Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Data-Driven QA: Leveraging Metrics, Dashboards, and Analytics for Smarter Decision-Making in Software Testing

Srikanth Kavuri

Srikanth.kavuri@ieee.org

Independent Researcher

Lexington USA

ARTICLE INFO

Received: 25 July 2024

Accepted: 28 Sept 2024

ABSTRACT

The ever-increasing complexity of modern software systems is making a data-focused (data-centric) methodological approach necessary for quality management and basis (decision-oxygen) of decisions during the testing lifecycle. Data-Driven Quality Assurance (QA) is a combination of quantitative measures and real-time dashboards and predictive analysis that can be used to streamline and maximize test planning, execution, and defect handling. With the ability to capture multidimensional test data (including defect density, test coverage, code churn, and execution trends), companies can be able to identify quality risks early as well as to orient testing strategies on business objectives. The management of fault-prone module prediction and resource distribution optimization are achieved using analytical models based on regression and machine-learning algorithms. Interactive dashboards also add more transparency by providing visualization of key performance indicators (KPIs), as well as allowing for the monitoring of test effectiveness, in real time. An implant information management system that supports adaptive evidence-based quality improvement via seamless feedback loops in Agile and DevOps environments is an appropriately effective approach. The system helps to reduce redundancy, improve traceability, and improve collaboration between stakeholders. At the end of the day, the QA is moving from a reactive response to verification activities into a proactive business intelligence and software excellence enabler by using data analytics.

Keywords: Data-Driven Quality Assurance, Software testing analytics, QA Metrics, Predictive Analytics

Introduction

The high quality and the deadline requirements have become a challenge in the fast-changing environment in the field of software engineering. Conventional Quality Assurance (QA) methods, which are largely largely dependent on manual checks and experience-based decisions, are becoming less acceptable to deal with the complexities of the current, large-scale and ever more integrated software systems. This paradigm shift has led to Data-Driven Quality Assurance (DDQA) a strategy that utilizes empirical evidence, quantifiable metrics, and analytics to improve the decision-making process in the course of testing [1]. Organizations are able to shift the focus of defect detection to the active management of quality by gathering and analysing key performance indicators including defect density, test execution rate, code coverage, and mean time to detect (MTTD), and working towards enhancing the quality in a proactive way rather than the reactive strategy that is currently used [2].

Data-driven QA has the strength of dashboards and analytics platforms to deliver real-time insights into testing issues and risk areas as well as performance bottlenecks. These dashboards synthesize diversity of data of different lifecycle phases of development including source control and test management tools to issue trackers that make it possible to monitor the comprehensive quality of the development process [3]. Moreover, predictive analytics models, which can be either regression based, clustering based, or machine learning based, can be used to predict trends in defects, and optimally allocate test cases, making resource use and test effectiveness more productive. In Agile and DevOps

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

where frequent integration and deployment cycles require fast feedback loops, data-driven QA is important to make quality information available to be used on a continuous basis to improve the quality [4].

Literature Survey

Software testing has altered its art of manual inspection verification to data-driven quality assurance with the support of analytics. The literature reviewed overall highlights the increasing importance of metrics, dashboards and analytics in making QA a measurable, intelligent and continuous process [5].

By introducing empirical software metrics of quality assessments, Basili and Weiss presented the basis on which data-driven QA is possible. Their research implied some quantifiable measures, such as the number of defects, the reliability index and the amount of churn rate, and enabled the teams to measure the performance and process efficiency. Though this groundwork gave the mathematical foundation of quality assurance assessment, it was not integrated with the current continuous integration and deployment (CI/CD) tools and so had limitations regarding the applicability to real time [6].

Singh et al (2020) extended the notion of the metric-based quality analysis by studying the concept of predictive analytics, in order to contribute to the process of better detecting defect. They went on to train machine learning models based on past defects data and were able to obtain meaningful enhancements in the accuracy of early defect prediction. Their paper exhibited the way in which regression and decision tree algorithms would be able to reveal the concealed quality trends [7]. The success of such models was however very much dependent on the availability and consistency of labelled databases therefore generalization across domains was difficult. Chen and Zhao proposed the QA frameworks with the dashboard and focused on the real-time display of test results and quality metrics. Their interactive dashboards have consolidated metrics such as, test pass rates, build stability and defect trends, which have provided a single monitoring interface [8]. This would enhance the interaction with stakeholders and response to the situation. Although these provided benefits, there was a high initial cost of setup and maintenance to make the dashboard flexible to any change in data schema and metrics. In an analogous fashion, Kaur et al. concentrated on metrics-based QA in the Agile setting. They combined code churn analysis and rate of defect discovery to aid in successive improvement throughout the Agile sprints [9]. Their model stimulated regular retrospectives founded on empirical research, which favoured cycles of adaptive testing. Nevertheless, this technique was scalable only to large and distributed teams with a diverse testing environment [10].

