SmartDelta: Automated Quality Assurance and Optimization in Incremental Industrial Software Systems Development

Mehrdad Saadatmand
RISE Research Institutes of Sweden
Västerås, Sweden
mehrdad.saadatmand@ri.se

Eduard Paul Enoiu
Mälardalen University
Västerås, Sweden
eduard.paul.enoiu@mdu.se

Holger Schlingloff
Fraunhofer FOKUS
Berlin, Germany
holger.schlingloff@fokus.fraunhofer.de

Michael Felderer
University of Innsbruck
Innsbruck, Austria
michael.felderer@uibk.ac.at

Wasif Afzal
Mälardalen University
Västerås, Sweden
wasif.afzal@mdu.se

***Abstract*—A common phenomenon in software development is that as a system is being built and incremented with new features, certain quality aspects of the system begin to deteriorate. Therefore, it is important to be able to accurately analyze and determine the quality implications of each change and increment to a system. To address this topic, the multinational SmartDelta project develops automated solutions for quality assessment of product deltas in a continuous engineering environment. The project will provide smart analytics from development artifacts and system executions, offering insights into quality degradation or improvements across different product versions, and providing recommendations for next builds.**

***Index Terms*—Research project, ITEA, EUREKA, software product lines, software variants, continuous system engineering, software quality**

I. Introduction

Industrial software-intensive systems are typically not designed and built from scratch for each new customer and order, but rather as increments over an existing product or as a modified version tailored for the needs of a particular customer, market, or region (e.g., in product line engineering). Similarly, for a single product and considering a continuous integration/continuous delivery context with frequent builds and commits, a system gets built incrementally and iteratively. This results in many intermediate builds and versions (e.g., in agile development). At Google, for example, on a typical workday 16,000-24,000 changes are made to the codebase [1]. Such an incremental development approach essentially leads to the emergence of different product versions, whether as intermediate versions or final ones ready for delivery. This means that over time companies end up having a big range of product versions and components that need to be maintained, and analyzed for re-use possibility whenever a new product is to be built. Moreover, each version can have different quality characteristics, for instance, due to addition, removal, or modification of some features or components. This, however, poses important problems in the development of industrial software systems, some of which are highlighted below:

- Usually, the process of re-use analysis (to identify what artifacts can be selected and re-used from previous product versions, and what new ones need to be implemented) is still done manually. This is prone to human errors and not scalable.
- A manual re-use analysis also means that the process is heavily dependent on the experience and availability of those key engineers who have knowledge of how previous versions were built and why certain design decisions have been taken.
- As new customer requirements arrive requesting new features to be added to an existing version of the product, system complexity also grows. This is often at the price of quality, sacrificing key characteristics such as performance and security. This issue can well exacerbate considering the constant pressure on companies to shorten the time-to-market to be able to stay ahead of the competition. In this regard, there are many incidents where, for instance, patching a bug has caused additional issues, at times making the system even crash and become unresponsive (as an example, consider a recent Windows 10 update [2]).
- Guaranteeing quality characteristics which are technically referred to as extra-functional or non-functional properties (EFPs or NFPs) is also a huge challenge in itself: EFPs are mostly interdependent and cannot be analyzed and addressed in isolation [3]. For instance, adding more security features such as encryption algorithms and components to a system can impair its performance. Vice versa, optimizing a system for performance may have negative impacts on its security or energy consumption. Therefore, in automating re-use analysis and making design decisions for development of new product versions, it is essential to have solutions for trade-off analysis to assess the impact of different EFPs on each other, and establish a desired level of balance among them.

To reduce the feedback cycle and optimize the development process, currently there are three main relevant paradigms:

i) product line engineering, to enable efficient production through reusing a set of shared assets across different product versions; ii) agile practices, advocating faster and shorter development cycles to facilitate accommodation of new customer requirements and providing faster feedbacks; and iii) DevOps, which is seen as complementary to agile methodologies and as a development method aimed at bridging the gap between Development (Dev) and Operations (Ops). However, all three paradigms fall short in addressing the above-mentioned industrial challenges, in particular with respect to guaranteeing quality characteristics across different product versions and builds. What is needed is an automated re-use solution that is driven by quality. This solution should be able to identify quality improvement or degradation patterns to construct trends of software quality characteristics across previous versions, derive mandatory quality assurance measures aligned with the identified quality trends, and ultimately provide recommendations for the next builds to improve or at least preserve the quality characteristics of interest.

