

Interpretability/ Explainability Applied to Machine Learning Software Defect Prediction

An Industrial Perspective

Szymon Stradowski¹, Wrocław University of Science and Technology and Nokia Corporation

Lech Madeyski², Wrocław University of Science and Technology

// We explore the interpretability and explainability of machine learning software defect prediction models in the context of commercial system-level



©SHUTTERSTOCK.COM/TITIMA ONGKANTONG

testing in Nokia 5G.

We evaluate how they can gain stakeholders' acceptance, analyze expectations, offer management strategies, and identify business impacts. //

MACHINE LEARNING (ML) implementations have increased in recent years, indicating a rapidly expanding business potential.^{1,2} However, there are remaining obstacles that researchers and practitioners need to overcome to ease the further propagation of this technology in commercial contexts.^{3,4} One such obstacle is providing a level of understanding of the algorithm's decisions so that practitioners and other stakeholders can accept the predictions. The prevailing concepts attempting to solve the issue are explainable AI (XAI) and interpretable ML (IML). Notably, many authors do not differentiate between the two terms, but Rudin⁵ draws a line between interpretable and explainable AI/ML: IML focuses on designing inherently interpretable models, and XAI focuses on providing post hoc explanations for existing black box models that are incomprehensible to humans or are proprietary.

XAI is based on the notion of output confidence, helping subject matter experts understand why and how the algorithms came to their conclusions.⁶ It allows practitioners and researchers to validate the models not only deterministically but also subjectively. Treating the learners as nonintuitive black box oracles, detached from any human

influence, understanding, and trust, creates consequential implementation barriers. Therefore, enabling the understandability of the results is vital for the industry adaptation of many AI/ML solutions. In contrast, only limited evidence is available from in vivo studies within software engineering.²

Despite XAI gaining growing interest among researchers and being recognized as one of the main enablers for industry adoption,^{7,8} only a handful of interpretability studies are coming from industry in the field of software engineering. Moreover, the role of the audience is not sufficiently reflected in existing explainability approaches.⁶ We aim to fill this gap not only in terms of presenting industry insight but also in an extensive discussion on the real usage of the XAI mechanism to manage stakeholders' expectations during the new technology adoption process.⁹ Thus, we take an original, not-yet-explored stance on interpretability and expand the field's understanding from a real-world perspective. Furthermore, we aim to show that XAI results can be actionable in vivo.¹⁰ For our project, we used a well-established framework with restricted company-specific features and results. However, we offer real-world expectations, focus group observations, stakeholder management strategies, practitioners' expectations, and examples of achieved business impacts to support future researchers and practitioners in similar efforts.

Project Context

We aim to validate interpretability and explainability mechanisms in the context of the Nokia system-level test process during the adoption of a new ML software defect prediction

(SDP) approach. Specifically, we want to improve the continuous development and continuous integration quality assurance process of the 5G gNB element by adding an ML SDP solution to increase product quality and decrease costs. We employ a lightweight alternative to SDP that is relatively easy to implement in our environment to make initial inroads. Consequently, the key characteristic of our solution is that instead of predicting software defects directly, we predict test failures that are induced by those defects.

Our dataset is a collection of historical test process metrics gathered automatically from the company's online test repository database. It contains almost 800,000 individual test run results for more than 100,000 test instances executed over 500 unique software builds. Consequently, we trained ML models using historical test failures that have been confirmed as software defects. The Nokia 5G test process tracks such cases to specific code modules where the correction occurred. Second, we added a simple XAI mechanism to understand the importance of particular features and enhance the solution's capabilities further. The most impacting features are related to the date of execution, test instance, and responsible organization. We adopted a centralized approach where specific ML/AI specialists use our framework to run the predictions, create post hoc explanations, and later provide the results to software engineers for analysis. Moreover, simulations are run periodically based on new test repository data and are subsequently used to verify if the predicted software modules introduced new defects.

