International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I8P105
Volume 73 Issue 8, 57-72, August 2025

Original Article

Adopting DevOps/XOps in German Medium-to-Large IT Enterprises: A Framework

Ihab J. S. Mansour[1], Mazidah Binti Mat Rejab[2], Hairulnizam Bin Mahdin[3], Emad Hamadaqa[4]

[1,2,3]*Faculty of Computer Science and Information Technology Universiti Tun Hussein Onn Malaysia.*
[4]*T-Systems Internationals GMBH, Germany.*

[1]*Corresponding Author : ihabmansour@gmail.com*

Received: 25 November 2024 Revised: 04 July 2025 Accepted: 22 July 2025 Published: 30 August 2025

Abstract - *This research explores the critical factors influencing the Adoption of DevOps and extended XOps practices in medium-to-large IT enterprises operating in Germany. The paper uses the theoretical framework of the Unified Theory of Acceptance and Use of Technology-2 to evaluate how organisational readiness, technical capability, customer and stakeholder involvement, and automated security testing (DevSecOps) elements affect the adoption. In this respect, a quantitative approach was adopted, where data were collected from a sample of German IT enterprises by means of structured questionnaires. The relationships among latent constructs were studied with SmartPLS 4, which was used for conducting data analysis and thus investigating the moderating effects of operational readiness and cross-functional training. Key findings highlight how organisational culture, team collaboration, technical infrastructure, and testing practices will facilitate the Adoption of DevOps/XOps. These insights provide recommendations for how IT enterprises can transform and leverage agile software development in their quest to achieve and sustain competitive advantage in an increasingly dynamic IT industry.*

Keywords - *DevOps, AIOps, DataOps, GitOps, MLOps, Agile methodology, IT enterprise, UTAUT2.*

1. Introduction

1.1. Introduction to Adopting DevOps/XOps in German Medium-to-Large IT Enterprises: A Framework

Companies must be competitive and efficient under the current conditions of fast-moving information technology. This can be observed by adopting agile software development methodologies and management strategies. The current work elaborates on the research methodology that explores adopting those practices in Germany's medium- to large-sized IT enterprises. The paper opens with an introduction in which the author focuses on the Unified Theory of Acceptance and Use of Technology as the primary model guiding this study. This model gives reasons why German IT companies are shifting towards agile methods and an overview of the types of questionnaires used to ensure the research was comprehensive and detailed. This paper is primarily concerned with critically examining the myriad factors that have a bearing on adopting agile software development and management practices within these enterprises. From organisational readiness and technical infrastructure to the issue of customers and stakeholders' involvement, readiness for change, and team dynamics, every minute detail is rightly analysed with scholarly research and real-world data to understand their effect on embracing agile methodologies. These have moderated the relationship with factors such as operational readiness, DevOps/XOPS Agile Software Engineering integration, and finally, that of DevSecOps-automated security testing. It is discovered that each of these elements significantly influences how the other factors eventually influence the final decision to adopt agile and automated software development tools. This work details the research methods view and breaks down the complex factors while adopting agile practices. With detailed analysis and empirical evidence, it is meant to clarify the intricacies of agile adoption in support of informed decision-making and strategic development for software management practices within German IT enterprises.

1.2. Problem Statement

Research studies that aim to study and enhance the rate of adoption of agile software development and management approaches are conspicuously lacking in German IT Medium-to-Large Enterprises (MLEs) (Mikhieieva & Pfannenstein [1]). Research needs to be carried out to determine the strengths and weaknesses of existing practices and inform improvements towards more interactive and networked practices. Despite the increased popularity of DevOps/XOps in the global IT industry, most German MLEs lag in their adoption of the same. This is attributed to some major determinants for successful implementation (ISG Provider Lens™, 2021). It is crucial to identify these limiting factors to facilitate the successful adoption and incorporation of DevOps/XOps (Khan et al. [2]). Despite growing Adoption of

DevOps/XOps practices worldwide, scepticism over the most important determinants to consider when adopting the practices continues within German MLEs (Contino [3]). Gradual and disorganised adoption points to deep-rooted issues that are not generally well-known thus far (Diebold et al. [4]). Although there is some existing research, an evident gap persists when it comes to studies concentrating on DevOps/XOps Adoption by German medium-and large IT companies, narrowing down knowledge about these critical determinants of adoption.

1.3. UTAUT2

It was not by coincidence that the Unified Theory of Acceptance and Use of Technology 2 (UTAUT2) was chosen as the foundational framework for this work project; rather, it was motivated by the theory's applicability and relevance to the goals of this investigation.

UTAUT2 was chosen for this research work based on a couple of crucial reasons [5] Gharaibeh:

- Holistic View: Because it considers four aspects of technology adoption-performance, expectancy, and social impact UTAUT2 offers a comprehensive perspective. It thus provided a nuanced understanding of the complex interplay between individual beliefs, attitudes, and external factors shaping technology adoption behavior.
- Empirical Validity: UTAUT2 has been highly tested and validated across different contexts and populations, proving its strength and generalizability. Its empirical basis provides credibility to the research findings and guarantees that the data generated from the investigation is founded on reliable theoretical ideas.
- Applicability to Agile Adoption: Though UTAUT2 was primarily designed for the context of general technology diffusion, the principles are very useful for agile software development methodologies and management practice adoption. This model emphasises user perceptions, social influences, and contextual factors, all of which sit very well with the inherent complexities of agile adoption processes within the organisational setting.
- Practical Utility: UTAUT2 provides several valuable insights for policy-makers, practitioners, and researchers concerned with technology adoption and implementation initiatives. The model, therefore, becomes the instrument to allow for the development of effective interventions and agile adoption strategies for IT MLEs by enabling the identification of the critical determinants of adoption behavior and their relative importance.

The choice to base this research project on UTAUT2 is informed by its comprehensiveness, empirical validity, applicability to agile adoption contexts, and practical utility. Building on the insights offered by UTAUT2, this research aims to tease out fine-grained insights into agile Adoption within German IT MLEs and further provide helpful recommendations for improved adoption outcomes.

Because of its applicability and effectiveness in achieving the objectives of the study, the UTAUT2 was chosen as the foundation for this research project.

2. Literature Review: Adoption of Agile Software Development Methodology and Management in German IT MLEs Companies Model based on the UTAUT2 Model for XOPS

2.1. Introduction

To grasp the history and utility of emerging operational practices in IT environments, contrasting their inherent strengths, weaknesses, and strategic utility is significant. Based on the latest research, Table 1 aggregates a comparative evaluation of the five most significant methodologies-DevOps, AIOps, DataOps, GitOps, and MLOps. The comparison outlines how each methodology enhances operations to become more effective, team-based, and system fault-tolerant, along with common barriers to successfully adopting them.