SmartDelta[1] *is an ITEA3 international project which aims to build automated solutions for quality assessment of product deltas in a continuous engineering environment by providing smart analytics from development artifacts (e.g., source code, log files, requirement specifications) and system executions, offering insights into quality improvements or degradation both in and of different product evolutions, and providing recommendations for next builds.*

It should be noted that the frequency and timespan between each build and version can vary significantly between different domains and industries. Examples are frequent nightly builds in enterprise software applications, in contrast to months or years between subsequent versions in the automotive or railway domain. Therefore, in this project, we use the term *delta* to refer in a general manner to any product version that is built as an update to a previous one, regardless of the frequency of builds and whether it is an intermediate version or final ready for delivery.

To solve the aforementioned challenges, SmartDelta develops solutions for

- verification and validation of quality characteristics in industrial-scale systems based on techniques such as static analysis, formal methods, and model-based testing;
- automating re-use analysis by machine learning techniques such as natural language processing (NLP) for processing of different development artifacts such as textual requirement specifications and test reports; and
- automated trend analysis and build recommendation with respect to quality characteristics using AI/ML for pattern recognition, optimization, and fault prediction techniques.

To facilitate industrial adoption of SmartDelta solutions, the project also builds a dashboard for visualization of software quality. This can greatly aid end-users of the project, such as system architects, and test and quality assurance engineers. Applying SmartDelta solutions enables them to accurately monitor the quality of the products that are being developed. Among different quality characteristics, SmartDelta explicitly targets performance, energy, and security, but is not be limited to these.

Considering the relevance and importance of the project topic for a wide range of industries offering software-intensive products, the project has attracted and brought together various partners from different sectors and market domains with complementary expertise, knowledge, and technologies to develop the proposed solutions and verify their technology readiness levels. In particular, the consortium consists of a well-balanced mix of partners from Sweden, Germany, Canada, Turkey, Belgium, Spain, Czech Republic, and Austria, including industrial use-cases from the railway, telecommunication, logistics and mobility, FinTech and banking, cybersecurity, and enterprise software domains.

The remainder of the paper is structured as follows. Section II provides a detailed description of the technological and business challenges that the project targets, as well as an overview of the solutions that the project intends to develop to solve those challenges. In Section III, the work structure of the project in terms of different work packages and their relationship and inter-dependencies is explained. Section IV describes the targeted industrial impact of SmartDelta, and in Section V SmartDelta is localized in the context of related projects. A summary in Section VI concludes this work.

II. Challenges and Solutions

Quality attributes are aspects that have to be addressed holistically from the early phases of the development process and ensured across all phases of agile development (i.e., in DevOps practices).

> *[Challenge 1] Traditionally, quality attributes are often treated after delivery on the code or at infrastructure level with specific changes to the standard product.*

The challenge is to automatically address these aspects continuously from design to operations. Since agile practices promote frequent software deliveries, analysis and verification methods artifacts should be automatically updated in a timely fashion to cope with the pace of the process. As companies embrace continuous integration and delivery, the number of delta version artifacts created increases, resulting in ever-increasing quantities of data.

> *[Challenge 2] The challenge for many companies stems from analysing, visualizing, and transforming this huge volume of data into actionable knowledge that can improve software quality of their developed systems.*

Product line engineering in agile practices is about fast, flexible system engineering that efficiently integrates development, delivery, and operations, thus aiming at quality deliveries with short cycle time to address ever-evolving product variants. Current system development practices are increasingly based

[1] https://itea4.org/project/smartdelta.html, https://smartdelta.org/

on using both off-the-shelf and legacy components, which make such systems prone to quality issues.

> *[Challenge 3] The challenge is to identify quality assurance measures that can be used to demonstrate the essential quality attributes without having to completely redesign test cases.*

A common industrial practice to adopt product lines is through the incremental development of overloaded assets, which are re-used. Deriving a new product from a product line requires re-use analysis to avoid redundancy and to support a high degree of assets re-use, by taking into account quality attributes.

> *[Challenge 4] Traditional re-use analysis approaches lack the automated support for optimization of quality characteristics (i.e., especially when taking into account the trade-offs between different interdependent properties such as performance, resource consumption and security), is time-consuming, and is heavily based on the experience of engineers.*

More often than not, the number of non-functional scenarios to be ensured explodes. For example, in the embedded software domain, the number of system interactions with the environment that are subject to security attacks is increasing and may result in vulnerabilities that can cause not only losses for end-users but also drastic increase in production and maintenance costs, especially if iterations are long and feedback comes late between different versions of the same system.