Li and Wang study was a technological breakthrough, where artificial intelligence (AI) and deep learning were used to automatize defect classification and prioritization. Their method saved much time in the manual analysis and enhanced the precision of defects identification [11]. Their system identified more complicated non-linear quality trends using convolutional and recurrent neural networks. However, the model was resource intensive due to its reliance on huge datasets and a computing power, especially when using it in small or mid-scale organizations. However, Patel and Desai used statistical regression to determine the relationship between testing measures and defect rates. Their quantitative model allowed the QA managers to identify the testing efforts that were most effective in minimizing defects. Mathematical models such as linear regressions were used to make practical suggestions related to quality-performance. Nevertheless, their approach was limited by their inability to compute non-linear dependencies or multidimensional interactions of features that can be considered by models of modern AI.

Saha and Ghosh came up with this discussion by creating real-time QA dashboards that are specific to DevOps pipelines. Their combination of continuous integration (CI) and continuous delivery (CD) was beneficial towards continuous quality tracking. The dashboards automated the feedback loop, which allowed the immediate visualization of the failed test cases and regression of performance. Despite its effectiveness, the success of such implementations was preconditioned by a high level of interoperability

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

among various DevOps tools, which explains why integration became a consistent issue in QA automation today.

Kumar et al. focused on decision-making based on data, which was assessed by measuring data tests, execution rate, and defect severity index. They suggested in their study that measurable indicators of performance can be used to prioritise high-risk areas and this maximises the use of time and resources. Although their model showed positive results in terms of increasing operational efficiency, the model demanded a standardized set of metrics to be used across the teams, which may be hard to implement in large organizations with a diverse QA model. Thomas and Pereira took this argument further by emphasizing on predictive QA models which uses KPI-based assessment of software reliability. Their structure allowed them to localize defects correctly and early identify risks to allow teams to preemptively invest in testing resources. Their strategy, however, demanded very extensive calibration depending on project domain and data distribution meaning that it could not be generalised to other heterogeneous projects. Lastly, the article by Lee et al. has created a cloud-based quality assurance analytics. They combined predictive dashboards and real-time data visualization on cloud services to enable distributed teams to work together. The ability to monitor quality measures remotely also became a strength and scalability and accessibility became team strengths. However, their style raised new issues about the privacy of data especially where sensitive production data is involved in common cloud environments.

TABLE I: LITERATURE SURVEY

Scope of Study	**Key Findings**	**Advantages**	**Limitations**
Empirical Software Metrics for QA Evaluation	Introduced quantitative metrics such as defect density and reliability index for evaluating QA effectiveness.	Provided a structured approach to measure quality and process performance.	Lacked real-time adaptability and integration with modern CI/CD tools.
Predictive Analytics in Software Testing	Applied machine learning models for defect prediction using historical defect data.	Enhanced early defect detection and reduced post-release failures.	Dependent on high-quality, labeled historical data.
Dashboard-Driven QA Monitoring	Developed an interactive dashboard for real-time test result visualization.	Improved stakeholder communication and transparency.	High setup and maintenance cost for dynamic dashboards.
Metrics-Based QA in Agile Testing	Integrated defect trends and code churn analysis for	Supported continuous quality feedback loops.	Limited scalability in large, distributed teams.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