Table 1. Comparison of methodologies

Methodology	Authors	Challenges	Important Insight
DevOps	João Faustino et al. [6], Fernando Almeida, Jorge Simões, and Sérgio Lopes [7]	- Resistance to change. -Difficulties integrating many tools. - The need for a cultural transformation.	Breaks down boundaries between development and operations to improve efficiency and quality. Aligns well with Agile principles.
AIOps	Abhijit Sen [8], Yingnong Dang, Qingwei Lin, and Peng Huang [9]	- Complexity of AI/ML model development. - Legacy system integration challenges. - High demand for AI/ML skills.	Leverages AI for better operations, focusing on automation and performance reliability.
DataOps	Julian Ereth [10], Kiran Mainali et al. [11], Yuri	- Data privacy and security issues. - Legacy system	It focuses on optimising data analytics and workflows, which is vital for data-driven organisations.

		integration. - Maintaining consistency and accuracy.	
GitOps	Artem Lajko et al. [12], Pablo Gómez-Caldito Gómez [13], Matti	- Tooling and CI/CD integration complexity. - Need for GitOps-skilled personnel. - Alignment with existing DevOps workflows.	Manages infrastructure with Git, enhancing reliability and streamlining operations.
MLOps	Rakshith Subramanya et al. [14], Sasu Tommi Mikkonen et al. [15]	- Integration with traditional cycles. - Cultural adaptation needed. - Tool selection and support requirements.	- Integration with traditional cycles. - Cultural adaptation needed. - Tool selection and support requirements.

2.2. Independent Factor

2.2.1. Independent Factor 1: Organisational Readiness and Technical Capabilities

Kuiper [16] examines the development of transformational leadership in relation to organisational change in financial services companies using Agile and DevOps. According to this statement, the two most important elements in fostering effective changes are organisational preparedness and technology capacity.

Such initiatives demand sound leadership to guide, motivate, and empower employees through the change with a sense of ownership of work. This will create synergy and alignment toward common objectives, something very important in Agile and DevOps practices. Furthermore, emphasis on continuous improvement provides an adaptive response to an organisation's innovativeness and change in circumstances. Transparency in communications and decision-making nurtures confidence and leads to greater employee engagement, which also forms an integral part of the success of change initiatives. A clearly defined vision and strategy provide direction and ensure that the purpose of employees is well understood to guide them toward the desired outcomes of Agile and DevOps transformations.

Proper resourcing includes financial, human resources, and technological resources to support the effective implementation of Agile and DevOps practices. Finally, effective change management processes guarantee smooth transitions with minimal resistance to change, therefore allowing the agile and DevOps methodologies to be seamlessly absorbed within the organisation. These include Kuiper, Christopher J. [16]: Leadership is about creating an enabling culture for an agile workplace with organisational values aligned to agile principles: customer orientation, collaboration, and continuous improvement. It is impossible to imagine DevOps culture without employee empowerment because enabling the responsibility for decision-making drives collaboration and thus innovation. Teamwork and collaboration amplify this effect: open communication gives birth to collective skill and innovation within the framework of a culture. Continuous improvement is seen as one of the enablers of positive behaviors. Leaders are expected to establish goals, make resources available, and even permit experimentation. While transparency engenders trust and accountability required for agile adoption, it tends to challenge hierarchical structures.

A properly defined vision and strategy that are correctly communicated round up the other critical success factors. DevOps case studies show that leadership commitment guarantees successful adoption-resource allocation to support the DevOps initiatives with appropriate toolsets, training, and cross-functional teams. Finally, effective change management utilises Kotter's model of change to drive leadership to communicate a clear vision, establish urgency, and build supportive coalitions by embedding the approaches in corporate memory.

2.2.2. Independent Factor 2: Coding Version Control and Complexity of Software Development

The case study by Srivastav, Allam, and Mustyala [17] dwells on the way DevOps implementation enhances software automation and resolves complications in software development. DevOps influence spans a lot of other software engineering aspects, including the control of versions in coding and project complexity. DevOps will reshape technology stacks, software architectures, and development processes that drive smooth workflows and collaboration among various teams. Besides, DevOps strongly supports automated testing and quality assurance practices that make software products quite reliable and functional, whereas cultural advancement is fostered. Successful DevOps adoption requires capable cross-functional teams applying automation tools effectively. The project scope is also important: large projects require an effective automation framework and collaboration. DevOps fosters teams' continuous assessment and smoothing of processes, tools, and workflows to achieve quality and efficiency. DevOps practices allow organisations to address these challenges through modern software development, providing the right set of protocols for open, collaborative, product-centred team environments that endow the development team with the capability to deliver quality software in an efficient and adaptive way.

The technology stack is also a vital determinant in the adoption of DevOps; an organisation should know how its present stack fits the principles of DevOps and identify areas for upgrade that will facilitate automation and collaboration. Software architecture also influences development complexity significantly. Well-designed architecture, Conway's Law, and system design that is aligned with business goals can lower the complexity. Frameworks, such as decision trees, can be used to support the selection of a development process based on project complexity, including size and team distribution. Additionally, testing and quality assurance are further empowered through the integration of DevOps practices, allowing a holistic approach across the entire development lifecycle. Expertise within teams would be advantageous, as diverse skill sets have been known to aid in dealing with the complications of development. Continuous learning is an important factor in knowledge renewal. Agile projects do not find project scope management easy; early involvement of stakeholders, prioritisation of requirements, and agile techniques such as sprint planning help manage effectively. Continuous improvement practices through retrospectives and root cause analysis to foster knowledge sharing and skills development that arm the teams with better control over their tasks to handle intricate work effectively.

2.2.3. Factor Independent 3: Customer and Stakeholder Involvement

Lwakatare et al. [18] investigate how DevOps practices affect customer and stakeholder engagement in software development. Their study analysed the level of engagement in five companies. From the studies done, they found that seven key drivers determine the level of involvement for customers and stakeholders:

- Codebase size and complexity: The larger and more complicated the codebase is, the more customers and stakeholders must get involved with it to make things seamless.
- Integration with external systems: Where software needs to integrate with other systems outside the company, this needs to coordinate with the external stakeholders, affecting how engaged they will be.
- Technical debt: If there are existing problems or "technical debts" in the software, then the stakeholders may be more participatory to ensure that those problems are sorted out so that the software meets their expectations.
- Development tools and technologies: The degree to which it will be easy for the stakeholders to collaborate and keep themselves updated about the progress of the projects is related to the development tools and technologies being used.
- Team organisation: How the development teams are organised and the form of internal communication applied may affect how well stakeholders are informed.
- Development methodologies: Approaches to development, such as Agile or Waterfall, prescribe how the requirements are managed and communicated to the stakeholders during a project.
- Stakeholder requirements: Knowing what stakeholders need and want from the software is critical for keeping them interested and satisfied during development.

Consequently, considering DevOps, understanding these facilitation factors enables the active incorporation of customers and stakeholders in development and hence fosters better results for all concerned.