Need for Interpretability/Explainability

A survey by Carvalho et al.¹¹ highlights several important objectives for introducing XAI:

- The *interpretability requirement* is the primary rationale behind adding XAI to our solution. Although we are not relying only on a single metric to measure real-world performance, it is also not enough to act only based on predictions to make business-impacting decisions. Also, Nokia practitioners expect to understand the models sufficiently before acting upon them.
- The *high-stakes decisions impact* is an argument that applies to our problem only partially as our addition is not substituting the existing process, only enhancing it. As a result, it does not pose a high business risk; however, transparency and accountability for the ML SDP decisions are necessary when each defect escape is expensive.
- The *interpretability problem* is essential in our context as predictive modeling itself only partly addresses our aspiration to improve the quality assurance process. Models show where the defects are located, easing their finding and fixing; however, understanding the prediction rationale opens opportunities for longer-term improvement actions. Knowledge discovery can help solve hidden issues in software architecture and intrinsic vulnerabilities in particular software modules or eliminate deficiencies in testing organizations.
- *Regulation and societal concerns* are not yet part of the

reasoning behind our XAI implementation; however, regulatory needs are expected to increase in the future.

Several benefits can be gained from building in interpretability potential to the ML SDP solutions, such as transparency and trust, accountability and compliance, debugging and improvement, domain expertise, and knowledge discovery. Nevertheless, high interpretability (high model transparency) typically comes at the cost of lower performance.¹⁰ If a company wants to achieve the highest performance but still wants to explain the ML model's behavior in human terms, model explainability may be the option to choose. Also, not all of the available approaches, premodel, in-model, and postmodel,¹¹ can satisfy all the requirements with the desired level of performance. In our case, postmodel interpretability, referring to providing understanding after the models are already built, was selected as the easiest to develop and adequately satisfied our predefined expectations.

Specifically, many practitioners see it necessary to understand the decisions made by any AI/ML solutions to consider using the predictions in their work¹²; therefore, we aimed to:

- understand the top-ranked features of built models
- enable new knowledge discovery and domain expertise
- support stakeholder management
- increase confidence in the predictions.

However, we also need to consider that the impact of particular features can change depending on the investigated resampling techniques and

learners, and such changes can introduce a negative impact on the interpretation.¹² Thus, our solution also needs to be robust, model agnostic, and adaptable to the methodology used for building the predictions, as well as its evolution over time.

Therefore, we decided to use an external, post hoc model-agnostic explainable AI approach to build initial inroads as it is currently one of the most preferred methods for understanding the essential characteristics that constitute built models, with several tools available for the most popular languages. Also, this technique explains the classifier for a specific single instance of an already created model and is therefore suitable for local explanations and sufficiently satisfies our expectations.

Project Results

In our pilot study, we used the R framework to analyze our industry dataset of approximately 800,000 test records with the following five learners: Naïve Bayes, Classification Tree, Random Forest, Light Gradient-Boosting Machine, and Cat Boost Gradient Boosting.¹³ The top learners (Cat Boost and Random Forest from the ranger R package) reached the highest Matthews correlation coefficient (MCC), providing good predictive performance.¹³ We ran the global Friedman test with follow-up pairwise Nemenyi post hoc tests to verify whether the results from different models were statistically different and nonparametric effect size measures to assess the sizes of the differences.¹³

Next, we used the DALEX tool, which allows permutation-based variable-importance evaluation to understand the impact of particular features on the predictions.¹³ The tool offers model-agnostic

explanations, which create numerical and visual summaries of prediction behavior. Obtained outcomes showed that the test date, test run/instance, and responsible organization had the most meaningful impacts on the effectiveness of the prediction measured by the MCC. Understanding the relative importance of the features led to meaningful new domain knowledge discovery and the building of stakeholder management strategies discussed in this article. Detailed research results, reproduction package, discussion, and threats to validity have been provided in Madeyski and Stradowski.¹³

In our case, the main objective for deploying the XAI solution was the understanding requirement as we wanted to use the results and explanations as an additional output for our users' benefit. Notably, one of the remaining reasons for the interpretability problem to remain unsolved is that it is subjective and, therefore, challenging to measure, compare, and benchmark.¹¹ Consequently, the usefulness of understanding predictions is domain- and context-specific, making it necessary to consider the benefit of the use cases and the added value of each specific discovery. Thus, we assessed the solution's effectiveness based on the actionable business impact obtained by discovered debugging and improvement opportunities, domain expertise, and knowledge.

