Deep Science Publishing
https://doi.org/10.70593/978-93-49910-35-5_3

Chapter 3: Smart quality assurance: integrating real-time monitoring with predictive error detection

3.1. Introduction

Software systems have reached an astounding level of sophistication, which provides complex services to users. Sometimes unconsidered or inadvertently, these systems become central to the operations of tens, hundreds of thousands or even millions of end users, contributing to delivering vital services. Any problem, however small, can mess up the operations of these services, generating huge costs to their own providers and, many times, also to the society. As a consequence, software quality assurance (QA) has become one of the hot topics in software development. QA strategies aim at reducing the number of errors in deployed systems, helping guarantee minimum enforced quality levels by discovering and eliminating defects in software systems (Martínez-Fernández et al., 2022; Gadhiya et al., 2024; Carnerud et al., 2025).

Increasingly, these strategies diversify, during all the life cycle of the software, from their inception, coding and testing to their deployment and maintenance. Every phase relies on somewhat complementary error detection and prevention techniques that need to be integrated together so that their coverage becomes systematized and the tools less intrusive. Although techniques like testing and peer review are based on an inspection of the system and control and, run on a continuous support, project and system data mining, testing and execution monitoring of the system execute indirect error evaluation generated as a byproduct of these and other processes. This process is not continuous, and they cannot ensure that every error contemplates that evaluation because, to be effective, they need to be fed and integrated with data from all the other methods. But, even so, as stated before, every technique relies on different procedures, accomplished on a data set of different features. This integration, if made properly and monitored for a long time, can improve the QA techniques´ effectiveness. This is our premise for developing smart techniques. These techniques, enhanced with data from other sources, constitute data mega repos, which provide a comprehensive software behavior view.

They allow us to detect inability patterns, which notify when the system is more prone to being error prone (Najihi et al., 2022; Ozkaya, 2023).

3.2. Overview of Quality Assurance

Quality assurance encompasses a range of planned activities aimed at ensuring quality-related goals in a specific project, process, or product. This control is executed through quality management processes, which consist of the main quality management and assurance elements and processes for QA implementation, namely project management processes, quality planning, quality management activities, and process and product measurements.

Fig 1: Overview of Quality Assurance

QA implements planned activities systematically, which create evidence that related quality requirements are being fulfilled. The evidence is generated and confirmed at some points in time, also known as quality assurance checkpoints, which usually consist of the main project phases or some specific QA-related activities. If these checkpoints confirm that fulfillment of the quality requirements is not in accordance with plans, some corrective measures are taken. The corrections mostly depend on the checkpoint status and chosen strategy. In the case of quality assurance system corrections, more severe changes are applied, with an extended period for quality recovery. Finally, the product is transferred into the operations phase and reapplied using maintenance QA activities.

An evolving area of QA research makes a considerable contribution to quality improvement. More and more companies specialize in new QA method and system development in software and systems engineering, which have also taken on significance in other areas where quality is critical. Among these areas, novel and complex areas of artificial intelligence, big data, system modeling, and simulations regarding risk management, requirements management, verification and validation, and process quality management are of growing importance.

3.3. The Role of Real-Time Monitoring

In Chapter 2, we established the benefits of adding predictive error detection to the current quality assurance practices, but we have also clearly stated that we do not want a world in which the quality assurance team increasingly restricts team members' activities in order to control quality. Globally, the vision is that quality control should get smarter, and as practitioners, we are all in for better support for managing change and dealing with real-world scenarios. This vision needs to be translated into real-world quality monitoring systems, which should cover many domains: different application domains, both agile and non-agile development paradigms, both new and legacy systems, etc. The primary goal is clear: spend less time on quality control without decreasing the quality or service increase the overall system quality. This leads us to the question of how to efficiently and effectively combine predictive error detection with current control techniques.

In this paper, we focus on an often-overlooked aspect of this question: the role of real-time monitoring of quality-related metrics. In the computer security community, real-time monitoring is already well established. The services detect and classify particular errors and exceptions that are server side triggered by clients; however, they are both designed for the state of a deployed software system rather than providing a runtime experience for the end user. In more traditional software engineering, this approach has not yet found a home; at the very least, construction productivity has not been a focus theme. Available quality monitoring islands within the software lifecycle are too isolated; when used alone, they are incapable of detecting anomalies neither in semantic quality nor in stability and performance.