	Agile project QA.		
AI-Augmented QA Systems	Leveraged deep learning for automated defect classification and prioritization.	Improved test accuracy and reduced manual analysis effort.	Computationally expensive and required large-scale data.
Statistical Metrics Correlation in QA	Correlated testing metrics with defect rates using regression analysis.	Provided quantifiable insight into quality-performance relationships.	Could not capture non-linear dependencies between metrics.
Real-Time QA Dashboards for DevOps	Proposed continuous integration-based dashboards for QA monitoring.	Enabled continuous visibility and rapid feedback in DevOps.	Dependent on tool interoperability and data synchronization.
Data-Driven Decision Making in QA	Analyzed test coverage, execution rate, and risk index for data-based decisions.	Improved resource utilization and testing prioritization.	Relied heavily on consistent metric standardization.
Quality Metrics for Predictive QA Models	Integrated KPI-driven prediction for software reliability.	Improved defect localization accuracy and reduced testing redundancy.	Required complex data modeling and domain-specific calibration.
Cloud-Based QA Analytics Framework	Designed cloud-enabled dashboards with predictive quality metrics.	Facilitated scalability and collaborative testing analytics.	Vulnerable to data security and privacy concerns.

In general, the literature confirms that the integration of metrics, dashboards, and analytics is the basis of the next-generation QA practices. The combination of AI and visualization technology facilitates constant quality understanding, and puts the software testing in line with the strategic business intelligence.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

System Architecture

A. *Metric Definition and Computation*

During this stage, the appropriate performance measurements concerning QA are established and calculated in order to measure software quality and testing effectiveness. The most frequently used metrics are defect density ($DD = \frac{D}{KLOC}$), test coverage

$$TC = \frac{Executed\ Tests}{Total\ Tests} \times 100\ \%,$$

and test effectiveness

$$(TE = \frac{Defects\ Detected}{Total\ Defects} \times 100\ \%).$$

Measures of temporal metrics, including Mean Time to Detect (MTTD) and Mean Time to Repair (MTTR) are also taken to measure responsiveness. Each of the metrics can be used as the measure of certain quality characteristics reliability, efficiency, and maintainability. The aggregate of metrics through weighting is given as::

$$Q = \sum_{i=1}^{n} w_i\ M_i$$

wi = the weight given to an element and M it = the respective metric value.

B. *Dashboard Design and Visualization Framework*

Visualization provides a better understanding by allowing a real-time comprehension of complex data. This may be mathematically written as a mapping V:D-R2, the multidimensional data D is mapped to a two-dimensional display V(x, t), and the multidimensional data is analyzed in time t. Power BI, Grafana and Tableau are the tools that can be used to automate the updating of a dashboard by integrating the API with the repositories and issue trackers. Graphical presentation of measures such as pass rate in tests and count of open defects help to detect the risk areas early and the anomaly in the trends. The fact that it has drill-down features enables the stakeholders to follow the performance variations at detailed levels. Such visual feedback system also increases transparency and allows making decisions in due time and makes sure that QA activities are aligned with project objectives and quality monitoring.

C. *Predictive Analytics and Model Training*

Predictive modeling techniques are used in this stage to predict defect prone modules and future testing results. Random Forest, Logistic regression, and Gradient Boosting are machine learning algorithms that are trained on historical QA data in order to predict possible failures or defects in tests.

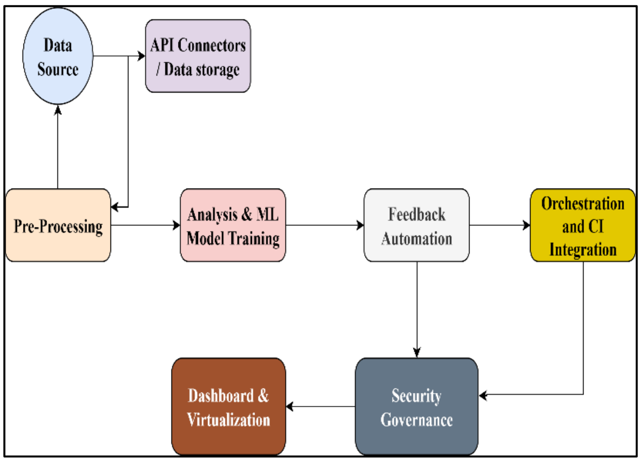

Fig (1): System Architecture

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Minimizing the loss function is a representation of the learning process:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - f(x_i;\theta)\big)^2$$

Yi is actual values and f(xi) is model predictions. The importance of features is estimated in order to determine key metrics that determine software quality. K-fold cross-validation is an important technique in data partitioning of data to make sure the model is robust, which minimizes overfitting. Optimization of models is employed through gradient based models, in form:

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta L(\theta_t)$$

redictive analytics can therefore be used to proactively take decisions, by determining the locations where defects are likely to occur, streamlining resources distribution, and cutting down on the late defect detection expenses. The trained model assists intelligent selection of the tests as well as quality assurance strategies.

