ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

IMPACT OF SOFTWARE PROCESS METRICS ON DEFECT DETECTION EFFICIENCY: A COMPARATIVE STUDY WITHIN ADVANCED SOFTWARE QUALITY ASSURANCE FRAMEWORKS

Rabia Kausar
Department of Software Engineering,Superior University, Lahore Punjab, Pakistan.
Corresponding Author Email: su92-mssew-f25-001@superior.edu.pk
Hamna
Department of Software Project Management,Superior University, Lahore Punjab, Pakistan.
Corresponding Author Email: hamna.noor1740@gmail.com
Saleem Zubair
Department of Computer Science and Information Technology,Superior University, Lahore Punjab, Pakistan.
saleem.zubair@superior.edu.pk
Sabah Arif
Department of Computer Science and Information Technology,UCP, Lahore Punjab, Pakistan.
sabah.arif@ucp.edu.pk
Ayesha Saddiqa
Department of Computer Science and Information Technology,Superior University, Lahore Punjab, Pakistan.
ayeshasaddiqa@superior.edu.pk

Abstract
The issue of software defects has been a major concern in the contemporary software development and has greatly impacted the reliability of the product, cost of development as well as delivery timelines. One of the main goals of Software Quality Assurance (SQA) practices is therefore effective defect detection. Software process metrics has become one of the useful tools that can be used to track the development activities and enhance quality results within recent years. The paper examines how software process metrics can influence the efficiency of defect detection by taking a combined approach to the advanced SQA and software process management practices. The research methodology includes a comparison, which is premised on a systematic study and review of empirical studies published in the period between 2015 and 2025. Various types of process metrics such as change-based metrics, effort and time metrics, defect flow metrics, and process compliance metrics are assessed in the study in various environments of the development. The findings suggest that change-based and defect flow metrics are the most influential ones in terms of the early and effective detection of defects, whereas effort and compliance metrics indirectly support the approach with the enhanced process control and resource management. The results indicate the significance of taking a more comprehensive, measurement-based, SQA that establishes process metrics against quality organizational goals. The study offers effective ideas that can be used by quality managers and practitioners to improve efficiency of defect detection in the process of improving maturity of software processes.

Keywords: Software Quality Assurance, Software Process Metrics, Defect Detection Efficiency, Software Process Management, Comparative Analysis, Software Defect Prediction

I. Introduction
In more recent software development settings, the burgeoning complexity of systems, the decreased release cycles and expectations of the users have made software quality a pivotal milestone towards success. In spite of the development methodology advances and automation tools, software defects remain a major problem in the systems and consequently, the system experiences high maintenance costs, delays in schedule implementation and substandard customer satisfaction. Consequently, the implementation of efficient defect detection and prevention mechanisms are viewed as the basic goals of the Software Quality Assurance (SQA) practices.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

Software process metrics have received a significant amount of attention in the recent years as quantitative measures used to estimate and enhance the process of software development. These metrics are based on the development activities including the code changes, defects reports, effort allocation, and process adherence that offer important information on the process performance and the quality of the product. Process metrics focus more on the way a software is developed than what is developed, unlike traditional product or code metrics that, process metrics are of great use in the initial detection of defects and continuous enhancement of a process.

Defect detection efficiency is one of the qualities that indicate the capability of an organization to detect defects at the very early stages of the Software Development Life Cycle (SDLC). The economic rework to be reduced is not only early detected, but also, the project predictability and reliability are enhanced. The successive development of SQA methods is based on using empirical information and decision-making grounded on measurements, where process measurements are crucial to detecting problematic regions of processes, tracking process variations, and assisting in proactive quality control.

Despite the fact that the problem of defect prediction and detection methods was already studied with the help of multiple software measures, the current literature is frequently mainly based on source code measures or machine learning-oriented methods of prediction. Relatively limited literature offers a combined study of the software process measures and their direct influence on the detection efficiency of defects under the more sophisticated SQA frameworks. Moreover, it is challenging to extrapolate the results of various development settings due to discrepancies in the choice of metrics, the criteria of evaluation, and context.

The gaps in this research are filled by providing a comparative analysis of software process metrics and the effect they have on the efficiency of defect detection. The paper is based on the previous empirical research and the systematic assessment of the role of various process-related metrics in augmenting the quality of defect detection in both a conventional and agile development environment.