Namely, in our case, interpretability information led to uncovering previously unknown relationships between defect-finding test instances and software module correction patterns. It triggered a dedicated subproject that is currently underway and has a significant estimated impact on the existing quality assurance process. Also, a similar analysis

was initiated for the second discovery of a larger-than-expected number of software modules being executed for specific hardware configurations, which can be used to steer and optimize test resources. Last, the high importance of the date of the test execution allowed steering certain test cases to be better optimized (riskier test instances to be planned first).

In summary, in this industry project, we analyzed the business context, defined expectations, and built a suitable solution to satisfy the company's goals. We do not provide specific interpretability/explainability results but offer an analysis of the findings and their applicability potential. Therefore, the contribution brought by our article lies mainly in the applicability highlighted in the discussed commercial context, which is a rarely explored aspect of industry adoption research. Last, the evaluation and discussion of the obtained results lead to a recommendation for the company on the commercialization of the solution.

General Discussion

The following discussion summarizes our findings throughout the XAI implementation project in Nokia,¹³ which was documented and reviewed together with six participating AI/ML experts from the company during dedicated focus groups, an open-ended interview format for a group discussion, which is cost-effective and quick for obtaining qualitative insights and input from software engineering practitioners. We held three sessions with six participants located in Poland (four software engineers, each with more than five years of experience, and two test managers, each with more than four years of experience, supported by a facilitator to guide the discussion).

Thus, our observations not only are a source of practitioners' expectations and guidelines for similar implementation efforts but also serve as an insight for new XAI solutions that can embed our findings into the objectives from the get-go in designing new explainable architectures.

First, a rarely discussed issue is the time and effort the practitioners need to learn about and how to use the implemented ML SDP XAI solution. Two immediate approaches can be proposed, centralized and decentralized, having a pivotal impact on the project:

1. The *centralized* approach requires a dedicated team of AI/ML integrators that maintain the solution, framework, and servers. They relearn the models in specific periods of time (e.g., once per feature build), implement improvements, and provide predictions and interpretations to developers. Advantages and disadvantages are as follows:
 - could become a bottleneck
 - limited solution know-how
 - limited improvement potential
 - easier to deploy, manage, and monitor
 - may decrease the buy-in of practitioners
 - fosters one-size-fits-all products and services.
2. The *decentralized* approach is where the entire working AI/ML framework is available to the engineers who are expected to use and build the predictions and interpretations themselves. Advantages and disadvantages are as follows:
 - knowledge distributed around the organization
 - increased potential for improvement

- longer adaptation time and need of engineers' time to learn and use, and slower to adopt best practices
- time to generate instance explanations possibly becoming important, and the need for more computation power
- efforts or resources being duplicated but empowering experimentation and innovation.

Consequently, another aspect that should have a considerable impact on the expectations and process design is the scale of deployment. In the case of Nokia, as a grand software project with millions of lines of code, a decentralized approach would mean providing the tooling and training to thousands of software engineers around the globe, requiring deployment automation, a dedicated maintenance team, etc. The centralized approach is less costly as the project team would provide ready-made models and interpretations that engineers could use. On the other hand, the aggregated knowledge gained and improvement potential would surge. Also, the centralized and decentralized division is connected to the global and local explanation approach to XAI, where the global approach enables understanding of how the algorithms work versus local explanations that regard particular predictions. Specifically, local and global explanations can support the decentralized process. On the other hand, the centralized approach is more fitting for global explanations. The scale should also consider a trade-off between model interpretability and performance,¹¹ indicating that the more complex and accurate the model, the harder it is to interpret. Regardless, neither the highest

possible predictive performance nor the highest degree of interpretability are critical in our context. We decided on the centralized approach and reevaluated the transition to a decentralized setup after full commercial implementation.¹³