3.3.1. Importance of Real-Time Data

Real-time data consists of real-time measurements and assessments about the quality of large-scale business processes. Due to their immediacy, real-time data convey the best current information on how business processes are being executed, although that information may be based on selective sampling of a subset of transactions. Real-time assessments about process performance often involve quantifying process metrics in comparison to pre-established thresholds or to predicted levels, the latter obtained either from an open-loop forecast based on historical patterns or from a closed-loop forecast that uses the current value of the process metric in conjunction with historical prediction relationships. Real-time data typically involve process metrics that are monitored at relatively frequent intervals and thus may be expected to change substantially from one monitoring interval to the next. Real-time monitoring offers the ability to immediately detect and react to shifts in the process, either to address an undesirable shift or to take advantage of a favorable shift. The availability of real-time data is one of the major innovations in today's large-scale business processes. Incorporation of real-time data into the decision-making process can enhance enterprise enablers of agility, capability, and adaptability during all types of operational activities. Real-time data can be used to address ad hoc issues or to conduct operational quality assurance in an autonomous fashion that is efficient and effective, utilizing the data to continuously pareto the relative importance of the business process execution, based on the costs of poor process quality.

3.3.2. Technologies for Real-Time Monitoring

In recent years, the popularity of the term “Internet of Things” (IoT) certainly contributed to the exponential growth of technologies for real-time monitoring. Essentially, IoT refers to a network of objects and devices that can be connected to a wireless network for the exchange of information and to actuate certain behaviors. The term IoT is particularly associated with the emergence of inexpensive sensors and actuator devices that can be embedded into different types of physical objects or products that are used in the daily routines of consumers, companies, and governments. These devices continually produce and transmit data to cloud-based real-time monitoring systems at low operational costs through communication protocols such as Bluetooth, Zigbee, WiFi, or 4G/5G. Due to this trend, it has become relatively easy for businesses and organizations to engage in real-time monitoring of micro-level data streams associated with production or service processes, such as product conditions, customer routines and behavior, and service agent interactions.

There is currently an abundance of IoT-based commercial solutions tailored for smart quality assurance (SQA) purposes developed by a new generation of tech-start-ups. Enabling technologies include advanced sensors, automatic identification, tag and sensor fusion, augmented reality, lab-on-a-chip, ultra-low power devices, and even artificial mobile agents. These available solutions are accompanied by a set of dedicated development platforms for cloud-based continuous data acquisition and real-time data analysis and management. Still, one of the big challenges for sensing technology developers, scientists, and engineers is how to design easy-to-use, low-cost, off-the-shelf sensors or sensor networks for innovative new applications, especially those that are new and have not yet been defined.

3.4. Predictive Error Detection

Quality assurance (QA) activities are performed mainly before and after software development. We shall consider another QA activity that can be performed during software development as part of the continuous delivery (CD) process: detecting possible errors using predictive models is a step beyond just monitoring software development. This approach has been named predictive error detection (PED). Using large amounts of data such as software development configurations, historical defect reports, and team behavior evolutionary trends, it is possible to create predictive models to be used during the software life cycle to guide software development and CD. Such a pedagogical tool mapped implementations of collaborative specification and agile programming as feedback loops. Quality assessment is not restricted only to post-delivery periods. During the software life cycle, it is possible to learn from and continuously improve the development process considering its quality attributes. Predictive error detection (PED) consists of using predictive models to identify possible stages/cases of an (agile) project, or team member, that will lead to diminishing the software quality, either by using defect density estimations or alerts for error prone modules and/or developers contributing to those modules. By doing this, developers will be able to better manage the software life cycle by anticipating possible problems and problems mitigation actions. The potential benefits of PED are: help on producing more reliable software, avoiding and anticipating errors at an earlier stage of the software life cycle, for example, before a sprint is finished or a software version is released.

3.4.1. Concept and Methodologies

We live in a world where activities performed by people, machines, networks, and devices are continuously growing, in number and complexity, and are constantly connected with each other. Privacy and security of these activities is of utmost importance. Accidental and malicious errors can lead to catastrophic situations with devastating repercussions. In the digital world these activities either have a larger digital footprint than the degree of damage, or are at the origin of digital twins capable of simulating their digital transformation at every instant. It is therefore possible to have a real-time monitoring of critical properties and at any time interval further analyses to find what could possibly go wrong. The data collected can also be used to train machine learning solutions in understanding the malicious efforts to permit the timely detection of future attacks.

Error detection and predictive safety analysis are two of the main research areas in the field of quality assurance of complex, realistic distributed systems, and that led recently to the Smart Quality Assurance concept. The main objective of predictive safety analysis is to understand what could possibly go wrong in the execution of the system under scrutiny. Predictive safety analysis has the goal of assuring that what went wrong corresponding, or potentially corresponding, to the violations of these properties occur very rarely. However, there are situations in which errors resulting in bad executions happen. These are typically situations regarding the modeling of the system that we easily identify a priori, or involving a node that is not correctly behaving but which is not detected and that is not responsible of the system being not able to satisfy the property under scrutiny.

3.4.2. Machine Learning Techniques

Predictive error detection leverages machine learning techniques to build predictive models tailored to different subsets of data from the monitored product attribute space. In order to capture patterns for all product variations, these models can consist of decision trees, neural networks, regression splines, nearest neighbor regression, and other learning machines integrated into a hierarchical architecture. The building blocks of this hierarchical architecture are predictive models that use the exact same input variables but predict different subsets of target attributes or different targets of the same attribute.