The paper will attempt to bring a formalized insight on the effectiveness of metrics and their practical applicability by combining ideas in superior SQA practices and software process management. The key contribution that this study could make is the integrated point of view which entails the fusion of SQA concepts and process management measures to determine their value in improving the effectiveness of defect detection. The results of the study will be helpful in guiding quality managers, project heads, and investigators to choose an adequate process metric and create measurement-based quality assurance plans in accordance with current software engineering trends.

The rest of this paper will be structured in the following way: Section II will consist of the review of related work on software process measures and defect detection methods. Section III outlines the methodology and framework of comparative analysis used in the research. The results and the key findings are discussed in Section IV. In the end, Section V will provide a final conclusion to the paper and give a description of future research directions.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

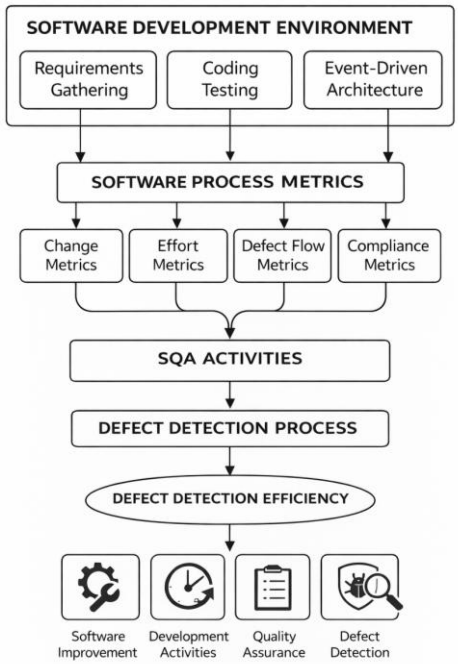

Fig. 1. Conceptual overview of software process metrics and defect detection efficiency in software quality assurance.

The overall research context and relationship between software process metrics and defect detection efficiency are illustrated in Fig. 1.

II. Literature Review

The detection and prediction of software defects has been an area of intensive research in software engineering and the main aim of the study has been to enhance the quality of the software and minimize the post release malfunctions. Early studies concentrated on product-based and source code metrics, however, current studies show the rising significance of software process metrics in the process of improving defect detection efficacy, especially in advanced Software Quality Assurance (SQA) frameworks.

A number of researchers have investigated the importance of software measures in the model of defect prediction. Ramadhina et al. provided a systematic literature review and in the process indicated how process metrics of code churn, frequency of changes and developer activity, which serve as strong predictors of defect-prone components, in comparison to the use of code metrics (only). According to their findings, process based data gives sooner and more practical data to quality assurance activities.

The mixed yet informative results of comparative studies between process metrics and traditional code metrics have been proven. Lessmann et al. have standardized several defect prediction models and made the conclusion that the success of fault detection highly depends on the relevance and quality of the metrics chosen but not on the prediction method itself [2]. This observation helps to argue the point that clear metrics of the processes may dramatically increase the results of defect detection when incorporated into SQA practices.

Researchers have paid more attention to change-based process metrics, as well as just-in-time process metrics, in the context of evolving and agile software projects. Kamei et al. demonstrated that the metrics associated with the change of code and the development process allow detecting the defects at the early stages of the rapidly developed releases [3]. Equally important, recent research has highlighted that the frequency of change, the size of commit and the amount of work that developers have are important process variables that have impact on defect introduction and defect detection effectiveness [4].

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

The use of machine learning methods in advanced SQA methods is growing in analyzing software metrics. Various researchers have applied the concept of supervised learning models in order to examine the predictive ability of process metrics. Malhotra and Sharma reviews showed that combined process measure model with historical data on defects outperforms model using source code measures only [5]. These results support the need to inject process management data into the automated quality assurance systems.

The recent literature has also been concerned with explainability and metric correlation. It is pointed out that researchers have reported that correlated or redundant measurements can adversely affect the accuracy of defect prediction and its interpretation [6]. The adoption of metric selection and validation has therefore been incorporated in SQA as a part and parcel of process measurement strategies. Research has highlighted that appropriately chosen process measures do not only bring about effective defect identification as well as decision making clarity by quality managers, but also contribute to decision making transparency [7].