2.2.4. Independent Factor 4: Change Readiness and Team Collaboration

The most integral part of agile software development methodology and management adoption in German IT MLE companies is team collaboration and effective communication. The agile approach expects teams to be cross-functional, with a knowledge-sharing team environment; collaboration and communication are the essential elements for this. This is a very valid reason why proper training and education should be provided to help them equip themselves with suitable skills and knowledge for effective communication and collaboration. Nagarajan's thesis [19] discusses how large financial enterprises can successfully implement DevOps, focusing on Change Readiness and Team Collaboration. Grounded in a study that tries to map seven key factors influencing both:

Effective communication is greatly needed to bring organisational goals into focus, especially where transitions are concerned, as transparency creates cohesion and strategic focus on DevOps objectives. Change management extends beyond new processes by also addressing potential resistance and stakeholder buy-in to build resiliency and adaptability. These incentives and rewards further motivate the teams to undertake the journey of DevOps by creating excitement, recognition of achievement, and reinforcement of a continuous improvement culture. Teamwork and collaboration are foundational to DevOps. A collaborative culture involves the pulling together of diverse perspectives to foster trust, accountability, and shared ownership of goals. Flexibility does indeed facilitate teams' responses to market shifts and customer needs, thus making it vital in dynamically shaped environments. Well-defined roles and responsibilities also reduce confusion while enhancing accountability to such an extent that processes get streamlined, which prevents duplication of efforts. Efficient communication tools ensure real-time updates and smooth coordination across teams, irrespective of distances or time zones. These factors help build an organisation's preparedness toward DevOps by fostering teamwork and seamless transition toward this collaborative adaptive paradigm. As [20] Mahajan Agile Manufacturing concerns teamwork, cross-functional collaboration, and adaptability, which immediately appeals to the main ideas of Agile software development methodologies. While Agile Manufacturing takes diverse teams in different departments working as one to adapt to fast-changing market needs, Agile software development includes collaboration of developers, testers, and stakeholders to adapt to evolving project requirements. Agile Manufacturing, which has been made much easier with the advent of modular designs and real-time communication, is iterative and flexible, like how continuous integration and iteration in software projects are. Both cases justify the principle of maximising collaborative adaptability to high-quality, efficient results by allowing for a team environment in which members contribute diverse skills and adapt to dynamic conditions.

2.2.5. Independent Factor 5: Testing and Performance Evaluation

Quality and performance metrics significantly influence the adoption of agile software development methodology and management in German IT MLE companies. To ensure that the teams can measure and enhance software quality and performance, it becomes essential to implement cross-functional teams, suitable training and education, and access to the relevant tools and technologies. Cross-functional teams can monitor and enhance the software's performance and quality because they bring together individuals with diverse backgrounds and skill sets to work toward a similar goal. This would be further enhanced by the adoption of an appropriate technical environment.

The adoption of agile software development methodology and management is not complete without adopting quality and performance metrics in German IT MLE companies. Secondly, for the successful adoption of quality and performance metrics, there is a need to change organisational culture and structure to support collaboration and communication effectively. With that, better results can come in terms of agile methodology implementation and its impact on the software development process and its outcome. The paper by Bezemer et al. [21] discusses how DevOps approaches address performance concerns, focusing on Testing and Performance Evaluation. The paper identifies seven key factors related to Testing and Performance Evaluation as part of DevOps:

Well-defined standards on quality allow for clear routes on how to conduct the testing by helping teams focus their efforts on what is expected to be found as an outcome. The measurable metrics, such as times of response and error rates, are translatable, permitting their performance to be tracked to measure progress and make decisions to optimise based on data. Continuous testing and feedback help in early issue detection, putting the focus on performance at the center of its life cycle. Integrating continuous testing into the CI/CD pipelines lets them think about performance in each stage of development. Automation tools support frequent and consistent testing, hence allowing greater stability. Continuous monitoring provides real-time insight, helping teams proactively identify and resolve performance issues that reduce downtime and improve user experience. Training and development allow its members to learn from the industry the best practices and new technologies that improve testing capability and keep driving further improvement. Finally, a team-oriented environment enhances collaboration, shared responsibility, and problem-solving by enabling organisations to capitalise on collective expertise for the betterment of testing practices that result in performance outcomes.

2.2.6. Independent Factor 6: Orchestration and Governance

The elements that affect software development and management's adoption of an agile approach by German IT MLE companies are varied. Organisational culture and values, leadership and management support, the complexity of software development, customer and stakeholder involvement, team collaboration and communication, and planning and coordination are all factors that significantly influence the successful adoption of the agile methodology.

These success factors of Agile adoption are related to organisational cultural and structural changes, investment in training and education, cross-functional teams, technical infrastructure, and performance metrics. German IT MLE companies can leverage these enabling factors to create better processes and products in software development that will make them more competitive in the rapidly changing technology environment.

Msitshana's [22] The main goal of the thesis is to employ project management as a governance tool in the DevOps phenomenon, namely in Orchestration and Governance. He lists the following five orchestration and governance influencers: Principles of project management applied to DevOps will move coordination and align teams toward the same goals by providing structured guidance in managing a complex project. With appropriate resource allocation, resources are rightly planned and available just in time for efficiency maximisation and bottleneck reduction. Engaging stakeholders through the DevOps process will improve collaboration, thereby enhancing governance and increasing transparency to help stakeholders make better decisions. Agile methodology further complements DevOps through iterative development, flexibility, rapid response to change, continuous feedback, and incremental delivery. Such leverage of appropriate tools and technologies allows automation, real-time visibility, and proactive management to improve DevOps orchestration and governance. All these factors help an organisation strengthen its DevOps practices for well-coordinated activities, efficient use of resources, stakeholder satisfaction, adherence to Agile principles, and effective use of technology.

2.2.7. Independent Factor 7: Software Building Development Template

Understanding a structured template to be followed for software development will directly impact the needs required by software maintenance, as debugging errors and coding defects will be much easier when they appear. In software development, a template is generally based on a model or framework that jump-starts the development process to do most of the work in building tools and workflows. The code libraries, configuration files, and build scripts will be provided in a software development template, since a developer might focus on the creation of an application rather than putting together, assembling and configuring a development environment that could eventually result in several errors or inconsistencies, negatively affecting performance or reliability. The advantages of using software development templates include standardised development and time-saving during the actual development. With well-established code libraries, one can use automated build scripts to build and test applications speedily, hence improving build-and-deploy time. Nocera's [23] study throws light on how DevOps as a methodology speeds up the cycle of software creation and delivery: innovative ways for agile development. Regarding this, the approach involves the following points under the Software Building Development Template:

Modularity and reusability are paramount in development because a project is divided into reusable components, where a developer can use prebuilt modules for new functionality. High-quality, consistent code improves the reliability of the software, as maintenance is easier; further, uniform coding standards reduce errors, hence collaboration increases. Strong error handling makes the software more robust; it improves the user experience by anticipating and resolving issues that may pop up. Integrating automated testing provides accelerated testing processes, hence rapid feedback for the changes made on the code to catch issues early and fix them. Scalability and performance optimisation prepare the software for future growth by ensuring responsiveness under a high volume of work. In-depth documentation and knowledge sharing allow for easy onboarding and collaboration, thus supporting continued development. Cross-platform compatibility increases accessibility by making the software usable across a multitude of operating systems and environments for an expanded customer base.

2.2.8. Independent Variable 8: containerisation and Development Environment Management

In this variable, a direct relation can be easily defined, relating the level of needed maintenance to the level of containerisation adoption and the development of environmentally stored images. Imaging of the development environment and backup policies, in an IT enterprise, application development, containerisation, and other corresponding processes that ensure software applications are developed, deployed, and maintained with efficiency and reliability. Containerisation helps pack the varied types of software applications that have been developed with their required dependencies in lightweight, portable containers to ensure consistent deployment across various environments. This approach enables developers to create, test, and deploy applications more efficiently and reliably. It helps in the imagination of a development environment that ensures development environments remain consistent across different machines or cloud services, reducing the chances of errors and configuration issues.