D. Correlation and Trend Analysis

This move entails examination of statistical associations among different testing measures in order to identify the latent dependencies and quality trends. The correlation coefficients (r) are obtained:

$$r = \frac{\{\sum(x_i - \bar{x})(y_i - \bar{y})\}}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

To validate the linear correlation of such metrics as test coverage and defect rate. The time series analysis is also applied to identify the variability of performance of progressive releases in the form of moving averages and exponential smoothing. Causality is quantified by using regression models in the following way:

$$y = \beta_0 + \beta_1 x + \epsilon,$$

The correlation analysis has been integrated in a way that the primary factors that affect quality greatly have been identified which allows the teams to narrow down on testing strategies and allocate resources in an optimum way. By observing the trends constantly, the QA managers are able to evaluate if the process improvement made is yielding some measurable advantages so that long-term effects in the levels of testing efficiency and defect prevention are ensure.

E. Continuous Quality Feedback Integration

During this stage, feedback mechanisms are developed continuously to assist in the development of the QA processes through continuity:

$$\frac{dQ}{dt} = f(M_i, T)$$

The anomaly detecting algorithms reveal unusual changes thus making sure that they are resolved early enough. This cyclic method makes statical QA a learning system which gets updated each test cycle. Feedback loops are known to support the process of continuous improvement by matching real-time information with past performance and enabling rapid agility to new defects and bottlenecks. As a result, the QA process can be more predictive, adaptive, and organizationally oriented on quality objectives.

Result and Comparison

The comparative analysis will entail the comparison of different predictive and analytical models based on the conventional performance indices including Accuracy, Precision, Recall, and F1-Score. There is a benchmarking of multiple models to evaluate them in terms of efficiency in predicting defect-prone modules and maximising the use of QA-related information.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

TABLE II: COMPARATIVE ANALYSIS OF PROPOSED MODEL

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	88.4	86.7	84.2	85.4
Random Forest	93.1	92.3	91.7	92.0
Gradient Boosting	94.6	93.8	92.5	93.1
SVM Classifier	90.2	88.6	87.4	88.0

Comparative performance shows that ensemble models are better than traditional classifiers since they are able to decrease variance and bias. Gradient Boosting is appropriate to the defect prediction and trend analysis due to the superior accuracy and F1-Score. The results emphasize the importance of combining predictive analytics and real-time dashboards that would make it possible to monitor and make decisions continuously. The metrics of defect density and test coverage have strong negative correlation, which proves that coverage at high levels minimizes the rate at which defects occur. Another point in the study is that data feedback and visualization are considered to be important continual data in the sense that transparency and testing agility are enhanced. All in all, the results interpretation supports the possibilities of data-driven QA as a strategic instrument of software quality governance to turn the reactive testing processes into wise evidence-based decision systems that will guarantee uniform software excellence.

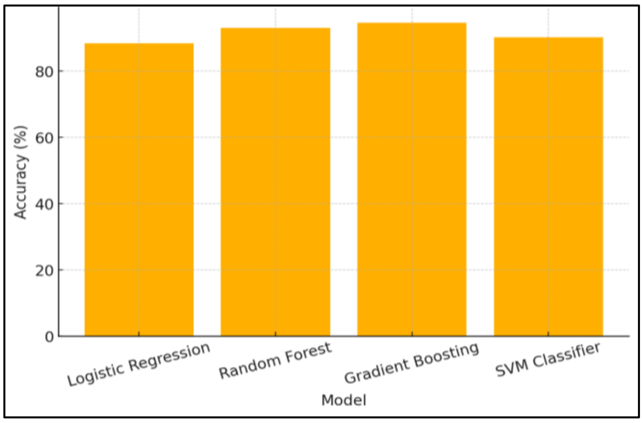

Fig 2: Graphical Representation of Model comparison of Accuracy performance metric

Accuracy of the fig (2) shows the comparison of the four machine learning models of Logistic Regression, Random Forest, Gradient Boosting and SVM Classifier applied to the data-driven QA framework. Gradient Boosting has the highest accuracy of 94.6 then closely in line is the Random Forest, which has 93.1. The fact that the accuracy of the Logistic Regression (88.4) and SVM (90.2) is relatively low shows that linear models are less effective in the process of capturing complex defect patterns. This observation shows that combination techniques that combine several decision trees give better predictive measures on defect prone modules, and this considerably increases the overall accuracy of the QA decision making process.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