The main criterion for evaluating prediction accuracy is based upon data minobjects lying within a prediction cell. A split of a model's input variable range is produced when there exists a target attribute for which a prediction error is committed by enough minobjects. As for all pattern recognition problems, two main questions arise: "How many models (and which type) should we use?" and "How do we build and update them?" To solve these questions, we developed two general methods: The decision-tree package provides a set of methods for automatically building decision-tree models. The second method, called FOCUS, adopts a cache-based approach for building and updating various sectors of feedforward neural networks. The above two software packages only deal with the first problem, as the need for building an ensemble of models is left to the user, based upon the characteristics of the problem at hand. We briefly summarize each of the two techniques.

3.5. Integration of Monitoring and Detection Systems

Quality assurance is not simply an event that happens just before releasing a software product, but a continual process that begins at the earliest phases of product development. During each of the phases of the software lifecycle artifacts are produced to articulate the intentions, expressions, and testing of expectations that define the objectives of the developers and stakeholders of a system. It's well known that a computer system cannot be fault free. There are different types of faults that occur during a system's lifecycle. Many of these faults can be predicted and detected at development time, such as specification and design errors, and design forcing user errors, while others arise during the delivery and operational phase and, these too can be typically predicted and monitored, such as run-time errors. Early detection means increased reliability, understandability and maintainability and reduced delay, costs, complexity, and debugging time.

Fault detection and monitoring techniques such as user monitoring, error monitoring, input/output space monitoring, dynamic execution, model monitoring, etc. detect different types of faults during different phases of the software lifecycle, some at development time, others at delivery time, and others still at operational time. Different techniques may sensibly complement each other and thus improve software quality in a more optimal way. Current research focuses on applying different fault detection techniques independently without much thought on their complementary use. Real-time monitoring, testing, and evaluation procedures activated by dependable monitors may be used for pre-planned quality assurance at any point in the systems lifecycle. They also instigate the analysis and improvement of product/document quality. Predictive fault detection techniques provide not only an indication of a potential fault occurrence in the future, but also the expected time interval for that occurrence.

3.5.1. Framework for Integration

A fundamental idea behind the integration of monitoring and predictive error detection systems is to utilize only data volume, which is normally sufficient for monitoring systems, and augment that data with information found in predictive models. Our belief is that by joining these systems together, the operational costs of the monitoring task are significantly reduced. The monitoring system benefits from the predictive system through the discounting of a large amount of “normal” data, thus focusing only on anomalous parts. Conversely, the predictive system benefits from the results of the studies that are normally made to build the predictive models. In most production discourse processes, only data volume is used for that purpose; it is rare to see a predictive model which takes into consideration parameterized information about the data being used, such as bigram relative frequencies, past age of the data being analyzed, and other feature-information already used in dialogue systems. The proposed framework consists of a real–time monitoring system as an upper layer that sends data to either a predictive anomaly detection system, or directly dispatches it to the upper level of a low-to-medium latency predictive-error detection system, which sends detections to the application, then to a decision layer that decides on how to react to those errors; the role of the integration suggested above can be used to assist either the upper monitoring system, or the predictive task systems. That is the proposed data flow for a closed-loop predictive monitoring system – a smart quality assurance framework.

3.5.2. Challenges in Integration

Integration to complete Smart Quality Assurance faces a number of challenges: Organizations often monitor quality in a variety of ways, not all of which can easily be integrated. They may already have invested in their existing monitoring systems, and not want to be forced to integrate with a more advanced central predictive error detection paradigm, creating a potential for vendor lock. Different monitoring components may utilize differing quality metrics; composability of monitoring components and of the predictive component itself is desirable but often difficult. For example, monitoring may be at a fine grain on individual outputs, even if prediction is to a coarser grain, possibly on sources of sets of outputs from a single data source. Logistics is also an issue. The available infrastructure on site may not be any more conducive to supporting such a level of integration than it is to directly incorporating quality measures into the underlying processing pipelines. There may be politics involved as well; resistance may arise on sites whose existing monitoring systems may already have lobbying power. Other potential sources of resistance are management with vested interests in their particular quality monitoring subcomponents - perhaps they are seen as experts in particular monitoring areas such as data cleansing or shot length. Integration to complete Smart Quality Assurance is, of course, also the most beneficial. Predictive error detection and monitoring are most powerful when they act together. Error monitoring can act to provide feedback information confirming the predictions of the predictive systems; such predictions would be suspect if they do not agree with the feedback from low-level monitoring. The reverse is also true; prediction can be used to highlight specific monitoring areas needing close attention, or making recommendations at coarser level than would be provided by actual monitoring.