Empirically, in the industrial context, empirical studies have shown that organizations that use measurement-oriented SQA models show better rates of defect detection and lower rework expenses. According to industrial case studies, compliance measures to processes, rate of arrival of defects, and effectiveness of reviews lead to effective early identification of faults in processes [8]. These results are in line with the current principles of software process management in which the continuous measurement provides quality improvement efforts.

Even though significant progress has been made, the available research shows that there are some gaps. The majority of the studies concentrate on the prediction accuracy or the individual metric effectiveness, and little is done in terms of the comparative analysis of the software process metrics in the various environment of development. Moreover, the combined use of advanced SQA practices and process management measures are not studied well especially on defect detection efficiency and not on defect prediction solely.

To conclude, the literature reviewed has proven that software process metrics are important in defect detection and quality improvement. Nevertheless, the systematic and relative assessment of these measures in the framework of the integrated SQA and process management is required. The proposed research is intended to fill this gap by the systematic analysis of the latest studies and evaluation of the efficiency of the software process metrics used to detect defects.

III. RESEARCH METHODOLOGY

The research will take the form of a comparative research to examine the effect of a software process measure on the efficiency of defect detectors in advanced Software Quality Assurance (SQA) and software process management processes. The methodology will be structured in such a way that the findings of current empirical literature are evaluated, compared and synthesized in a systematic manner, thus methodology is rigorous, and relevant to the contemporary software development setup.

A. Research Design

The study is based on a comparative analysis method of research using secondary sources as a qualitative and quantitative approach. The peer-reviewed research articles published in 20152025 were systematically reviewed to determine the most popular metrics of software processes and the impact they reported on detecting defects. The chosen studies consist of empirical experiments, industrial case studies, and systematic literature reviews, which cover all points in both academic and practical views. The comparison will be based on the studies that:

1. Use software process measures to detect/predict defects,
2. Report quantifiable results of the efficiency of detecting defects
3. Use these metrics in the SQA or process management models.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

B. Selection of Primary Studies

The main researches have been chosen through the systematic filtering process. The sources of data were digital libraries and open-access repositories like IEEE Xplore, Science Direct, Springer Open, MDPI, arXiv, and Research Gate. Such keywords as software process metrics, defect detection efficiency, software quality assurance and process management have been used to extract relevant publications. The following criteria were used as the inclusion criteria:

- Publications between 2015 and 2025, Studies that address either software process metrics or change-based metrics,
- Evaluation of performance of defect detection or prediction, which is empirical,
- Presence of English full-text articles.

The research was limited to studies that only emphasize the use of the static code measures and none that highlight process indicators to maintain the research focus.

C. Comparative Framework

A comparative framework was created in order to assess the efficiency of software process measurements on various levels. The individual studies picked were analyzed and compared according to the following criteria:

Type of Process Metrics: Change based measure, effort measure, defect based measure and process compliance measures.

Context Development: Conventional, agile or hybrid development.

- Detection Phase of a defect: Requirement, design, implementation, testing or post release.
- Meanwhile, the methods of evaluation may include statistical analysis, machine learning, or observations.
- Performance Indicators: Rate of defects detection, accuracy, recall, cost saving and early detection.

This framework allows to compare the contributions of various process metrics to detecting defects efficiently in different circumstances in a systematic way.

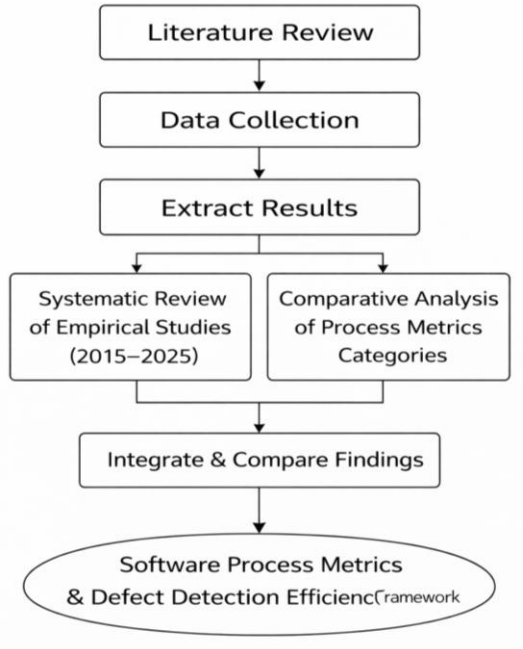

Fig. 2. Research methodology framework for comparative evaluation of software process metrics.