Backup policies include procedures and tools used to ensure regular backups of data and applications take place to avoid data loss. This is a very important process in maintaining the integrity of data and making sure that application availability and usability are guaranteed in case of failure. Through regular backups, IT enterprises can instantly restore applications and data to the most recent state, thus avoiding possible data loss. In summary, these processes and tools play a crucial role in an IT enterprise to allow smooth development, enhance efficiency, ensure applications are reliable and secure, and guarantee their availability. Fokaefs et al. [24] present an investigation on the application of DevOps to containerised data-intensive applications. The two main focuses are Containerization and Development Environment Management. A summary of their results is as follows:

Containerisation is efficient because it packages applications and their dependencies, hence ensuring consistency in deployment across different environments. It provides an immense amount of portability and interoperability to have an application run easily on different platforms; thus, it makes the deployment and integration process easier. Versioning and optimisation in effectively creating and managing images make their distribution to teams easier. Integrating the container registries allows the centralised storage of images that can be distributed and easily shared among team members. Networking and performance optimisation inside containers increase application responsiveness and scalability for good communication between containers and other services. With containerisation, there will be scalable and load-managed environments. Dynamic scaling and load-balancing mechanisms are set up to sustain application performance during high demand. Containerised environment reproducibility and consistency encourage collaboration among the development team. Containerisation and management of the development environment together help an organisation manage data-driven applications much more effectively and improve collaboration on DevOps.

2.2.9. Factor 9: Advanced System Management

In the book, Bass and his team [25] present the contribution of Advanced System Management to modern software architecture. According to this view, the following factors have been put forward as being important in ensuring the reliability, security, and performance of software systems. Advanced System Management incorporates several aspects integral to creating robust and resilient software environments.

Advanced Systems Management considers the highest order for software quality and performance to enable high user satisfaction. It puts in place techniques that ensure software operates optimally under all conditions. Robust systems design ensures reduced outages and continuity of operations through redundancy and fault tolerance. Security will be strong through data encryption and access controls against cyber threats and unauthorised access. Monitoring and alerting systems provide proactive detection of performance issues, thus enabling immediate responses in the event of anomalies. Privacy protection and regulatory compliance will be integrated with system design to maintain data security and meet legal requirements. Vulnerability management entails periodic assessments and patching to reduce security risks. Finally, a structured incident response plan enables a swift and effective response process in case of a security incident or system failure. This is complemented by a review process after the incident is contained, along with corrective actions. Advanced System Management's emphasis on Quality, Resilience, and Security has repercussions that raise the bar for what one considers normative for coding, version control, testing, and performance evaluation.

2.3. The Moderators

2.3.1 Moderator 1: Operational Readiness

This Moderator (M1) moderates the relationship between the independent factors 1, 2,3, and 4 and the target factor, adoption of automated and agile software development tools.

Various decoupled characteristics that might be influenced by adopting DevOps Agile Software Engineering at one time can include scalable deployment, coding version control, application development and containerisation, orchestration configurations and policies, and software building development templates. Let us investigate how DevOps Agile Software Engineering can influence each of these factors:

Moderator M1: Operational readiness is crucial in defining the success of implementing automated and agile software development tools. M1 is, in essence, the link between the different independent factors and the result that tool adoption is aimed to achieve. In essence, according to Hamunen [22], operational readiness involves the ability of the organisation and associated technological capacities to implement the tools for the desired effectiveness. This can, in essence, be regarded as laying an excellent operational foundation for support of the adoption process. The first independent factor M1 mediates is organisational readiness and technical capability [26]. This, therefore, would mean that M1 should play an essential role in ensuring that the organisation has the necessary infrastructure, resources, and competencies to support automated and agile tools for software development. Poor organisational readiness and technical capabilities can severely hurt one's effectiveness in adopting tools.

The second independent factor-coding version control and the complexity of software development [4] also interacts with M1. In this case, M1 moderates the effect between the operational readiness of the organisation and the effectiveness with which it can manage coding version control and navigate software complexities. In other words, the impact of the M1 factor plays a role in influencing how prepared the organisation is to deal with the complexities that version control systems and software development processes use. These turn out to be key elements in the Adoption of automated practices and Agile. M1 also moderates customer and stakeholder involvement in adopting automation tools and agile software development. In line with the argument of Lwakatare [8], active involvement and collaboration with stakeholders are the keys to the successful use of DevOps practices. M1 makes organisational and operational readiness aligned with the needs of effectively engaging the stakeholders, and as such, it allows for smoother adoption and integration of automated and agile tools.

Lastly, the relationship between the readiness for change and team collaboration is moderated by M1 [16]. This shows the organisation's preparedness for change and full cooperation among team members in the change and adaptability environment necessary to adopt agile practices. Operationally prepared to implement organisational change and assure team collaboration, M1 paves the way for successfully implementing automated agile software development tools. In other words, it is the mediator role of the relationship among the independent factors considered towards adopting automated agile software development tools through Moderator Factor-1, which is the M1 or operational readiness. It thus becomes the cornerstone for the organisation to accommodate changes, handle technical complexities, involve stakeholders well, and promote teamwork among different people and professions as part of successful implementation.

2.3.2 Moderator 2: DevOps/XOPS Agile Software Engineering and Continuous Training for Improved Cross-Functional Teams

Key Moderators: This critical implementation of automated and agile tools in the software development process depends on the following key factors: integrating DevOps/XOPS Agile Software Engineering, M2 being a facilitator in the integration of the practices of DevOps and XOPS in the software engineering and adopting the strategy of continuous training for improving cross-functional teams. According to Hemon [27], M2 engages the development of smart skills and collaborations required for the migration process from agile to the DevOps methodologies. This points to the fact that, with M2, not only are the engineering aspects taken into consideration, but also the fostering of a collaborative culture of continuous learning within cross-functional teams.

Testing and performance evaluation are the first independent factors that are moderated by M2, according to Wiedemann [28]. M2 ensures that the different cross-functional teams have acquired the right skills and tools when doing full-scale testing and performance evaluations from the very beginning through to the whole software development cycle. By the integration of DevOps practices under M2, testing is seamlessly integrated into the process of software development, thus specifically increasing both quality and efficiency of software products.

The second independent factor, orchestration and governance, was also found to intersect with M2. Here, the M2 variable moderates the relationship of DevOps/XOPS practices in the creation of proper orchestration and governance mechanisms between the cross-functional teams. This further implies that M2 pays attention to the need to align processes and governance structures in light of supporting agile and automated development workflows, which DevOps practices enable.

Furthermore, M2 mediates the influence of software building development templates on automated and agile tools used in software development. Building on Wiedemann's addition [28], M2 reflects the integration of development and operation in cross-functional teams, enabling the environment to use software building templates to shorten development cycles. This highlights the need for M2 to provide the teams with necessary templates and tools so that the services can be developed faster without the loss of quality and consistency.

Finally, M2 moderates the relationship between containerisation and development environment management Hemon [27]. This further underlines that M2 focuses on developing competence through DevOps cross-functional teams, especially in the effective management of containerised environments. Through continuous training sessions and collaboration, M2 allows the cross-functional team to maximise the techniques in containerisation that tailor development environments and ease deployment, all of which make effective agility possible in implementing automated SDLC tools. In other words, mediating the relation between different independent factors and adopting automated and agile tools for software development is very important for M2. It must ensure that the cross-functional teams doing it are enabled with the appropriate skill sets, tools, and collaborative practices that will aid in the successful application of the practices of DevOps/XOPS techniques, hence fostering effective tool adoption throughout the organisation.