3.6. Case Studies

The integration of CAM and PRED enables Smart Quality Assurance, which achieves real-time visibility and error prediction while preserving low overhead. It is effective on all available executable files without source code access, enabling large-scale deployment in a short time. In addition, it is efficient, with both negligible runtime overhead and minimal debugging error concealment. We illustrate these points with case studies of the production use of Smart Quality Assurance in three sectors, which face different time-to-market pressures: manufacturing, which employs embedded systems for manufacturing process control; software, which develops software that interfaces directly with customers; and healthcare, which implements software-assisted treatments that may impact patients' safety. Centralized systems are critical for manufacturing process control, as they regulate the behavior of equipment and process parameters. In practice, small debugging errors hide in tested code and cause production accidents not reproducible in testing. The adverse effects include stop production downtime, monetary losses, and poor product quality. We integrated Smart Quality Assurance with a Scada system, covering 80% of reported defects and deployed in the live environment of a semiconductor manufacturer. Its production safety is also improved by the use of digital twins that minimize the debugging error impact. Time to market is of utmost importance in the software sector, where products are developed and debugged iteratively, with each iteration involving the release of a new version to users. Despite the large number of daily software releases, debugging errors still evade testing. These concealment bugs impact users' experience and company revenues. We integrated Smart Quality Assurance into a web tracking and reporting system of a digital insurance company. The production environment logs millions of lines of API calls issued from clients' browsers, connected with the backend task execution system. High-volume fraud detection is typically deployed in the insurance context on cloud-based servers whose inefficiency may affect customer experience.

3.6.1. Manufacturing Sector

Research in wireless sensor networks (WSNs) for applications in the context of process control has already reached a mature state and is continuously growing. Recent publications cover a large variety of topics and focus on state-of-the-art applications in industrial environments. Commercial wire-line monitoring systems have appeared at multiple industrial sites for years and the evolution toward a completely wireless solution is necessary in future designs. Guided by the outlined research perspectives, smart solutions to quality assurance in the context of process monitoring, predictive error detection, and complaint management are implemented for the manufacturing sector, software development, and healthcare solutions.

The presented application for manufacturing addresses in particular the electrical localizing of faults and errors in brownfield processes and system devices. After an extensive state-of-the-art survey on Fault and Error Management systems in manufacturing in general and PCB assembly in special, it was decided to realize a conceptual solution with supportive wireless modules. The experimental validation is performed at an associated PCB assembly environment currently using a wire-line embedded solution. The integrated approach consists of selective additions to the manufacturing equipment regarding Composite Sensors, Monitoring Software, and a set of Wireless Smart Quality Modules for packaging and functionality which embed a remaining useful life algorithm. One example for the manufacturing sector is the development, implementation, and validation of a smart quality assurance solution in the context of PIN-in Hole processes in the PCB assembly. The testbed application is based on an advanced in-house developed wire-line module using an embedded probe-surface attached fault detection system.

3.6.2. Software Development

Predictive error detection has been recently introduced also for the Software Development domain: some pioneering studies have explored the possibility of predicting the module defects during the development phase of software systems in order to allow the software engineers to take the required actions to eliminate the fault before it affects the final product. Real-time monitoring, on the other side, have been mostly used and applied for in-use software systems with the goal of improving the software reliability in the operational phase of the system for more than two decades already. Monitoring has been successfully applied to different types of software in multiple application domains: distributed systems, middleware, web-based applications, transactional systems, component-based systems, and mobile applications. The approach supports a number of application domains: Java Enterprise applications, C# web-based applications, asynchronous JavaScript applications. The approach has been initially introduced for web-enabled transactional applications, and extended for Java-based enterprise applications, serving engines, and service-oriented applications. Multiple tools and infrastructures have been developed for monitoring web-services, distributed systems, user-centric and component-based services, and Java Enterprise Applications.

The integration between real-time monitoring and predictive error detection offers benefits in both directions: it allows the early detection of problems that may be hard or impossible to be captured with on-line techniques because of their low occurrence, it also allows the dimensioning of the monitoring phase, through the proper resource allocation, discarding the possibility of performance problems that are in practice not persistent like in services that suffer of bad performance during only short periods of time. The integration supports the dimensioning of the monitoring phase, through the proper resource allocation, discarding the possibility of performance problems that are in practice not persistent like in services that suffer of bad performance during only short periods of time.

3.6.3. Healthcare Applications

In healthcare, models are used not only to detect errors or problems in the data but also to monitor and measure data on key healthcare system performance metrics. In this sector, data is captured, modeled, and monitored on the key metrics including patient satisfaction survey results, hospital readmission rates, hospital-acquired infection rates, patient mortality rates, and resource consumption metrics. New non-clinical models and innovative technology, including predictive multi-source data acquisition, imaging, and analytic solutions powered by artificial intelligence and predictive analytics, are being introduced. These new solutions go beyond the traditional clinical model and utilize advanced computational capabilities to model and predict patient outcomes, as well as enable the capture, collection, processing, and analysis of the entire clinical environment – patients and their family, healthcare professionals, hospital premises, equipment, and technology – to understand data interdependencies and provide real-time situational awareness.