> *[Challenge 5] Many traditional analysis and verification approaches do not support continuous feedback loops.*

Even if automated testing, nightly testing and continuous integration facilitate more rapid feedback, poor feedback on test results combined with the lack of traceability between quality attributes and artifacts can results in slower or missing feedbacks. This has an impact on quality (for example, a test engineer waiting for feedback on a system change could break the code implementation).

SmartDelta brings together fast analysis and verification solutions through modelling as well as static analysis, test generation, selection, execution and visualization capabilities to enable companies to deliver quality systems with confidence in a fast-paced agile environment. Figure 1 depicts the overall concept of the project.

SmartDelta is focusing on optimizing the analysis and verification activities of a series of product versions and deltas, by automatically creating and adjusting verifiable models directly from development artifacts like requirements, logs, code, test logs, and using these models to check extra-functional properties against system models and generate artifacts such as tests or monitors that can be used later on in the product-line process for future variants. More concretely, to leverage the quality assurance in continuous development of industrial systems, it brings together European and Canadian industrial and academic communities in order to develop and demonstrate SmartDelta technologies ranging from processes, methodologies to tools and demonstrators. The project pursues the following technical objectives (mapped to the SmartDelta overall concept in Figure 1):

1) *Creating and extracting models from different software artifacts as well as consistency checking between deltas into a model chain architecture.* This includes developing techniques for creating and extracting models for different software artifacts from usage scenarios and logs.
2) *Automated consistency checking and validation based on model testing and model checking techniques.* The project brings novel techniques supporting automated verification of delta products through the use of model specifications both at development and runtime.
3) *Developing an automated approach for delta-aware re-use analysis of product line features* as well as trade-off analysis between different extra-functional architectural properties of interest.
4) *Automated test generation and analysis for extra-functional properties* (e.g., resource consumption, security, performance) based on the model specification of the corresponding requirements that target real industrial continuous development challenges.
5) *Intelligent/adaptive and ML-based analysis, visualization and recommendations* for the next system delta versions. Shortening the development, deployment, and feedback loops by using quality checks and trend analysis to provide actionable recommendations and identify vulnerabilities in system models early in the developed product.

III. Work Structure

The work in the SmartDelta project is structured into seven work packages. Detailed specification of industrial use-cases and elicitation of their requirements, establishment of a baseline of use-cases, and set-up of industrial demonstrators for evaluation of project results with respect to the defined key performance indicators are done as part of the work in WP1. The SmartDelta use-cases are depicted in Table I.

In WP2 on model generation and evolution, automated solutions are provided for identifying and extracting different software artifacts representing different product versions and deltas, for creating product models from these artifacts, for maintaining and managing their evolutions across different versions and builds using model repositories, and for automated consistency checking and model validation. From this perspective, WP2 has a close dependency and collaboration with WP1 to identify and extract relevant software artifacts from industrial use-cases based on their requirements and the problem domain.

Development of solutions for automated verification and validation of quality attributes is done in WP3. In this work package solutions such as model-based testing for automated