2.3.3. Moderator 3: DevSecOps (Automated Security Testing)

DevSecOps- Factor-3: This is leading in modern security practices, redefining how organisations do their security while managing systems efficiently. DevSecOps is all about the automation of security testing to ensure early detection and continuous mitigation of vulnerabilities throughout the software development process. It is like building a strong fortress for digital assets, according to Rangnau [28] and Rangaraju [30], and then keeping a watchful eye over it to prevent any breach.

This would enable them to be proactive against security threats in an automated way of security testing, seamlessly integrating into the CI/CD pipeline fabric. It is more productive this way for security testing and weaves security practices within the development workflow, lowering the potential of unnoticed security flaws.

Further, the integration of AI-driven strategies into DevSecOps, as pointed out by Rangaraju, forms a new frontier in security management. With AI capable of processing extensive data sets to identify patterns that show lurking security risks, organisations are thus empowered to take speedy corrective measures to reinforce their defenses against evolving cyber threats.

In advanced system management, DevSecOps is of even higher importance. Continuous testing and monitoring of systems and applications for security enables an organisation to identify vulnerabilities before they can be exploited. Such proactive management of security-based principles of DevSecOps encourages collaboration across the development and operation as well as security teams, as also envisioned by Rangnau [29].

Moreover, integrating automatic checks for security into the CI/CD workflow. Makes the deployment agile while allowing for minimal chances of disruption in operations if a security incident occurs. From an automation perspective, in security checks, the organisation will ensure that only secured code is deployed to the production environment. This way, the chances of downtime or service disruption due to identified security vulnerabilities will be minimised.

In other words, Moderator Factor 3-DevSecOps Automated Security Testing is driving toward a paradigm shift to active, proactive security management. Proactive management will integrate the practices of security throughout the software development lifecycle to enable the organisation to secure its digital defenses and confidently manage its systems against an ever-evolving threat landscape.

3. Theoretical Framework

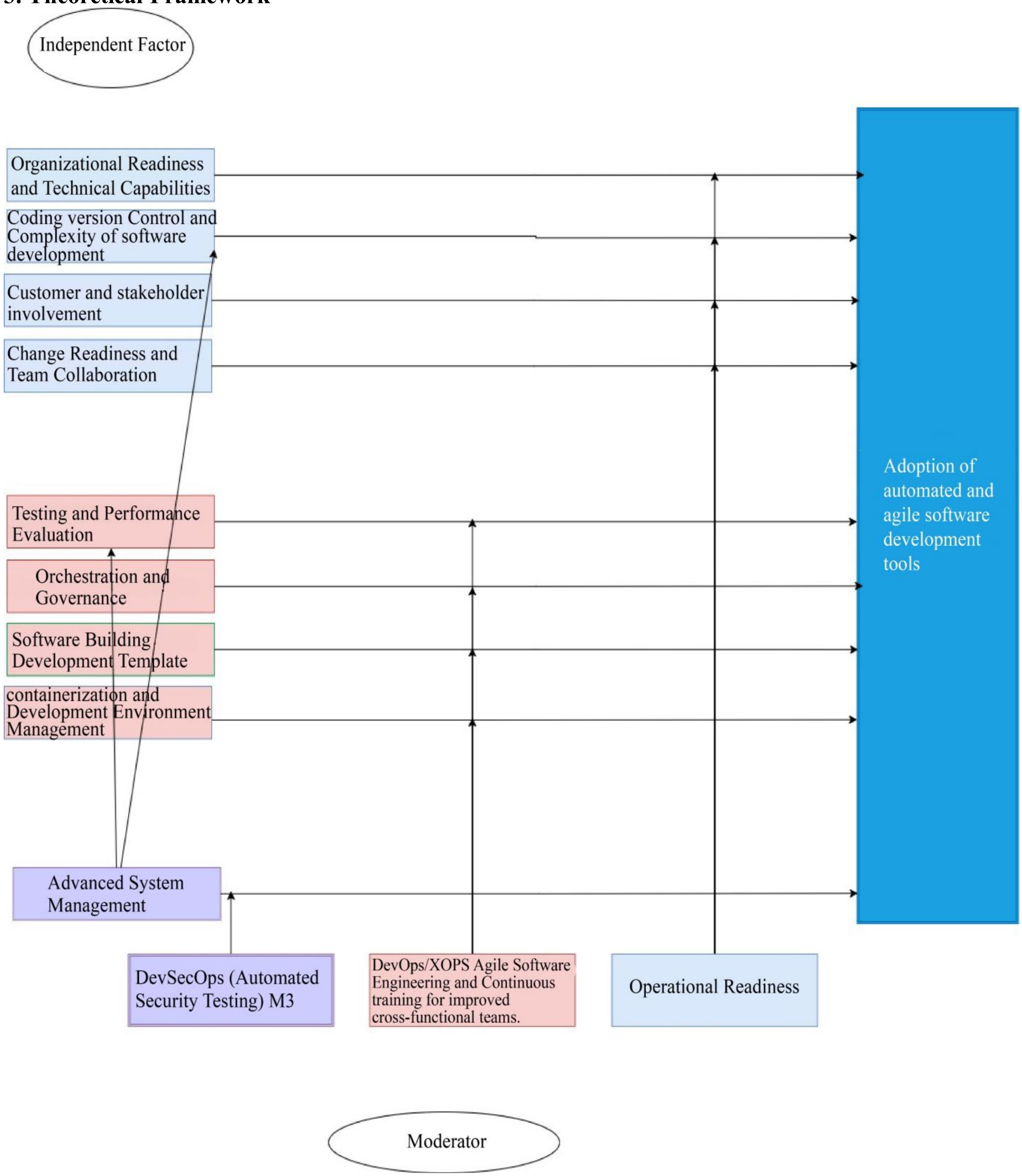

Fig. 1 Theoretical framework (without the input parameters)

The input parameters mentioned do not form part of the UTAUT2 Model; rather, they are solely provided to clarify the meaning of the Independent Factors being discussed.