Some examples of these advanced predictive solutions include solutions used to model and predict patient satisfaction levels while they are still being treated at the hospital. These new solutions work by capturing non-clinical real-time data on patients' health state and body language measured by facial coding technology, healthcare professional behaviors such as empathy and validation measured by speech analytics technology, hospital environment conditions such as air quality, noise level, and water quality measured by IoT devices, and hospital resource consumption metrics such as hospital staff turnover and cost per patient episode. They analyze the patient, hospital premises, and hospital resource consumption data in real-time, and correlate the data to each patient-specific question to understand all the drivers of patient satisfaction, especially for the questions with a low score.

3.7. Benefits of Smart Quality Assurance

The Smart Quality Assurance (SQA) Model integrates real-time monitoring and predictive error detection in quality assurance, shifting the focus from after-the-fact debugging to pre-emptively preventing the creation of defects. The Smart QA Center enables analytics-driven real-time monitoring, correlating application-specific metrics with business outcomes to serve as a predictive alert system, triggering closer investigation at the appropriate times. Key benefits of this model include enhanced productivity of project teams without compromising quality, significant time and cost savings due to the reduction of rework associated with post-implementation defects, and improved customer satisfaction. By working in parallel with development and release processes, the Smart QA Center provides active intelligence and safety net roles for the project teams. The SQA approach also enables better prediction of error detection rates in the delivery process and better planning of resource allocations. In fact, the data-driven nature and early warning predictive ability of the SQA Center's work-product and defect analytics can help build financial models for delivery.

Enhanced Efficiency

The SQA Center's presence during each delivery step allows breach of agreed levels of quality risk without letting it fester to the point of costing another project team significant rework due to defects manifesting post-implementation. Additionally, the SQA Center enables each project team to focus on delivery as they are busy working on other task details. Why is this important? Well-documented studies in the software industry indicate that costs of fixing a defect after systems go live tend to be significantly higher than for fixing the same defect during the development process. By preventing defects from getting living water during the delivery process, the SQA Center helps save time, money, and people’s peace of mind.

3.7.1. Enhanced Efficiency

The key aim of any initiative that seeks business excellence, such as Smart Quality Assurance, is to improve the capabilities and operation of a business. As mistakes in a business come at a price and can be detrimental to both the employees and the organization, having a Smart Quality Assurance enables better informed decisions that reduces errors as well as precautions to help alleviate them at any stage. Having a Smart Quality Assurance system in place can help companies enhance their efficiency in areas such as data validation, report generation, trend tracking, automated decision making resulting in promotion management, and anomaly detection. Smart Quality Assurance allows organizations to closely monitor their performance, by performing validation at regular intervals, rejecting samples that do not meet predefined criteria. This helps establish consistency between processes as specified by the organization. Generic reports are also created that compare performances of each strategic business unit or key accounts and determine if any corrective preventive action is needed. A Smart Quality Assurance system also keeps track of its performance on a regular basis to check performance trends with respect to parameters defined. Any department that is not performing satisfactorily is flagged and assigned further monitoring. This helps in providing data inputs that either confirms or rejects warning flags that are in place. Frequent and efficient reminders help in correcting data, paving the road for faster decision making.

3.7.2. Cost Reduction

Apart from the enhanced efficiency, Smart QA provides benefits for the partners, clients, and the business, including the decreased overall costs. Different partners, clients, and groups are trying to optimize the development processes. In numerous cases, at the end of the chain, the client is the one who will invest more resources in the issue resolution. Thus, several partners invested in automation and other methods to decrease costs at various stages of development. The last phase of the chain, integrating quality assurance, has not been touched upon as much yet. After starting to position the role of Smart QA in testing and developing, it can create and design test automation in different and easier ways. By catching bugs earlier in the process, it is possible to cut testing time and reduce the final product cost. The Smart QA service can design test strategies that are meant to template the final product-testing life cycle and thus optimize it. Smart QA can keep testing in focus, minimizing the overall costs, both for the client and the testing agency.

Such actions will not only dynamically minimize the financial demands from production testing once the product goes live, but will also generate long-term optimization of the quality assurance resources during the product life cycle. Due to already applied tests, it will also allow tracking issues that generated financial losses to the clients' and partners' businesses. The Smart QA service can minimize the generated costs in five different manners: reducing the time spent on testing by creating test strategies based on machine learning; utilizing historical data to template and automate future tests on existing products; reducing the amount of false positives detected by using the data from the products and services running in production; reducing the test cycle time by detecting issues early in the development cycle; reducing iteration time by detecting issues that are delaying the production deployment on a regular basis.