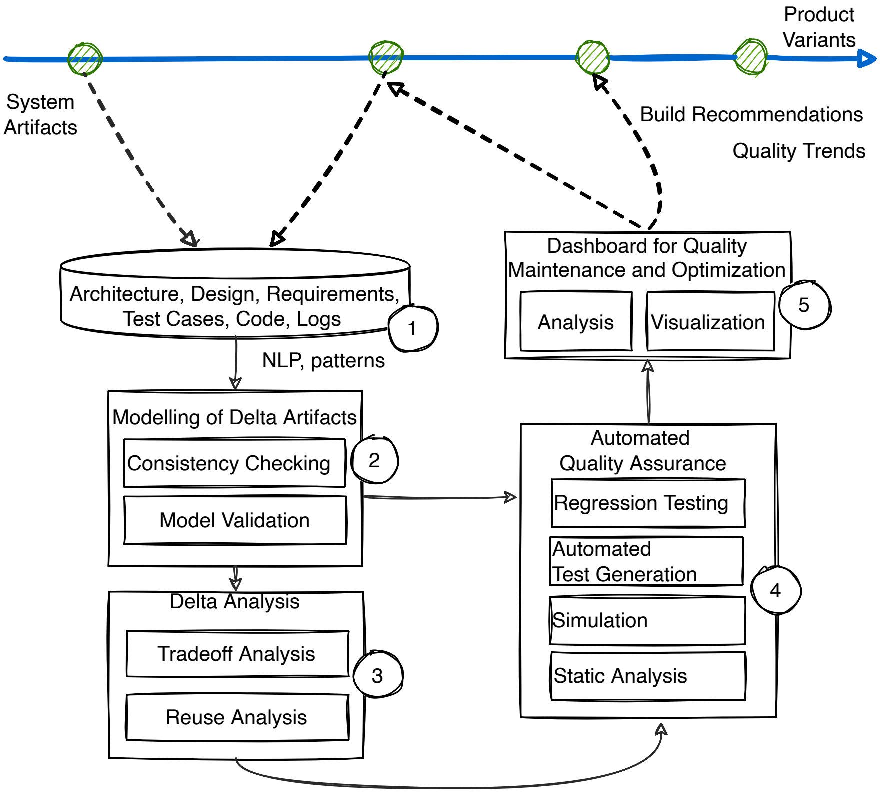

Fig. 1. SmartDelta Overall Concept. SmartDelta is focusing on optimizing the analysis and verification activities of a series of product versions and deltas, by leveraging system artifacts to analyze extra-functional properties and generate tests or monitors that can be used in the product-line process.

TABLE I
SMARTDELTA INDUSTRIAL USE CASES.

Country	**Company**	**Domain**	**Use Case Topic**
Sweden	Alstom	Railway	Quality in agile model-based system and product line engineering
Germany	AKKA	eMobility	Charging communication controller software for electrical vehicle
Canada	eCAMION	eMobility	High quality and cybersecure software in deployable energy hubs
Turkey	NetRD	Telecomunication	AI based fault and performance analysis in cloud communication services
Turkey	Kuveyt Turk Bank	Banking and Finance	Continuous improvement of code quality, security and performance in core banking software
Germany	Software AG	Enterprise Software	Continuous security and quality improvement in enterprise software
Austria	c.c.com	Logistics and Personal mobility	Continuous quality monitoring & improvement in automated traffic detection software
Canada	GlassHouse Systems	Cybersecurity	Continuous improvement of cybersecurity solutions

generation of test cases, static analysis targeting performance, security or energy consumption issues, and tailored and optimized solutions for delta-oriented testing in a continuous engineering context are developed.

In WP4 novel methods are developed to analyze and predict software quality trends across different product deltas, to perform similarity and change impact analysis, and to offer analytics results and recommendations on re-use and design decisions for optimizing desired quality characteristics of the system in the next builds (based on industrial requirements from WP1). AI/ML-based techniques for pattern recognition, prediction, natural language processing, and optimization are of special interest in this WP.

In WP5 a SmartDelta visualization dashboard and integration is derived. The research is focused on techniques for software quality visualization and presentation of the results coming from other work package techniques and tools in appropriate formats for end-users. Integration of the solutions that are developed in the aforementioned work packages will also take place as part of the work in WP5, e.g., through a set of interfaces and APIs. Additionally, inclusion and presentation of training materials and user guides as part of the SmartDelta visualization dashboard will be considered here.

Finally, dissemination of the project results is done as part of the work in WP6. This includes the production of training materials, and the planning for exploitation during and after the end of the project including activities for commercialization and industrial adoption of project results beyond the project consortium. Management of the project and coordination of the activities ensuring successful collaboration among partners and interfacing the communication with the ITEA office is the focus of WP7.

IV. Industrial Impact

Currently, the European software market is extensive (estimated revenue of approximately 100 billion US$ per year[2]), and further growth is expected. Thus, enhancing the overall performance of software development processes can lead to considerable economic gain. The demand for automated software quality assurance solutions is massive. In particular, AI, ML, and predictive analytics are playing an increasingly important role in quality assurance and automation of software development activities[3]. This is especially true for applications developed in highly iterative development processes with strong feedback to and from operations and users. By monitoring and recording end-user interaction with software systems, frequently performed application scenarios can be identified and captured in terms of data. In addition, end-to-end automation and the extensive use of repositories and databases in the development process provide extensive opportunities to access development data and metadata. All of this provides a rich basis for evaluation of software repositories using AI and ML algorithms, as well as automation based on them. A particular high potential is given in the context of DevOps processes.