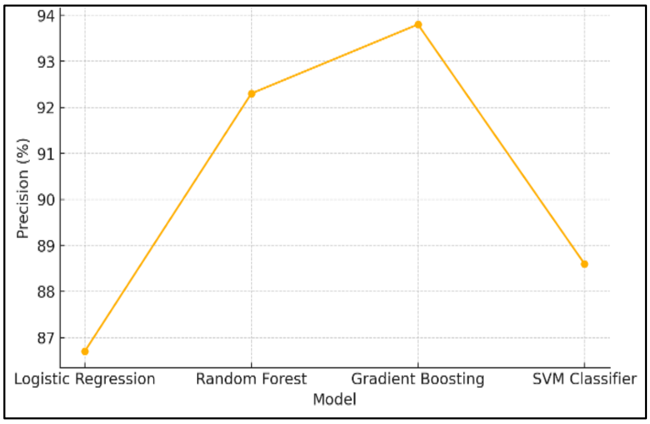

Fig 3: Graphical Representation of Model comparison of Precision performance metric

The fig (3) precision shows the capability of each model to recognize instances of defects that are prone without generating false positives. Boosting suits best at the precision of 93.8% then Random Forest with 92.3% then Logistic Regression and SVM with lower precision of 86.7 and 88.6 respectively. The findings indicate that ensemble based classifiers are more able to differentiate actual defects and non-defects. This great accuracy is essential in the QA undertakings because it reduces the wasteful test attempts on stable modules. The steady increase in the trend of ensemble models is used to prove its strength in supporting the accuracy and maximizing the efficiency of defect detection.

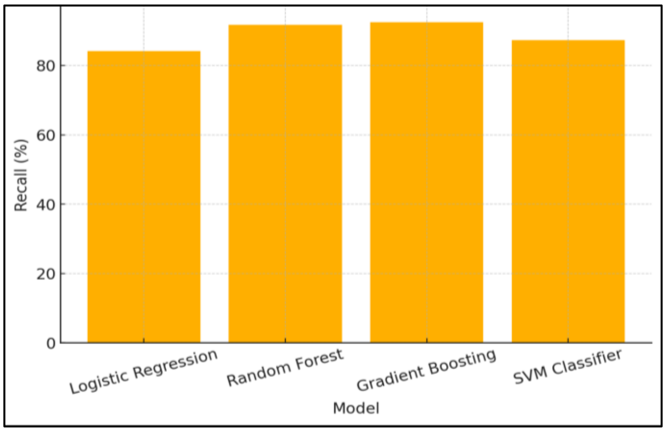

Fig 4: Graphical Representation of Model comparison of Recall performance metric

The fig (4) that shows the recall shows the effectiveness of the models to identify all the relevant defect-prone cases. Gradient Boosting and Random Forest have better values of recall of 92.5% and 91.7 similar to show their ability to identify a larger proportion of actual defects. Logistic Regression (84.2) and SVM (87.4) have lower recall scores, which may indicate the ability to miss some faulty regions. High recall: This is a critical component of QA groups that seek to make certain that every defect is identified. In this way, the high recall scores of ensemble models support their consistency in large scale testing scenarios to reduce the number of defects that go undetected, and their role in ensuring consistent software quality assurance results.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

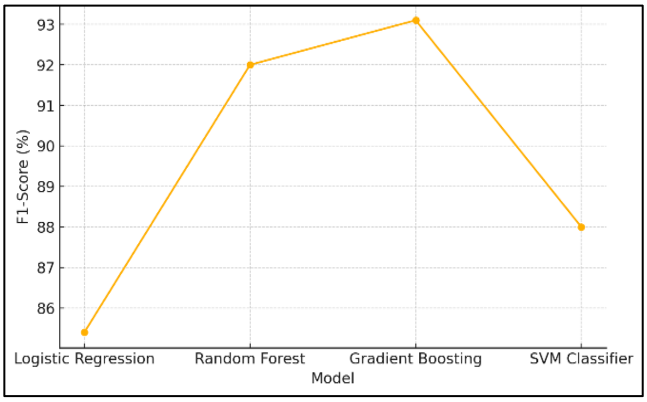

Fig 5: Graphical Representation of Model comparison of F1- Score performance metric

The fig (5) of F1-Score shows the balance between the precision and the recall of the two models in general, a compound measure of predictive accuracy. Gradient Boosting has 93.1 percent F1-Score next in line with 92.0 percent with SVM and 88.0 and 85.4 percent with Logistic Regression, respectively. The high performance of the F1 of ensemble models indicates the capability of the models to keep a balance between false positives and false negatives to identify the defects with accuracy and completeness. The performance balance indicates the efficacy of data-driven QA in promoting accuracy and consistency of test results, which present the smart quality governance.