Importantly, explanation needs vary depending on the type of user and the problem being solved by ML methods. However, we expand this issue to the stakeholder management process¹⁴ and argue that each group needs specific information coming from the predictions, which should be used to satisfy expectations accordingly. For example, developers can regard false positives as even worse than false negatives, requiring ample time and effort to analyze sufficiently and causing unnecessary frustration. If the prediction fails to convince the developer of its rationale, more issues can be assumed as false positives and correct predictions can be ignored. Even more so, communication between developers and testers is one of the leading causes of conflict, and quality assurance experts can have difficulties convincing developers about the existence of real bugs or misbehavior of the system. SDP results have a minimal capacity to convince with limited persuasion power and can be ignored much more easily than test engineers. Accordingly, interpretability alone is insufficient, and it needs to come together with the capacity to defend the actions, provide relevant responses to questions, and be audited. Therefore, each prediction instance must be documented, i.e., labeled and commented on after analysis. The process requires time and effort from the developers, but it is invaluable for further performance improvements. Analyzed interpretations of individual predictions (true or false) can also complement the root cause analysis and escape

defect analysis¹⁴ efforts within the quality assurance process. Models expand the data collection process, and understanding the prediction decisions facilitates longer-term improvement actions by knowledge discovery, leading to solving hidden issues in software architecture or eliminating deficiencies of test organizations. Last, a high threshold for adoption still exists in ML SDP and other fault localization techniques.¹ Despite practitioners' enthusiasm in this field, many expectations are hard to satisfy in vivo. New methods to obtain and apply the understandability of predictions are needed to overcome the remaining industry adaptation barriers.

Stakeholder Management

Stakeholder management enables establishing and maintaining effective working relationships with all involved parties.¹⁴ Namely, the process includes continuous identification, analysis, prioritization, planning, communication, and monitoring of needs and relationships. Specifically, the first step of identification of all related parties (with a thorough examination of expectations, attitudes, roles, and levels of influence) needs to be performed. Based on the outcomes, determining communication requirements to be satisfied by creating and executing engagement strategies should be applied to optimize the chances of final success. Stakeholder management is also critical to the process of embedding the new solution into a broader context of commercial software development machinery. Furthermore, established Six Sigma¹⁴ tools like Voice of the Customer and Voice of Business can be used to understand the expectations in a broader context.

Most available research addresses the needs of developers and technical

experts and not those of other types of stakeholders. On the other hand, there is evidence that successful software engineering projects are related to the involvement of a large project team of different stakeholders in the decision-making process within the various stages of the project.¹⁵ Thus, there is a need to gain trust and commitment from a wide range of audiences, and enabling interpretability/explainability for AI/ML can be seen as a powerful tool to manage expectations and engagement.

Practitioners' Expectations

New technology readiness can be viewed as a summarized impact of four personality dimensions: optimism, innovativeness, discomfort, and insecurity.⁹ Thus, to increase readiness, we need to expand the understanding of universal expectations of the interpretability and explainability of ML/AI prediction capabilities. Hence, throughout our project, we meticulously gathered the expectations and feedback of the participating software engineers from Nokia during our focus group sessions. Consequently, we have divided the practitioners into three attitude categories: skeptics, believers, and agnostics, all needing to be addressed during the SDP introduction.