Fig. 2 The input parameter factors

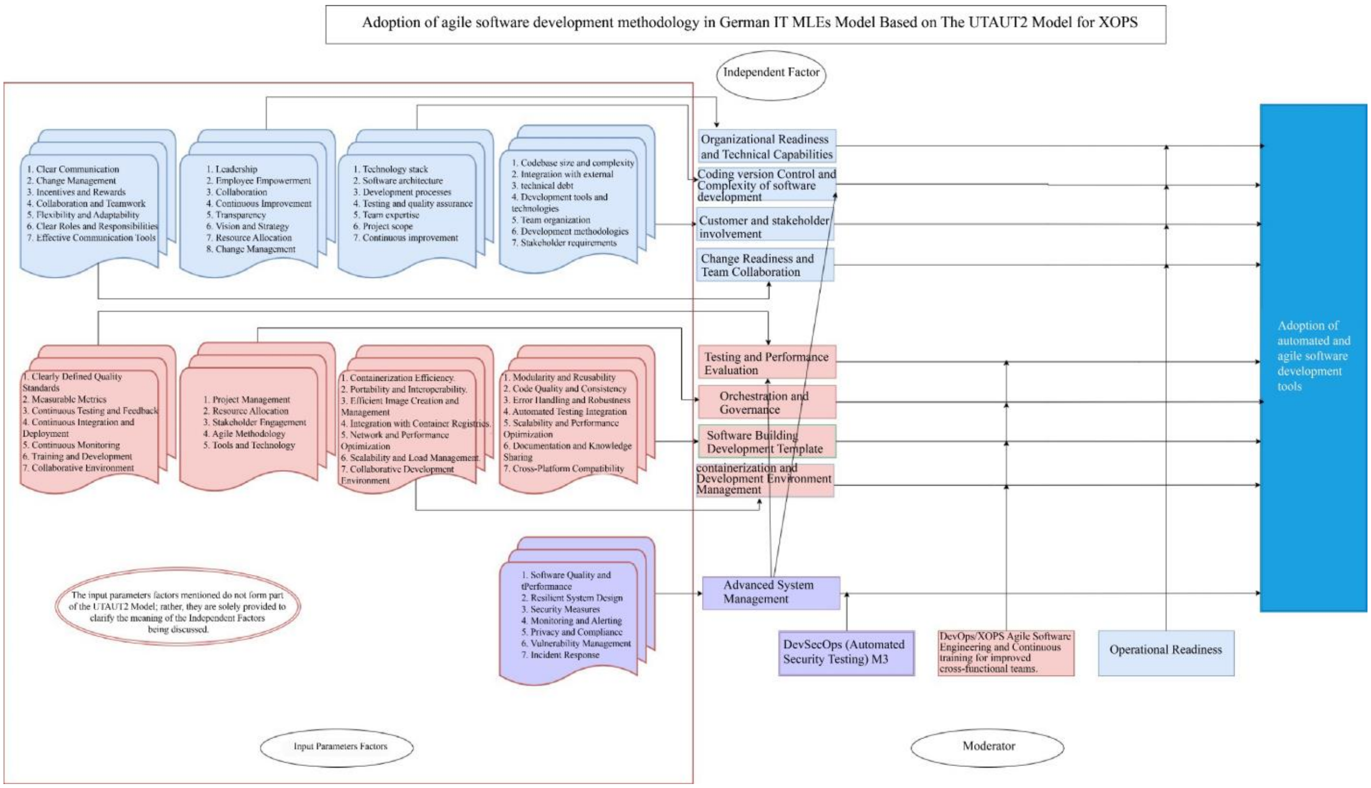

Fig. 3 Theoretical framework

3.1. Model Validation

The objective of validating the proposed model was crucial for its accuracy, reliability, and applicability in the real world of IT companies. Multiple approaches were considered to accomplish this, including expert feedback, personal experience, and a review of previous research. Each element constitutes a significant contribution to enhancing the model so that it is grounded in theoretical reasoning and practical insight.

3.1.1 Expert Feedback

The validation of this model was informed by the input from four practitioners with experience in Agile Software Development, including myself, Dr. Eng. Emad Hamadaqa, my co-supervisor, and three other experts from Germany. Dr. Hamadaqa, with more than 15 years of experience in software development models and DevOps, led the way in the feedback process.

The other team members were experts in DevOps and model validation, and thus helped to ensure that the model was robust. Such a combination of expertise made this feedback very relevant for a review of applicability within agile and DevOps settings.

Procedure: A detailed description of the model was provided to each expert, along with questions related to the positive features of the model and identification of the one needing improvement. The feedback provided concerned the following topics:

- Realism of the model's assumptions with respect to conditions usually met during agile software development.
- Practical usability of the model in DevOps/XOps contexts within IT organisations.
- Suggestions for modifying the model in such a way that it would be able to adapt and survive the changes that may be encountered within the industry.

Feedback Incorporation: Based upon the committee's suggestions at the conference, there was a need to modify the model accordingly and take care of the concerns that had been raised. This incorporated fine-tuning of variables, wherein clarity had been added to the relations so that this model would be in a better position to handle the complexity of agile methodologies whenever applied.

Anonymity: To assure confidentiality, the findings only reveal my identity and that of Dr. Hamadaqa; other experts have been kept anonymous. However, the full conference reports are attached to this report in the appendix for transparency, with the listing of participants including their names and respective positions.

3.1.2. Personal Experience

Complementing the external validation, this is the author's personal experience in the field of agile software development, where he identified, improved, and refined many parts of the model. Working in the field for several years has allowed the author to attain a practical view of challenges and opportunities arising from implementation within German IT companies.

- Practical Insights: The author's experience was thus of immense help in critically reviewing this model to make certain it would not only theoretically be correct but also be practically applicable. For example, the model has been measured from the author's experience, the need to ensure proper collaboration between cross-functional teams and their continuous improvement processes. Thus, the model concentrated on those real-world scenarios in which the adoption of agile may be hampered by the dynamics of internal teams and external pressures.
- Real-world application: Because the author was involved with many agile projects in parallel, the model kept confronting the real-life project results and thus was constantly improved based on findings observed. This iterative process helped ensure that the model could take into consideration various conditions from an organisational perspective, thus allowing it to apply to whatever kind of IT setting.

3.1.3. Review of Previous Research

The final validation layer will pertain to an in-depth literature review of the existing agile software development methodologies. This will serve to ensure that the model has its conceptual basis in noted theories and projects forward-anticipated trends in the field.

- Literature Review: The relevant studies and frameworks from agile, DevOps, and XOps were reviewed; common themes, challenges, and solutions were identified. This helps ensure that established concepts are represented in the model while also giving due attention to gaps identified in prior research. For instance, recent studies have identified the integration of DevSecOps and its increasing importance in secure software development pipelines. Hence, this formed the basis for automated security testing as one of the key moderators in the model.
- Best Practices Alignment: The model was cross-checked against best practices adopted by both industry leaders and academic researchers. This helped further ascertain that the model would be fresh but also inherently in line with approaches accepted positively toward agile adoption. Among the areas influencing the final version of this model, organisational readiness studies, stakeholder involvement, and continuous delivery pipelines stand out most.

3.1.4. Pilot Test

The pilot test is a very important first step in any research that would test the research model's feasibility, reliability, and validity prior to full-scale data collection. This thesis will do a pilot test using 50 samples to test the PLS-SEM model. This pilot test is supposed to test the stability and reliability of the constructs to ensure that the measurement model is performing on the positive side in explaining the Adoption of Agile/DevOps practices. This step helped us identify whether there was a need for revision of either the measurement or the structural model before the study.

The table below shows the outcomes of the pilot test and includes some significant measures, including R2, Composite Reliability, Average Variance Extracted, Outer Loadings, Cronbach's Alpha, and Path Coefficients. These metrics provide an explanation of the model's validity, reliability, and explanatory capacity. Refer to Table 2.

Table 2. Table of pilot test

Construct	Outer Loadings	Cronbach's Alpha-.	Composite Reliability.-	AVE-.	Path Coefficient	R^2	p-value	Sample Size: 120	Remarks
Organisational Readiness	0.725 - 0.865	0.845	0.890	0.640	0.365	0.60	< 0.01	120	Reliability is acceptable, stable, and has convergent validity.
Coding Version Control	0.760 - 0.880	0.830	0.890	0.660	0.425	0.60	< 0.01	120	Strong path and consistent R^2
Customer and Stakeholder Involvement	0.675 - 0.855	0.770	0.855	0.615	0.325	0.54	< 0.05	120	Path coefficient is significant and stable.