However, there is currently a lack of market-ready solutions that realize such automation of quality assurance, in particular for non-functional requirements. Complexity and the implementation of critical use cases makes security testing a top priority for enterprises. Moreover, software vendors have begun to favor a user-centric approach to quality across the software development life cycle. This includes solving and preventing potential performance problems right at the beginning of the product lifecycle. Consequently, the analysis of non-functional characteristics such as scalability and stability of the application under various operating conditions is of increasing importance.

SmartDelta addresses this need for AI-based quality assurance solutions and automation specifically with respect to non-functional requirements. The innovative tools developed within the project can flexibly bring customized solutions to the market and sustainably influence software industries.

SmartDelta's topic is of strategic importance for both small and large software enterprises. SmartDelta's consortium is competent to fulfill the aim of optimizing the system development process characterized by continuous changes. The expected results will be an increase in the generated value out of digitalization including productivity, increased revenues, and market shares of involved companies through quality enhancements and the creation of jobs. SmartDelta also provides a uniform platform for tool providers from diverse backgrounds, to get feedback and to share experiences. For example, Belgium looks forward to having an early access to tools in the domain of Industry 4.0, while Canada looks forward to improving quality of the control software in mobility systems. Similarly, German industrial use cases are from the domains of enterprise and automotive software development, both representing strategic important application areas for Germany. Through SmartDelta's solutions, all participating countries expect to strengthen their delivery of high-quality software systems.

V. Related Work and Projects

SmartDelta develops automated solutions for quality assessment of product deltas in continuous engineering environments by providing smart analytics from development artifacts and system execution. Besides software quality engineering in general, SmartDelta is related to software product line engineering, continuous software engineering and machine learning in software quality engineering in particular.

Software product line engineering (SPLE) refers to software engineering methods, tools and techniques for creating multiple variants of the same software product [4]. The SPLE process consists of two main sub-processes known as domain engineering and application egineering. In the domain engineering phase, the domain of the product is analyzed, and domain knowledge is created. In the application engineering, the focus is on deriving product variants from the knowledge

[2] According to a report by Statista Research Department: https://www.statista.com/aboutus/our-research-commitment

[3] Worldwide Automation Testing Market 2018-2023, https://tinyurl.com/yrevh7h8

TABLE II
RELATED EUROPEAN PROJECTS.

Project Name	Program	Time Period	Technical Focus
AIDOaRt	ECSEL	2021-2024	Model-Driven Engineering (MDE) principles and techniques to provide a framework offering proper AI-enhanced methods and related tooling for building trustable CPSs
VALU3S	ECSEL	2020-2023	Verification and validation of automated systems, safety, and security
IVVES	ITEA3	2019-2021	Industrial-grade verification and validation of evolving systems
XIVT	ITEA3	2018-2021	Method and toolchain for testing highly configurable, variant-rich embedded systems
TESTOMAT	ITEA3	2017-2020	Test automation in agile development, mutation test, test selection and prioritization for single-system development
MegaM@Rt2	ECSEL	2017-2020	Scalable model-based framework for continuous development and runtime validation of complex systems
ENABLE_SE	ECSEL	2016-2019	Automation of verification and validation methods for automated cyber-physical systems
DiSIEM	H2020	2016-2019	Improvement to SIEM systems based on diversity related technology
REVaMP2	ITEA3	2016-2019	Round-trip engineering and variability management platform and process
STAMP	H2020	2016-2019	Pushing automation in DevOps through innovative methods of test amplification
MEASURE	ITEA3	2015-2019	Implementation of a comprehensive set of tools for automated and continuous measurement of software engineering activities
HyVar	H2020	2015-2019	Scalable hybrid variability for distributed evolving software systems
ATAC	ITEA2	2011-2014	Advanced test automation for complex and highly-configurable software-intensive systems
MBAT	ARTEMIS	2011-2014	Combined model-based analysis and testing of embedded systems
SecureChange	EU FP7	2009-2012	Methods and tools for ensuring compliance to evolving security requirements for long-running software systems

base created in the domain engineering phase. Approaches supporting the application engineering aid different activities in the process, such as product configuration [5], test re-use [6], and feature re-use [7]. However, product deltas (i.e., micro-variants) in continuous software engineering, as considered in SmartDelta, so far have not been considered.

Continuous software engineering, which mainly comprises continuous integration (CI), continuous deployment (CD) and continuous delivery (CDe), aims to accelerate the development and deployment of software [8]. While CI addresses the integration of ongoing development and internal quality assurance, CD and CDe are about the ability to quickly pass results to customers in order to enable rapid customer feedback [9]. In CI, the application is built and tested with every change to the codebase to achieve shorter and more frequent release cycles. DevOps is a culture shift towards lean principles, for the purpose of continuously delivering software [10]. It integrates development and operations, aiming at shortening the lead time between a change request and deployment. Continuous software engineering and the DevOps paradigm provides the context for SmartDelta.