The research methodology adopted in this study is presented in Fig. 2.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

D. Software Process Metrics Selection

The categories of software process metrics to be compared based on frequency of use and the reported effectiveness in literature that was reviewed include the following:

1. Change- Based Metrics: Number of changed files, churn, and size of a change and frequency of commits. These metrics are sensitive to the dynamics of development activity and they were widely reported as good indicators of defect-prone changes.
2. Effort and Time Metrics: Process workload and resource allocation Process workload and resource allocation are reflected by developer effort, review time, and task duration.
3. Defect Flow Metrics: Flow of defects, rate of arrival, time-varying defect density, and defect removal throughput, which gives one a clue on the process effectiveness.
4. Process Compliance Metrics: How well the established development processes are followed, the coverage of reviews, the test execution rates which show the maturity and discipline of quality practices.

The reason why these metrics were chosen is because they are not only relevant to the advanced SQA practices and software process management but also applicable to various types of projects.

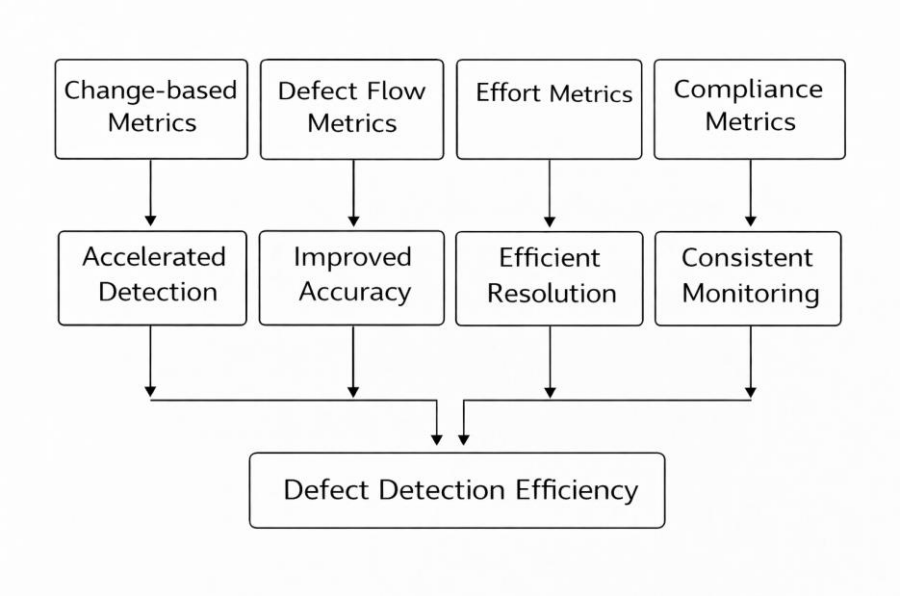

Fig. 3. Comparative framework illustrating relationships between process metric categories and quality outcomes.

Fig. 3 illustrates the comparative relationships used to analyze the impact of different metric categories on software quality outcomes.

E. Data Analysis Approach

This was done by extracting and normalizing the reported results of the selected studies, to make them able to compare their results meaningfully. In cases where quantitative data was provided, the defect detection efficiency was determined by the widely used performance metrics like precision, recall and detection rate. The qualitative data were examined to detect trends, strengths, and weaknesses of particular categories of metrics.

The results of comparison were then synthesized to measure:

- The relative effectiveness of the various categories of process metrics,
- Their input in the early fault detection,
- Their practical use in real- world SQA settings.

F. Validity Considerations

In order to minimize the possibilities of bias, several studies were examined under each metric range and results compared in various development settings. Internal and external validity of

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

the research are further enhanced by the fact that the recent and peer-reviewed literature and open-access empirical research have been used. The methodology offers a systematic and repeatable way of assessing the software process measures and defect detection efficiency. The comparative framework allows drawing evidence-based conclusions and provides a solid basis of the results and discussion provided in the further section.