The engagement and communication strategies should reflect the person's expectation of the solution's benefits, as well as the general approach to the AI/ML field. The proportions among the three mentioned groups can be context dependent, and a dynamic startup environment may have more believers than a more established company. Likewise, the ratios change over time, but the transitions can be expedited with the project team's active support of

management and effective communication. Thus, all groups and individuals that are or will be impacted by the implemented solution need to be addressed in the designed stakeholder management process.^{8,15}

We identified five groups of stakeholders for ML SDP XAI in Nokia and provided an excerpt from our management strategy matrix in Table 1. Each identified group has its own expectations for the solution's usefulness, and interpretability results can help (or hinder) the perceptions. Below, we suggest a few important

premises observed during our feedback sessions that summarize the main characteristics of each group, used to guide stakeholder management efforts throughout the entire ML SDP adoption:

- *Engineering*: The main interest of software engineers in using AI/ML solutions is to make the work easier and/or faster and for their results to be perceived as better overall. Second, knowledge discovery and domain expertise increase are sought to

satisfy curiosity and enable further improvements to the work deliverables. Hence, practitioners want to understand the predictions but, most importantly, see the direct benefits to their daily work, emphasizing the need for interpretability in process innovation. Also, this is aligned with the premise that engineers perceive that understanding the characteristics associated with software defects in the past is useful.¹²

- *Management*: One important aspect related to management's

Table 1. An exemplary stakeholder management strategy matrix for an ML SDP adoption project using XAI.

Profile	Description	Strategy	Actions	XAI Impact
Group 1	Sponsors and decision makers, with high influence on the project	Manage closely, regular reporting	Gather expectations, measure engagement, report regularly, communicate the timelines and achievements	Provide very precise information on gained knowledge, high-level evidence of XAI business value, impact on business metrics, potential regulatory and societal requirements
Group 2a	Technical staff (believers and agnostics), using the solution	Engage, frequent communication	Gather feedback and improvement proposals, measure engagement, provide frequent updates, enable room for innovation, offer training, consult	Offer as much output information as possible on all XAI aspects and results
Group 2b	Technical staff (skeptics), using the solution	Advocate and convince, moderate communication	Gather feedback, measure engagement, provide training, provide evidence of good predictive performance, use precise informative messaging	Offer feature importance analysis, limiting false positives, domain expertise increase
Group 3a	Management staff (believers)	Satisfy expectations, less frequent communication	Gather input, consult on business cases, measure engagement, focus on business impact and effectiveness increase	Provide evidence of predictive performance and efficiency improvements due to XAI, showcase future possibilities, develop quality improvement plans
Group 3b	Management staff (skeptics and agnostics)	Satisfy expectations, less frequent communication	Gather feedback, measure engagement, focus on business impact and effectiveness increase	Provide evidence of added business value, demonstrate initial inroads, use XAI for knowledge discovery, show positive business impact examples and cost-benefit considerations
Group 4	Underlying process owners	Consult, occasional updates	Gather process input, provide project updates and results for inroads	Discover relevant/interesting information, provide evidence of process efficiency improvements
Group 5	Technical and management staff outside of the project	Monitor and keep informed, occasional updates	Open chat for questions, provide infrequent and mass updates on purpose and concept (newsletter, all hands, etc.)	Use XAI as a source of interesting facts and talking points to raise interest in the project

stake in AI/ML projects relies upon cost considerations and expected return on investments. Building and defending a business case may be advisable, including a dedicated cost-benefit evaluation.⁸ Additionally, quality management is an important subset of this group, mainly interested in process improvement discoveries and defect containment improvements supported by interpretability/explainability.¹⁰ The tangible effects and impacts on efficiency, costs, and product quality are essential. XAI only partially satisfies those expectations; however, it helps build trust and spark innovation and may ease the process of gaining the technical staff's support (as engineers can be the best ML SDP advocates to management if the solution provides sufficient assistance).

- **Legal:** From a requirements analysis perspective, legal and regulatory obligations can be straightforward because the expectations should be clearly defined and formalized. On the other hand, a disconnect may arise from what is possible or helpful from the engineering perspective. Thus, purely legal obligations are an unavoidable cost, where the interpretability brings no direct benefits to practitioners and just satisfies formal requirements.

