2267-2281. 2017

[16] Osman, H., Ghafari, M., & Nierstrasz, O.. Automatic feature selection by regularization to improve bug prediction accuracy. In Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE), IEEE Workshop on (pp. 27-32). IEEE., 2017

[17] Maddipati, S. S., Pradeepini, G., & Yesubabu, A.. Software Defect Prediction using Adaptive Neuro Fuzzy Inference System. International Journal of Applied Engineering Research, 13(1), 394-397., 2018

[18] Xuan, J., Jiang, H., Hu, Y., Ren, Z., Zou, W., Luo, Z., & Wu, X.. Towards Effective Bug Triage with Towards Effective Bug Triage with Software Data Reduction Techniques. 2017