IV. RESULTS AND DISCUSSION

There are the results of the comparative analysis of the chosen software process metrics and their effect on the efficiency of the process of defect detection in the framework of advanced software Quality Assurance (SQA) and software process management systems. The findings are synthesized based on empirical evidence of research found in reviewed literature and evaluated through the framework of comparison outlined in Section III.

A. Effects of Change-Based Process Metrics.

It was found that change-based process measures are always highly correlated with efficiency in detecting defects. Measures like code churn, number of modified files as well as the frequency of commits have been reported to be very effective in determining defect-prone components especially in agile and fast moving projects. Research indicates that large and frequent changes have a higher likelihood of increase in defects and therefore changes are better metrics of early defects detection in the implementation and integration stages.

Change-based metrics, in comparison to the traditional, static code metrics, allow maintaining earlier and more context-aware detection of defects, which will facilitate proactive SQA. The findings indicate that the incorporation of these metrics in ongoing integration and review processes plays an important role in improving the level of defect detection and minimize defects after the release.

B. Effort and Time Metrics Effectiveness.

Effort metrics such as developer workload, time of tasks and effort of reviews have moderate and significant effects on the defect detection efficiency. The studies reviewed show that overworking and tight schedules usually result in less efficient reviews and an increase in defect leakage. On the other hand, an equal distribution of effort and adequate time to review it is positive towards early detection of defects.

The measures of effort and time can be not the direct predictors of defects but, in combination with other process measures, enhance the SQA decision-making. The metrics are most applicable when managing the process and resource planning, which indirectly enhances the effectiveness of defect detection.

C. Role of Defect Flow Metrics

The flow measures, including the arrival rate of defects, the trend of defects density, and the effectiveness of defects removal, were discovered to be powerful measures of process efficiency. The findings indicate that projects with active tracking of defect flow measures can detect quality deterioration trends sooner and change testing and review plans.

As far as SQA is concerned, defect flow statistics offer feedback-based information, which allows improving the process. They are most effective in the testing and system validation phases in which they assist in evaluating the competence of the defect detection mechanisms.

D. Process Compliance Metrics Impact.

The metrics of process compliance such as the ability to follow the established procedures, coverage of the review, and the number of tests performed indicate that they have a serious influence on the efficient detection of defects when applied to mature organizations. The results demonstrate that the greater the process compliance, the more similar the results of detecting the defects and the less variation in quality performance.

Nevertheless, the findings also indicate that the strictness of process implementation without the contextual adjustment can decrease the flexibility in particular agile setting. As such,

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

compliance measures are effective to the extent that they are consistent to project attributes and maturity of the organization.

E. Comparative Discussion

The comparative analysis shows that none of the categories of process metrics can be adequate to ensure maximum efficiency in detecting defects. Rather, a synthesized and integrated application of more categories of process metrics has the best outcomes. Change based metrics are quite useful when dealing with detecting defects at an early stage, defect flow and compliance metrics are useful when dealing with process control and continuous improvement of the process. The
effort and time metrics are used in managerial decision-making, and indirectly lead to quality outcomes.

These results denote the necessity of having the matching metric choice with the advanced SQA goals and the software process management objectives. It also has consequences of the change in metric-centric evaluation to measurement-based quality strategies in which metrics are evaluated as decision-support tool instead of being used as independent variable.

Process Metric Category	**Representative Metrics**	**Defect Detection Phase**	**Impact on Defect Detection Efficiency**	**SQA & Process Management Relevance**
Change-Based Metrics	Code churn, commit frequency, modified files	Implementation, Integration	High – enables early identification of defect-prone changes	Strong support for proactive SQA and CI/CD quality checks
Effort & Time Metrics	Developer workload, review time, task duration	Design, Testing	Medium – indirect impact through workload balance	Supports process planning and resource management
Defect Flow Metrics	Defect arrival rate, defect density, removal efficiency	Testing, Validation	High – reflects effectiveness of detection mechanisms	Enables feedback-driven process improvement
Process Compliance Metrics	Review coverage, test execution rate, process adherence	All SDLC phases	Medium to High – ensures consistency and discipline	Indicates process maturity and SQA effectiveness