Satisfying practitioners' expectations is vital to increase the success probability of new technology adoption.⁹ Even the best-engineered process can only succeed if it can demonstrate initial progress and procure buy-in from critical stakeholders,¹⁵ especially in the field of

ABOUT THE AUTHORS



SZYMON STRADOWSKI is an Implementation Ph.D. student in the field of software engineering at Wroclaw University of Science and Technology, 50370 Wroclaw, Poland, and also project office manager at Mobile Networks Radio Frequency, Nokia Corporation, 54155 Wroclaw, Poland. His research interests include applying ML to SDP in vivo. Stradowski received a master's in teleinformatics, a bachelor's in finance and accounting, and an MBA from Franklin University. He is a Member of IEEE. Contact him at szymon.stradowski@pwr.edu.pl.



LECH MADEYSKI is an assistant professor in the Software Engineering Department, Institute of Informatics, Wroclaw University of Technology, 50370 Wroclaw, Poland. His research interests include empirical software engineering, data science in software engineering, reproducible research, and robust statistical methods. He is a Senior Member of IEEE. Contact him at lech.madeyski@pwr.edu.pl.

software engineering, where there are many unknowns hindering understanding and blurring potential outcomes. Hence, not having interpretability/explainability information available may cause the transition of ML SDP to practice impossible in many contexts. We recommend considering XAI from the outset of any ML adoption project.⁸

Threats to Validity

For external validity, as we use a proprietary industrial dataset and process, the generalizability of the interpretability results themselves is restricted.¹³ However, similar ML SDP methods and tools we used have been validated in many contexts, academic and real world, in the past. Furthermore, other large-scale software projects using similar development and testing processes can benefit from our proposals.⁴ Likewise, all gathered feedback,

stakeholder management strategies, and described lessons learned can be helpful to other ML SDP implementation projects.

For construct validity, our dataset contains a mix of automatically and manually generated data; hence, it varies in quality, as is typical for in vivo studies. However, the research rigor and the robust MCC performance metric we applied help ensure that the results are trustworthy and the proposed solution and conclusions can be valuable for industry and research. Also, the underlying study provides a complete dataset and software package to allow reproduction.¹³

In this industry article, we report the outcome of using an explainable AI method that offers model-agnostic explanations for predictions to be understandable to

practitioners. First, our pilot study results proved to be actionable and confirmed that the implemented mechanism can be used to interpret ML SDP models meaningfully for Nokia 5G system-level testing in vivo.¹³ Second, we gathered comprehensive observations during and after the adoption process and discussed them in focus group meetings with the company's practitioners. Finally, we devised stakeholder management strategies to increase the chances of our ML SDP implementation's success, according to the project and quality management perspectives. As the next steps, we plan to observe the long-term sustainability of the solution and the evolution of the subjective trust of particular stakeholder groups in the predictions to improve the proposed strategies.

Finally, creating dedicated management strategies for different categories of stakeholders (including approaches and roles) is essential to increasing effectiveness and engagement for ML SDP adoption projects. Hence, our considerations can be beneficial in future improvements to the field of ML in software engineering, including aspects like finding a balance between interpretability and performance, providing tailored interpretations, discovering new domain knowledge, and supporting future solutions like informed ML and human-in-the-loop approaches, as well as new interpretable AI/ML architectures. 🌐

Acknowledgments

This research was done in partnership with Nokia (with Szymon Stradowski as an employee) and was financed by the Polish Ministry of Education and Science "Implementation Doctorate" program (ID: DWD/5/0178/2021).