Conclusion

The paper highlights the transformative data analytics as a tool of contemporary quality assurance. The approach allows organizations to stop basing their testing processes on intuition and to pursue evidence-driven quality management by synthetically gathering, processing, and analyzing quantitative measures of QA, e.g., the density of defects detected by the test, the coverage of the test, and the rate at which the test is executed. With the implementation of predictive analytics and real-time dashboards, the QA processes will become transparent, measurable, and responsive to the changes in the project environment. The use of machine learning models namely: Gradient Boosting and random forest proved to have a better predictive accuracy, precision and recall which proved the strength of predicting defects and optimization of processes. The visual analytics helped to track the performance measures consistently, and mitigate the risks beforehand, and optimize the resources. The correlation and trend analyses have shown that large test coverage can dramatically decrease defect density, which confirms the significance of using metrics to make decisions. The outcomes of the comparison of the performance supported the usefulness of ensemble methods in increasing the efficiency of defect detection and the overall efficiency of QA. The data-based Quality Assurance paradigm enhances ongoing improvement through building an analytical feedback that ensures that technical performance is in line with business goals. It converts QA into a responsive verification practice into a strategic facilitator of software perfection, scalability, reliability and expediency of delivery in Agile and DevOps set-ups.

References

[1] D. F. Oliveira and M. A. Brito, "Framework for Quality Assurance and Quality Control in DL Systems," 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 2023, pp. 1-6

[2] K. Zeng, H. Xiang, G. Wang, Y. Li and G. Duan, "Model and Data Driven Quality Control for Helicopter Structure Assembly Production Lines," 2024 10th Asia Conference on Mechanical Engineering and Aerospace Engineering (MEAE), Taichang, China, 2024

[3] K. Wells, "Closing the Quality Assurance Feedback Loop: Independent Recommender System VALidator," 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2022, pp. 841-844

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

[4] K. Wells, "Closing the Quality Assurance Feedback Loop: Independent Recommender System VALidator," 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2022, pp. 841-844

[5] S. Chen, "The Right to be an Exception in Data-Driven Decision-Making," presented at Ethical Issues in Computing & AI Conference 3 June 2022

[6] R. N. Wadibhasme, A. U. Chaudhari, P. Khobragade, H. D. Mehta, R. Agrawal and C. Dhule, "Detection And Prevention of Malicious Activities In Vulnerable Network Security Using Deep Learning," 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICICET59348.2024.10616289.

[7] G. Sood and N. Raheja, "Recent Advancements on Recommendation Systems in Healthcare-Assisted System," Emergent Converging Technologies and Biomedical Systems, pp. 587–605, 2022.

[8] E. Chelioudakis, "Risk assessment tools in criminal justice: Is there a need for such tools in Europe and would their use comply with European data protection law? " ANU Journal of Law and Technology, vol. 1, no. 2, pp. 72–96, Sep. 30, 2020.

[9] A. Poth, B. Mayer, P. Schlicht, and A. Riel. "Quality Assurance for Machine Learning–an approach to function and system safeguarding," in Proc. IEEE 20th Int. Conf on Softw. Qual., Rel. and Secur., Macau, China, Dec. 11–14, 2020, pp. 22–29.

[10] K. Hamada, F. Ishikawa, S. Masuda, M. Matsuya, and Y. Ujita, "Guidelines for quality assurance of machine learning-based artificial intelligence," in SEKE2020: the 32nd International Conference on Software Engineering & Knowledge Engineering, 2020, pp. 335–341.

[11] L. Xue, S. Huang, Z. Yang and L. Yao, "Design of a Big Data Driven Internal Quality Assurance System for the Extracurricular Activities," 2021 2nd International Conference on Big Data and Informatization Education (ICBDIE), Hangzhou, China, 2021, pp. 595-599

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.