3.7.3. Improved Customer Satisfaction

To maximize revenue, all companies need to engage and maintain their customer base, which is not easy in a highly competitive market environment. A common way to respond to competitive pressures and avoid customer churn is through the introduction of defect prevention and product/service improvement programs. While these initiatives are effective in terms of customer satisfaction, they are also costly. In this context, it is important that the quality processes introduce the least dissatisfaction to the client. For a company's customers, quality is often reflected in product performance – reliability, convenience of service, and attention to product safety. If the customer is let down by products and services, accident and reacquisition costs count those companies among the trading losses and record the efforts involved in quality assurance as lost expenses. Customer satisfaction is of critical importance for business success. Satisfied consumers, once conditioned by favorable experiences, become the most reliable asset of the company since they are less price-sensitive and more likely to recommend the product or service to others. A customer's commitment is the most convincing sign of satisfaction, and therefore it is understood as the main objective of any specialized marketing function. Research indicates that companies providing high quality outperform competition in both profits and sales growth. They tend to have higher revenue margins, greater resultant profits, and a faster growth rate than their competitors. Companies that place a strong emphasis on customer satisfaction significantly outperform those with less emphasis in areas such as sales, revenue, profit margins, and profitability.

3.8. Future Trends in Quality Assurance

The companies that are most successful in the future will be those that best manage resources harnessing effective planning and control resources. To gain better control, managers will rely on better, more timely information about the performance of their processes, people, and systems – including revenues and costs associated with each product or service category and with each step of the process for delivering it to customers. They will also be able to have objective, timely information about their customers' preferences, behaviors, needs, and willingness to pay. Business systems that can provide managers this level of information will certainly start to become available; principles outlined here will be put into practice in a variety of forms in dozens of companies.

AI and Automation

Integrating real-time quality assurance with error prediction will require a good deal of organizational innovation, as well as a fair amount of technological innovation.

Advanced process-centered architectures will be developed. Developments in intelligent agents and neural nets. Better tools for knowledge acquisition, storage, sharing and retrieval, tools for query answering and data mining. The new generations of concurrent-like and visual programming languages. New sophisticated rule-based languages and other expert-system ai shells. Distributed databases that allow organizations to readily share and exploit the huge amounts of data they will have on customers and their purchasing habits, preferences, and responses. Productivity enhancers and debugging and error detection tools for hard and soft products.

These will all help engineers and managers design and implement holistic business process structures that engage, satisfy and delight ever more discerning customers. Fill their days with interesting and challenging activities while helping them make a comfortable living, support their families, educate their children, and contribute to their community. The end result will be a world of work characterized by collaboration, trust, and celebration, instead of fear and distrust. Reducing the negative impact on the environment of business and personal activities will also be a high priority across the world. Quality assurance methods that are integrated with real-time process monitoring and predictive error detection will help effect this change.

3.8.1. AI and Automation

A multitude of new tools have rapidly entered the world of software testing including AI driven capabilities, data intelligence and data warehousing technologies. Automated data intelligence can identify and review interfaces that are in use and how often which can assist with the design of automated tests. Management teams also require Data Data Warehousing or Data Lake type environments to access and track the company products and business performance overview. If using an offshore supplier, a clear indication of hourly and overall costs as well as overall productivity is essential.

AI driven capabilities will soon offer tools to identify and document unexpected changes in workload(s) and offer patch designs with internal tests. Automation tools have been developed to assist in the creation of progress reports and identify standards and procedures that are causing project delays and budget overages.

Predicting areas of risk is becoming automated with products that identify and report actual application usage and component interface paths. These products derive their analysis from monitoring and analyzing production supplies in an "as is" state along with tracing current usage patterns both totalling number of various paths as well as tracking for spikes occurring during specific events. These type of early detection systems along with monitoring tools to ensure that the proposed patch or upgrade is following the original change design are beginning to be incorporated into the methods and tools as part of normal operations and being utilized in conjunction with disciplined software development lifecycle pipelines supporting DevOps initiatives and processes.

Fig 2 : AI and Automation

In actuality, these systems are utilizing event driven analytics in a predictive rather than reactive fashion and identifying unpredicted abnormal performance or risk areas that warrant immediate attention from the business or IT as part of the normal operations.

3.8.2. Data Analytics Evolution

From Digital Analytics to Predictive Error Detection Inside Business Processes of the Enterprise

The domain of enterprise operations data analytics has evolved in a few main steps that are presented below. Digital analytics augment the traditional activity of business reporting and introduce a real-time dimension to relevant enterprise business data. The next step in this evolution is concerned with the semi-automation of tedious and repetitive reporting tasks that included the generation of standard reports in the Supply Chain of eCommerce distribution. The third step associates automated business process reporting on an enterprise Business Intelligence server with advanced BI visualization dashboards and it is the path that brought data analysis to business domain experts. Report automation has grown wiser through the integration of Business Process Management Foundation models with company data in Business Process Analytics solutions. This opportunity led to the concept of Intelligent Business Process Automation, with the emphasis on data and business domain knowledge, where data is digged only for elaborating highly-qualified process logic to yield the smoothest process data analysis possible. The fifth step in the evolution of enterprise data analytics is Enterprise Activity Automation which shifts the paradigm to the next level by predicting the error scenarios related with company KPIs and links alerts with business optimization modules.