References

1. V. Garousi and M. Felderer, "Worlds apart: Industrial and academic focus areas in software testing," *IEEE Softw.*, vol. 34, no. 5, pp. 38–45, Sep. 2017, doi: [10.1109/MS.2017.3641116](https://doi.org/10.1109/MS.2017.3641116).
2. S. Stradowski and L. Madeyski, "Industrial applications of software defect prediction using machine learning: A business-driven systematic literature review," *Inf. Softw. Technol.*, vol. 159, Jul. 2023, Art. no. 107192, doi: [10.1016/j.infsof.2023.107192](https://doi.org/10.1016/j.infsof.2023.107192).
3. M. Lanza, A. Mocci, and L. Ponzanelli, "The tragedy of defect prediction, prince of empirical software engineering research," *IEEE Softw.*, vol. 33, no. 6, pp. 102–105, Nov./Dec. 2016, doi: [10.1109/MS.2016.156](https://doi.org/10.1109/MS.2016.156).
4. L. Briand, D. Bianculli, S. Nejati, F. Pastore, and M. Sabetzadeh, "The case for context-driven software engineering research: Generalizability is overrated," *IEEE Softw.*, vol. 34, no. 5, pp. 72–75, Sep. 2017, doi: [10.1109/MS.2017.3571562](https://doi.org/10.1109/MS.2017.3571562).
5. C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019, doi: [10.1038/s42256-019-0048-x](https://doi.org/10.1038/s42256-019-0048-x).
6. A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52,138–52,160, 2018, doi: [10.1109/ACCESS.2018.2870052](https://doi.org/10.1109/ACCESS.2018.2870052).
7. H. Dam, T. Tran, and A. Ghose, "Explainable software analytics," in *Proc. 40th Int. Conf. Softw. Eng. (ICSE)*, Gothenburg, Sweden: IEEE/ACM, 2018, pp. 53–56, doi: [10.1145/3183399.3183424](https://doi.org/10.1145/3183399.3183424).
8. S. Stradowski and L. Madeyski, "Bridging the gap between academia and industry in machine learning software defect prediction: Thirteen considerations," in *Proc. 38th Int. Conf. Autom. Softw. Eng. (ASE)*, Kirchberg, Luxemburg: IEEE/ACM, 2023, pp. 1098–1110, doi: [10.1145/3183399.3183424](https://doi.org/10.1145/3183399.3183424).
9. P. Godoe and T. Johansen, "Understanding adoption of new technologies: Technology readiness and technology acceptance as an integrated concept," *J. Eur. Psychol. Students*, vol. 3, no. 1, pp. 38–52, 2012, doi: [10.5334/jeps.aq](https://doi.org/10.5334/jeps.aq).
10. C. K. Tantithamthavorn and J. Jiarapakdee, "Explainable AI for software engineering," in *Proc. 36th Int. Conf. Autom. Softw. Eng. (ASE)*, Virtual event: IEEE/ACM, 2021, pp. 1–7.
11. D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019, Art. no. 832, doi: [10.3390/electronics8080832](https://doi.org/10.3390/electronics8080832).
12. J. Jiarapakdee, C. K. Tantithamthavorn, H. K. Dam, and J. Grundy, "An empirical study of model-agnostic techniques for defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 48, no. 1, pp. 166–185, Jan. 2022, doi: [10.1109/TSE.2020.2982385](https://doi.org/10.1109/TSE.2020.2982385).
13. L. Madeyski and S. Stradowski, "Predicting test failures induced by software defects: A lightweight alternative to software defect prediction and its industrial application." (2024) (in reviews). [Online]. Available: <https://madeyski.e-informatyka.pl/download/MadeyskiStradowski24p.pdf>
14. T. Pyzdek and P. Keller, *The Six Sigma Handbook: A Complete Guide for Green Belts, Black Belts, and Managers at All Levels*, 3rd ed. New York, NY, USA: McGraw-Hill Companies, 2017.
15. J. McManus, "A stakeholder perspective within software engineering projects," in *Proc. IEEE Int. Eng. Manage. Conf.*, vol. 2, Piscataway, NJ, USA: IEEE, 2004, pp. 880–884, doi: [10.1109/IEMC.2004.1407508](https://doi.org/10.1109/IEMC.2004.1407508).