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

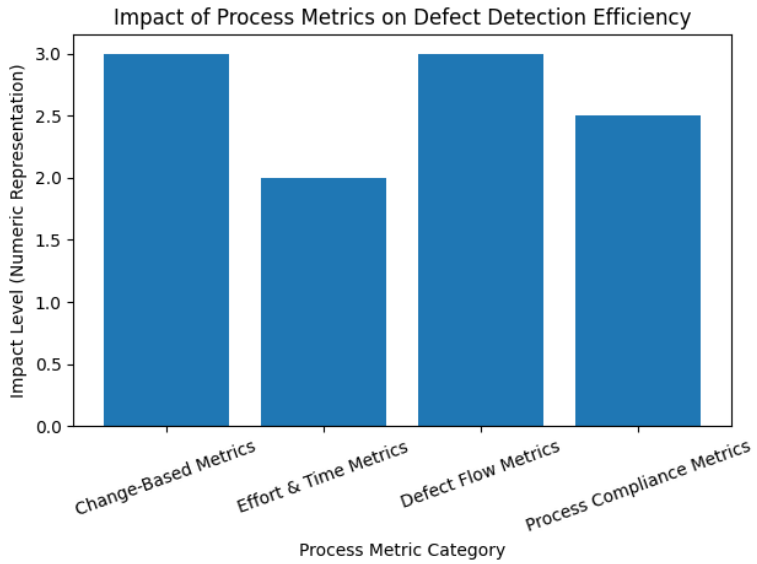

Fig. 4. Comparative impact of software process metric categories on defect detection efficiency.

Key Insights from Results

Change-based measures give the best contribution towards detecting defects at an early stage.

- Defect flow measures are important in tracking and enhancing detectiveness.
- Compliance and effort measures reinforce management decisions and control of processes.
- The combination of the usage of metrics is much better than the utilization of isolated metrics.

V. CONCLUSION AND FUTURE WORK

This paper has given a comparative approach to software process measurement and measurement effects on detection of defects in a built in model on further Software Quality Assurance (SQA) and software process management. Through the systematic review and synthesis of the published empirical studies on the topic, between 2015 and 2025, the study assessed the role of various types of process metrics in detecting defects early and effectively in various software development settings.

The results prove that software process measures are very vital in the improvement of the efficiency of detecting defects. Specifically, metrics based on change showed the most significant impact on the early detection of defects, as they represented the dynamics of development activity and pointed out defect-prone changes. The flow metrics of defects proved to be very efficient to evaluate the overall efficiency of detection processes and to provide the quality improvement based on the feedback. The measures of effort and time, and the process compliance measures played a contributing part in enhancing the allocation of resources, discipline in processes and managerial decisions. These findings affirm the fact that a combined and balanced application of different process metric categories is more successful in the detection of defects than the application of the metrics separately.

In terms of SQA, the paper highlights the move to measurement based quality assurance related practices where process metrics are utilized as strategic instruments of proactive defect control instead of being used as post-hoc indicators. The comparative framework presented in the study provides viable guidelines or guidelines in the selection and utilisation of process metrics that would be in line with organization objectives, development approaches and the complexity of projects. The insights can be useful to quality managers, project leaders and practitioners aiming at enhancing the reliability of software and process maturity.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

Irrespective of the contribution made, this study has its limitations. The analysis is based on the secondary data of the existing literature, and the different sets of data, project settings, and methods of evaluation can influence the generalization. Also, in the study, a major emphasis is put on reported metrics of detecting defects, which are not always descriptive of the context and human factors of the software quality.

This study can be further developed in the future through the use of empirical validation to determine the effectiveness of chosen process metrics using real-life industrial data. The work on hybrid methods involving the integration of process metrics with modern machine learning and explainable AI tools may be expanded to increase the interpretability of the defect detection system and its trust. Also, the investigation of selective adaptive metrics strategies applicable to agile, DevOps and large-scale distributed development settings is an encouraging future in the development of SQA and software process management studies.

References

[1] S. Ramadhina, R. F. Sari and H. T. Sukmana, Software defect prediction using process metrics: A systematic literature review, Information and Software Technology, vol. 137, pp. 1-15, 2021.