These trends are possible due to disruptive changes in Data Storage with Data Lakes, Open Source Data Pipelines, Streaming Process Data Storage and Processing and massively adopted Cloud Services for Data Visualisation, all combined with the internal democratization of Data Analytics in the business milieu around trusted Data and BI Experts.

3.9. Ethical Considerations

There are various ethical considerations associated with integrating real-time monitoring and predictive error detection in software development workflows. One key issue is data privacy. Because predictive error detection requires historical data about the software and its development, violations of data privacy could occur if users are not made aware of how this data is being generated or used. Another ethical challenge within this data collection process is potential bias in predictive models. If the data is skewed in any way, detection bias can cause some features or components to experience an unfairly higher or lower amount of error predictions.

Data Privacy

There are two major areas of concern regarding sensitive data in predictive error detection models: the act of building predictive models and the use of the predictive models. Eventually, predictive models will likely be built using large quantities of data from many different sources and organizations. There are numerous aspects of the development process that could produce sensitive data, such as release notes, bug trajectories, user stories, the code itself, and even shortened snippets of code. This data could potentially include information regarding security exploits or sensitive user information, as well as data regarding sensitive projects that may be classified in some way. It is also possible for other developers' code to include sensitive information. When training the predictive models, a general model would be trained using a massive quantity of public data from various developers. It is possible that this model could be inadvertently trained on sensitive data, and as a result, any use of the model produces the risk of sensitive data leakage, even if the prediction itself is not conducted on sensitive data.

3.9.1. Data Privacy

Data privacy concerns arise when monitoring the events of use of user applications are collected from user environments, for example, through a monitoring lightweight agent. Such monitoring embraces monitoring and predicting according to the implicit predictive models these data are entrusted to. Luckily, the requirements are set during the creation of the agreement when users give their consent to behave according to the limitations imposed by the agreement. What to do in case of access by external third parties to sensitive data or inappropriate use of it by a software company responsible for maintaining user applications? Formulating the expectancy of an average application user, we can state that the probability of intrusions into their privacy is evidently lower for using the application with no errors than not abiding the restrictions of the agreement. Customers' lower expectancy of intrusions for the case when the user application works correctly contributes to the probability of not intruding into users' privacy despite that the system can potentially collect sensitive data.

These utilizations have different ways of influencing the expectancy of a software company's behavior. The first one uses the pre-agreed legal course of action while predicting on confidential data and the latter makes an attempt to affect the expectancy using the behavior of the software company. However, while using the pre-agreed course of action, which suggests that the behavior is conditional on taking the side of the software solution's user, the expectancy is influenced by the probability of an external company's minimizing monetary losses for breaking the terms of the agreement with respect to potential profits for doing it. The law has to help do it in the interest of application users.

3.9.2. Bias in Predictive Models

During the era of "big data", especially in the ML/DL applications, questions regarding the societal impact of model use continue to gain support. Automated model decisions are often described as "black boxes" because the output is not substantiated and understood well enough. Increasing scrutiny in many government decisions have triggered a demand of explainability of predictive models in societal, financial, and political domains. For example, a given predictive model for job application rejection and approval decision is often reviewed for biases due to characteristics such as age, gender, disability, experience etc. Therefore, quantifying and validating predictive models for identified bias behavior for specific groups is often desired via fair machine learning which lays down methods and practices to render predictive models as unbiased.