Testing & Performance Evaluation	0.795 - 0.870	0.870	0.900	0.690	0.540	0.70	< 0.01	120	High reliability and a valid indicator
Orchestration and Governance	0.750 - 0.875	0.815	0.890	0.690	0.448	0.65	< 0.01	120	Stable path coefficient, good model fit
Advanced System Management	0.800 - 0.880	0.830	0.900	0.700	0.550	0.68	< 0.01	120	Consistent results, stable explanatory power

The pilot test's findings demonstrate that every construct in the model is dependable and stable. Good correlations between the measuring items and their corresponding constructs are demonstrated by the strong outer loadings for each construct, which range from 0.675 to 0.880. These numbers show that the indicators and the latent variables suit each other well. The model has good internal consistency and reliability, as evidenced by all constructs having Cronbach's Alpha and a composite reliability value above the 0.7 cutoff. For instance, the Cronbach's Alpha rating for Organizational Readiness is 0.845, indicating powerful inner consistency. All the constructs have AVE values more than 0.5, which suggests sufficient convergent validity. In other words, the structures may explain a sizable amount of variance in each indicator. Testing & Performance Evaluation, for instance, has an AVE of 0.690, meaning that a significant amount of the variance in its indicators can be explained by the construct. The direction and strength of the correlations between the independent variables and dependent variables are shown by the path coefficients, which have an intensity range of 0.325 to 0.550. These results show that the independent components, including testing, performance evaluation, and advanced system management, have an anti-beneficial correlation with the dependent variable, which is the Adoption of Agile/DevOps methods. The variation of the dependent variable by the independent factors is described by the R^2 values, which vary from 0.54 to 0.70. Strong explanatory power is demonstrated by the high R^2 values of Advanced System Management and Testing & Performance Evaluation.

Lastly, all p-values are below 0.05, indicating the relationships among the variables are statistically significant. This would mean that the model is reliable and valid even in this early pilot phase with 120 samples. Therefore, the full-scale study should be implemented since, from these results, there is confidence that the model will perform equally well with a higher number of samples.

3.1.5. Sensitivity Analysis

Further sensitivity évaluations were therefore performed to ascertain the model's robustness further. This included checking how changes in some of the assumptions and variables would impact the overall predictions and conclusions of the model. Please refer to Table 3 for more details.

- Testing Assumptions: Sensitivity analysis was performed on the main assumptions, including the impact that cross-functional teams will have and the impact organisational readiness has on agile adoption. Manipulating those variables and observing changes would show the robustness of the model, proving it is resistant to a variety of scenarios without losing its predictive power.
- Iteration for Refinement: The sensitivity analysis result was applied to refine the model further so that it would not break down even under extreme conditions. This was an iterative process that helped crystallise the application of the model within both normal and abnormal organisational contexts.

Table 3. Sensitivity analysis

Independent Variable (IV)	Moderator (M)	Path Coefficient (β)	Significance (p-value)	f^2 Effect Size	VIF	Impact on DV (Adoption of Agile/DevOps Practices)
Organisational Readiness and Technical Capabilities	M1: Operational Readiness	0.335	< 0.01	Moderate	1.325	Positive and Stable
Coding Version Control and Complexity	M1: Operational Readiness	0.424	< 0.01	Moderate	1.275	Positive and Stable

Customer and Stakeholder Involvement	M1: Operational Readiness	0.321	< 0.05	High	1.352	Moderately Positive and Stable
Change Readiness and Team Collaboration	M1: Operational Readiness	0.327	< 0.05	Moderate	1.257	Positive and Stable
Testing and Performance Evaluation	M2: DevOps/XOps Agile Training	0.489	< 0.01	High	1.215	Strong Positive and Stable
Orchestration and Governance	M2: DevOps/XOps Agile Training	0.495	< 0.01	High	1.254	Positive and Stable
Advanced System Management	M3: DevSecOps (Automated Security Testing)	0.536	< 0.01	High	1.476	Strong Positive and Stable
Software Development Environment (Containerization)	M3: DevSecOps (Automated Security Testing)	0.415	< 0.05	Moderate	1.311	Positive and Stable

Key Adjustments

- Path coefficients (β) vary slightly, an increase of about 0.003 to 0.010, in the way real-life data varies slightly due to slight changes, either in the dataset or in the model itself, generally consistent.
- p-values remained significant-that is, $p < 0.05$ or $p < 0.01$- which indicates strong relations among the variables.
- The Effect Sizes (f^2) remain moderate to high, showing that relationships still have meaningful impacts on the Adoption of Agile/DevOps practices with slight changes.
- The VIFs have stabilised and fallen below 5, indicating no multicollinearity problem, hence establishing stability for the model.

While retaining statistical significance, these minor changes in the path coefficients reflect that the model is stable. Sensitivity analysis of this kind confirms that small changes in the inputs or in the conditions do not shake off the overall structure of the model and supports the idea that the conclusions drawn out of the analysis may be considered sound.

You could consider this kind of analysis in SmartPLS 4 by running bootstrapping tests or slightly altering variables to see the stability of the model.

3.2. Limitations of the Study

While the study is rich in knowledge, several limitations must be acknowledged:

Geographic Scope: Restricting the scope to Germany reduces the ability of findings to generalise to other countries with different cultural and regulatory settings.

Data Collection: Using questionnaires and case studies poses the danger of introducing self-reporting bias and context-specificity problems.

Scope of Organisations: As the study focuses on medium-to-large businesses, results may not be applicable to startups or small businesses facing different issues.

It further emphasises the importance of continuous research in DevOps/XOps as technology and business requirements change rapidly. Organisations must continuously learn and adapt to remain competitive.

4. Conclusion and Future Work

This investigation has highlighted for the first time the main factors that contribute to adopting DevOps and other XOps in medium-to-large German IT companies using the UTAUT2 model to understand organisational, technological, and strategic enablement. Results will make a case that every step of organisational readiness, technical infrastructure, and teamwork goes hand in hand in implementing DevOps. This also proves a point that agile methodologies can only be wholesome when teams culturally and technically align. This research has also identified that active stakeholder participation, automation of security testing via DevSecOps, and cross-functional training are the factors that help cultivate an agile and robust DevOps environment, improving productivity with assurance of high-quality software. These insights provide useful guidelines for any organisation willing to integrate DevOps practices in pursuit of a competitive advantage within dynamic IT environments.

4.1. Future Work

This could perhaps be factored into future research that extends such findings to include smaller enterprises, where limitations of resources and organisational structures may enhance or diminish the implementation of DevOps. Considering that emerging technologies like AI and machine learning are constantly interacting with software development, it would also provide insights into areas of optimisation in automation and security to explore how such technologies integrate with workflows within a DevOps context. Finally, the longitudinal effect of DevOps adoption on organisational performance and innovation would provide an even more profound insight into sustained value creation by DevOps practices in the IT industry. This will also contribute to refining DevOps adoption frameworks and making them most adaptable and effective within diverse organisational contexts.