Machine learning plays an important role in modern software quality engineering. For instance, different machine learning models have been applied to predict defects [11]. Machine learning is also the basis for recommender systems. Recommender systems in software development so far have focused mostly on the implementation and maintenance phases, but not so much on the software testing phase [12].

SmartDelta is one of the first projects to apply recommender systems to this phase. Natural language processing [13] and visualization [14] have not sufficiently been investigated in the context of software product deltas in continuous software development. These techniques play an important role in SmartDelta.

Table II shows European projects related to SmartDelta. For each project, the funding program, the time period, and the technical focus that links it to SmartDelta is provided.

VI. Summary and Conclusions

SmartDelta is a 3-year ITEA3 international project which focuses on the challenges of quality assurance and optimization in incremental industrial software systems development. To solve the challenges, SmartDelta builds automated solutions for quality assessment of product deltas in a continuous engineering environment by providing smart analytics from development artifacts and system execution, offering insights into quality improvements or degradation of different product versions, and providing recommendations for next builds. Towards this goal, the project will optimize the development process of systems that are built in an incremental manner, e.g., in a continuous engineering environment, as part of a product line, or based on previous legacy systems. Therefore, the outcomes of the project can have strategic significance for industries producing software-intensive products, and to improve their market competitiveness. To facilitate industrial adoption of the project results, the SmartDelta consortium intends to produce training materials, and organize international workshops and tutorials, which will be announced on the project's web page.

Acknowledgment

This work has been supported by and done in the scope of the ITEA3 SmartDelta project which has been funded by the national funding authorities of the participating countries: https://itea4.org/project/smartdelta.html.

References

[1] R. Potvin and J. Levenberg, "Why google stores billions of lines of code in a single repository," *Commun. ACM*, vol. 59, no. 7, p. 78–87, jun 2016. [Online]. Available: https://doi.org/10.1145/2854146

[2] Darren Allan, "Windows 10 update is reportedly freezing or crashing some PCs (and even causing boot loops)," https://www.techradar.com/news/windows-10-update-is-reportedly-freezing-or-crashing-some-pcs-and-even-causing-boot-loops, Aug 23, 2020.

[3] M. Saadatmand, A. Cicchetti, and M. Sjödin, "Toward model-based trade-off analysis of non-functional requirements," in *2012 38th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2012, pp. 142–149.

[4] K. Pohl, G. Böckle, and F. van der Linden, *Software product line engineering: foundations, principles, and techniques*. Springer, 2005, vol. 1.

[5] Y. Li, T. Yue, S. Ali, and L. Zhang, "Enabling automated requirements reuse and configuration," *Software & Systems Modeling*, vol. 18, no. 3, pp. 2177–2211, 2019.

[6] R. Ramler and W. Putschögl, "Reusing automated regression tests for multiple variants of a software product line," in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops*. IEEE, 2013, pp. 122–123.

[7] M. Abbas, M. Saadatmand, E. Enoiu, D. Sundamark, and C. Lindskog, "Automated reuse recommendation of product line assets based on natural language requirements," in *International Conference on Software and Software Reuse*. Springer, 2020, pp. 173–189.

[8] J. Humble and D. Farley, *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2010.

[9] B. Fitzgerald and K.-J. Stol, "Continuous software engineering: A roadmap and agenda," *Journal of Systems and Software*, vol. 123, pp. 176–189, 2017.

[10] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *IEEE Software*, vol. 33, no. 3, pp. 94–100, 2016.

[11] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A systematic literature review on fault prediction performance in software engineering," *IEEE Transactions on Software Engineering*, vol. 38, no. 6, pp. 1276–1304, 2011.

[12] M. Gasparic and A. Janes, "What recommendation systems for software engineering recommend: A systematic literature review," *Journal of Systems and Software*, vol. 113, pp. 101–113, 2016.

[13] V. Garousi, S. Bauer, and M. Felderer, "NLP-assisted software testing: A systematic mapping of the literature," *Information and Software Technology*, vol. 126, p. 106321, 2020.

[14] S. Diehl, *Software visualization: visualizing the structure, behaviour, and evolution of software*. Springer Science & Business Media, 2007.