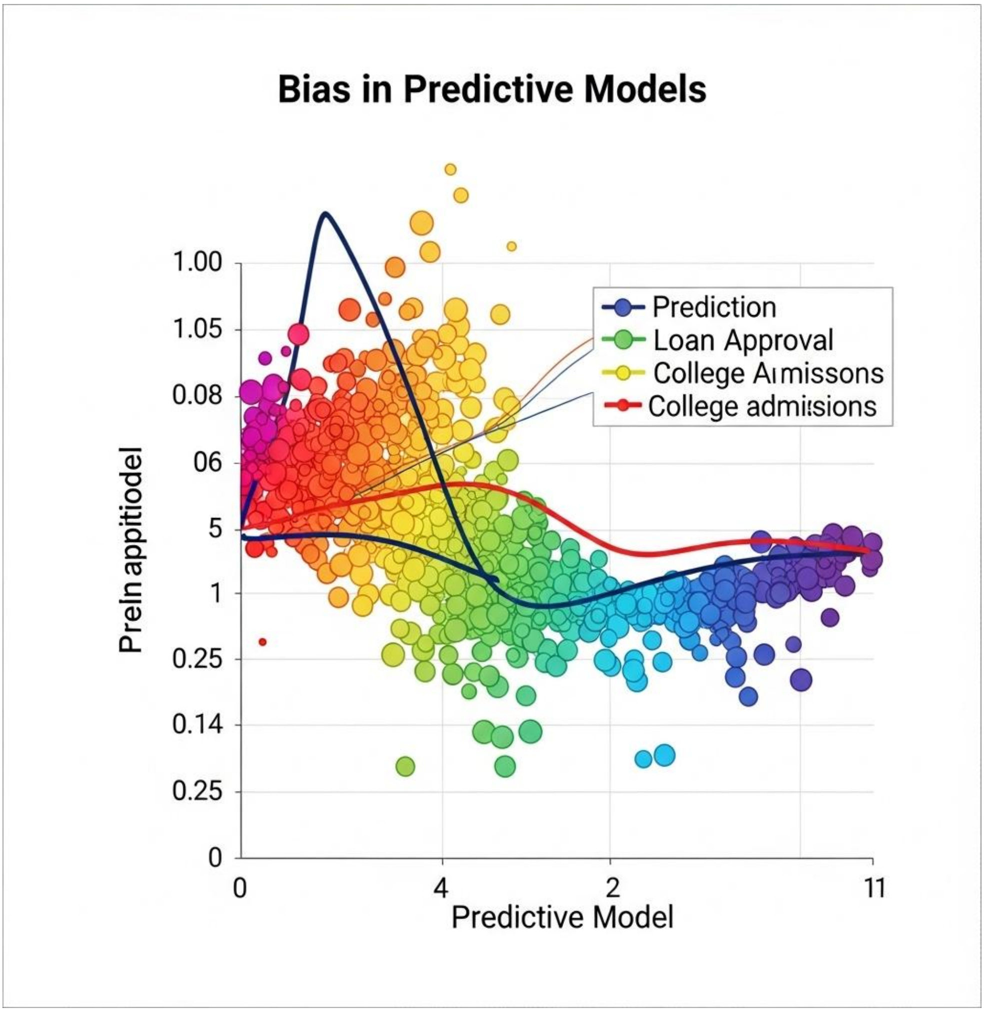

Fig : Bias in Predictive Models

No matter how much the model results are rendered as unbiased, the very nature of assumptions made during model development might result in retrofitting of already existing biases in the tasks. An obvious issue, however, is the choice of the protected sectors of society with specific classes of attribute assignments which are used to be protected against bias. Transitioning to some other types of protected assignment classes could lead to entirely different conclusions and actions. For example, establishing accuracy for any predictive model for credit assignment or default estimation is often a big challenge without the institution's prior bank activity data as models are prone to biases against specific groups.

3.10. Conclusion

The history of product design shows ever higher integration of quality functionality into the very details of product design. As realization becomes more intense and accurately realizes the stated purpose of a product, so users accept increasingly rigid constraints on the means employed in realizing that purpose. These constraints shape the models used for intelligent error detection. Just as design removes errors of the previous design by making assumptions explicit, so the intelligent systems of economy and production take advantage of the information contained in the possible vectors of contract fulfillment, production capability, and contract combination cost to define and satisfy a more restricted class of contracts. The key idea is a feedback model of the intelligent economy and a feedback role for the independent intelligences at the top and bottom levels of an intelligent system of economy and production.

Once upon a time, humans provided the learning ability necessary to carry out job assignment and scheduling. Better communication and on-line scheduling is one way to allow lower level programmers to take on more responsibility, relieve some of the burden of the initialize system and at the same time take advantage of some of the on-line learning abilities possessed by intelligent expert systems. As on-line scheduling becomes enhanced with expert system supervision, using predictive error checking on-line help becomes an essential part of future scheduling approaches. In fact, interactive, computer controlled scheduling seems to be an ideal application area for expert systems.

This paper has discussed the critical elements that must be addressed for a hybrid on-line and expert system animated scheduling capability providing predictive error detection to be successful. Through a variety of examples, we have honed in on the motivation for the approached described above, the type of architectures that can be used and the key concepts needed to implement that architecture. As expected, we have as yet not painted the complete picture. Given the time pressures of financial concerns, and the job shift and maze optimize coupled with the need for on-line help, we feel the described approach has valid components for producing something needed in the real world.

References

S. Martínez-Fernández, J. Bogner, X. Franch, "Software engineering for AI-based systems: a survey," in *Software Engineering*, 2022. [PDF]

D. Carnerud, A. Mårtensson, K. Ahlin, "On the inclusion of sustainability and digitalisation in quality management–an overview from past to present," Quality Management & …, 2025. tandfonline.com

S. Najihi, S. Elhadi, R. Ait Abdelouahid, "Software Testing from an Agile and Traditional view," Procedia Computer Science, vol. 200, pp. 123-130, 2022. sciencedirect.com

I. Ozkaya, "Application of large language models to software engineering tasks: Opportunities, risks, and implications," IEEE Software, 2023. ieee.org

Y. Gadhiya, C. M. Gangani, A. B. Sakariya, "Emerging Trends in Sales Automation and Software Development for Global Enterprises," IT Journal of ..., 2024. researchgate.net