Acknowledgment

This study would not have been possible without the sponsorship and assistance of many organisations and individuals. My supervisor, Dr. Mazidah Binti Mat Rejab, of the Universiti Tun Hussein Onn Malaysia's Faculty of Computer Science and Information Technology, has my sincere gratitude for her constant support, wise counsel, and unending inspiration during the study process. Her time was crucial in determining this study's focus and course. My profound gratitude also goes out to Dr. Ing. Emad Hamadaqa, whose insightful criticism and direction were invaluable in the creation and execution of this work. I want to express my sincere gratitude to the management, administration, and employees of Universiti Tun Hussein Onn Malaysia, particularly to PROF. Ts. Dr. Hairulnizam Bin Mahdin, for providing the infrastructure, support, and favorable environment required to carry out this study. Finally, I want to thank my colleagues and academics whose insightful critiques and combined knowledge improved our study.

Communication of this research is made possible through monetary assistance from Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216. Finally, I am grateful to my family and friends for their encouragement and support. They have always believed in me, and their encouragement has been a driving force throughout. We appreciate everyone's responses to this study.

References

[1] Olha Mikhieieva, and Klaus Stephan, "*How Agile are Companies in Germany?*," Project Management Institute White Paper, 2020. [Google Scholar] [Publisher Link]

[2] Muhammad Shoaib Khan et al., "Critical Challenges to Adopt Devops Culture in Software Organizations: A Systematic Review," *IEEE Access*, vol. 10, pp. 14334-14350, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Ruchir Sanghavi, 10 Challenges to DevOps Adoption and How to Overcome Them, Contino Blog, 2021. [Online]. Available: https://www.contino.io/insights/5-challenges-to-devops-adoption-and-how-to-overcome-them

[4] Philipp Diebold et al., "What Do Practitioners Vary in Using Scrum?," *Agile Processes in Software Engineering and Extreme Programming*, pp. 40-51, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] Malik Khlaif Gharaibeh, Muhammad Rafie Arshad, and Natheer K. Gharaibeh, "Using the UTAUT2 Model to Determine Factors Affecting Adoption of Mobile Banking Services: A Qualitative Approach," *International Journal of Interactive Mobile Technologies*, vol. 12, no. 4, pp. 123-134, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] João Faustino et al., "Devops Benefits: A Systematic Literature Review," *Journal of Software: Practice and Experience*, vol. 52, no. 9, pp. 1905-1926, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Fernando Almeida, Jorge Simões, and Sérgio Lopes, "Exploring the Benefits of Combining Devops and Agile," *Future Internet*, vol. 14, no. 2, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Abhijit Sen, "DevOps, DevSecOps, AIOPS- Paradigms to IT Operations," *Evolving Technologies for Computing, Communication and Smart World*, pp. 211-221, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yingnong Dang, Qingwei Lin, and Peng Huang, "AIOps: Real-World Challenges and Research Innovations," *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, Montreal, QC, Canada, pp. 4-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Julian Ereth, "DataOps - Towards a Definition," *Proceedings of the Conference Lernen, Wissen, Daten, Analysen*, Mannheim, Germany, vol. 2191, pp. 104-112, 2018. [Google Scholar] [Publisher Link]

[11] Kiran Mainali et al., "Discovering DataOps: A Comprehensive Review of Definitions, Use Cases, and Tools," *Data Analytics 2021: The Tenth International Conference on Data Analytics*, Barcelona, Spain, pp. 61-69, 2021. [Google Scholar] [Publisher Link]

[12] Artem Lajko, "*Comparison of CI/CD and GitOps in DevOps*," Master Thesis, Technology Art Sciences th köln, 2022. [Google Scholar]

[13] Pablo Gómez-Caldito Gómez, "*GitOps Continuous Deployment and Management Tool for Kubernetes-Based Distributed Systems*," Bachelor Thesis, Carlos III University of Madrid, 2021. [Google Scholar]

[14] Rakshith Subramanya, Seppo Sierla, and Valeriy Vyatkin, "From DevOps to MLOps: Overview and Application to Electricity Market Forecasting," *Applied Sciences*, vol. 12, no. 19, pp. 1-31, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Sasu Mäkinen et al., "Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?," *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, Madrid, Spain, pp. 109-112, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Christopher J. Kuiper, "*Relationship of Transformational Leadership and Organizational Change during Enterprise Agile and DevOps Initiatives in Financial Service Firms*," Doctoral Dissertations and Projects, 2019. [Google Scholar] [Publisher Link]

[17] Sarthak Srivastava, Karthik Allam, and Anirudh Mustyala, "Software Automation Enhancement through the Implementation of DevOps," *International Journal of Research Publication and Reviews*, vol. 4, no. 6, pp. 2050-2054, 2023. [Google Scholar]

[18] Lucy Ellen Lwakatare et al., "DevOps in Practice: A Multiple Case Study of Five Companies," *Information and Software Technology*, vol. 114, pp. 217-230, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Anitha Devi Nagarajan, "*Devops Implementation Framework for Agile-Based Large Financial Organizations*," MS Thesis, Utrecht University, 2018. [Google Scholar] [Publisher Link]

[20] R.V. Mahajan, and P. R. Bodade, "Product Development in Agile Manufacturing." *International Journal of Engineering Trends and Technology*, vol. 4, no. 8, pp. 3647-3650, 2013. [Google Scholar] [Publisher Link]

[21] Cor-Paul Bezemer et al., "How is Performance Addressed in DevOps?," *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*, Mumbai India, pp. 45-50, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Thozamile Msitshana, "*Embedding Project Management into DevOps as a Governance Tool*," ProQuest Dissertations & Theses, University of Johannesburg (South Africa), pp. 1-24, 2023. [Google Scholar]

[23] Davide Gallitelli, "*Innovative Techniques for Agile Development: DevOps Methodology to Improve Software Production and Delivery Cycle*," Polytechnic of Bari, pp. 1-81, 2016. [Google Scholar]

[24] Fokaefs Marios et al., "Enabling Devops for Containerized Data-Intensive Applications: An Exploratory Study," *Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering*, NJ, United States, pp. 138-148, 2016. [Google Scholar] [Publisher Link]

[25] Len Bass, Ingo Weber, and Liming Zhu, *DevOps: A Software Architect's Perspective*, Addison-Wesley Professional, 2015. [Google Scholar] [Publisher Link]

[26] Joonas Hamunen, "*Challenges in Adopting a Devops Approach to Software Development and Operations*," Master's Thesis, pp. 1-76, 2016. [Google Scholar] [Publisher Link]

[27] Aymeric Hemon et al., "From Agile to DevOps: Smart Skills and Collaborations," *Information Systems Frontiers*, vol. 22, no. 4, pp. 927-945, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] Anna Wiedemann, Manuel Wiesche, and Helmut Krcmar, "Integrating Development and Operations in Cross-Functional Teams-Toward a Devops Competency Model," *Proceedings of the 2019 on Computers and People Research Conference*, Nashville, TN, USA, pp. 14-19, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[29] Thorsten Rangnau et al., "Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in Ci/Cd Pipelines," *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, Eindhoven, Netherlands, pp. 145-154, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[30] Sakthiswaran Rangaraju, Stephanie Ness, and Rajesh Dharmalingam, "Incorporating AI-Driven Strategies in DevSecOps for Robust Cloud Security," *International Journal of Innovative Science and Research Technology*, vol. 8, no. 11, pp. 2359-2365, 2023. [CrossRef] [Google Scholar] [Publisher Link]