[2] S. Lessmann, B. Baesens, C. Mues and S. Pietsch, Benchmarking classification models to software defect prediction: A proposed framework and new findings, IEEE Transactions on Software Engineering, vol. 34, no. 4, pp. 485-496, July-August 2008.

[3] Y. Kamei, S. Matsumoto, A. Monden, K. Matsumoto, B. Adams, and A. E. Hassan, "The revisiting of common bug prediction findings with effort-aware model, in Proc. IEEE Int. Conf. Software Maintenance, 2010, pp. 1-10.

[4] T. Zimmermann, R. Premraj, and A. Zeller, Predicting defects in Eclipse, in Proc. 3 rd Int. Workshop Predictor Models in Software Engineering, 2007, pp. 9-15.

[5] R. Malhotra and A. Sharma, A systematic review of machine learning methods in software fault prediction Applied Soft Computing, vol. 27, pp. 504-518, 2015.

[6] S. Herbold, A. Trausch, and J. Grabowski, The fourth article is a comparative study which aims at a benchmarking cross-project defect prediction methods, IEEE Transactions on Software Engineering, vol. 44, no. 9, pp. 811-833, Sep. 2018).

[7] The WEKA data mining software: An update by M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, ACM SIGKDD Explorations, vol. 11, no. 1, pp. 10-18, 2009.

[8] N. Nagappan, T. Ball and A. Zeller, "Mining metrics to predict component failures," in Proc. 28th Int. Conf. Software Engineering 2006, pp. 452-461.

[9] A. E. Hassan, Predicting faults based on the complexity of code changes, in Proc. 31 st Int. Conf. Software Engineering, 2009, p. 78-88.

[10] F. Peters, T. Menzies and A. Marcus, Better cross-company defect prediction, in Proc 10th IEEE Working Conf. Mining Software Repositories, 2013, pp. 409-418.

[11] J. Nam and S. Kim, Heterogeneous defect prediction, 10th Joint Meeting Foundations Software Engineering, 2015, pp. 508-519.

[12] S. Kim, E. J. Whitehead and Y. Zhang, "Software changes - Clean or buggy? A classification of changes. IEEE Transactions on Software engineering, vol.34, no.2, pp.181-196, Mar.-Apr. 2008.

[13] M. D'Ambros, M. Lanza and R. Robbes, An extensive comparison of bug prediction approaches, in Proc. 7th IEEE Working Conf. Mining Software Repositories, 2010, pp. 31-41.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

[14] P. B. Li, J. Zhu, He, and M. R. Lyu, Experience report: System log analysis to detect anomaly, in Proc. IEEE Int. Symp. Software Reliability Engineering, 2016, pp. 207-218.
[15] S. H. Kan, Metrics and Models in Software Quality Engineering, 2 nd ed. Boston, MA, USA: Addison-Wesley, 2002.
[16] B. Kitchenham, S. Charters, "Guidelines to carrying out systematic literature review in software engineering," EBSE Technical Report EBSE-2007-01, 2007.
[17] A. Mockus and D. M. Weiss, "Risk of software changes prediction," Bell Labs Technical Journal, vol. 5, no. 2, pp. 169-180, 2000.
[18] Y. Zhou, H. Leung, and B. Xu, the relationships between the object-oriented metrics and change-proneness, whether there is a possible confounding effect of class size. IEEE Transactions on Software Engineering, vol. 35, no. 5, pp. 607-623, Sep.-Oct. 2009.
[19] M. Cataldo, P. A. Wagstrom, J. D. Herbsleb, and K. M. Carley, "Identification of coordination requirements: Implications of collaboration and awareness tools design, a topic at the ACM Conf. Computer Supported Cooperative Work, 2006, pp. 353-362.
[20] H. Kagdi, M. L. Collard and J. I. Maletic, A survey and taxonomy of mining software repository approaches, Journal of Software Maintenance and Evolution, vol. 19, no. 2, pp. 77-131, 2007.
[21] S. Herbold, Training data selection, cross-project defect prediction Empirical Software Engineering, vol. 23, no. 5, pp. 2664-2692, 2018.
[22] Z. Li, X. Jing, and X. Zhu, Heterogeneous defect prediction with two-stage ensemble learning Automated Software Engineering, vol. 25, no. 4, pp. 801-836, 2018.