Assessment of the Impact of the Hybrid Software Development Approach

Shyam Gawade and Vidyavati Ramteke*
Symbiosis Centre for Information Technology, Symbiosis International (Deemed University), Pune, Maharashtra, India
Email: *vidyavati@scit.edu

Abstract. The paper focuses on enhancing the current or modern software development practices in Information Technology (IT) organizations. The research paper aims to address the gap between various hybrid software development approaches and existing practices that contribute to software development practices. The methodology adopted here is to obtain data from software professionals and understand how the current practices in the software development industry have contributed to the productivity of the employees and the IT organization, which has been done with the help of a survey questionnaire and by analyzing existing research articles and papers on the software development practices. This research is a conceptual model focused on a modern hybrid approach for software development, rather than focusing on a single software development approach, which can help IT organization increase productivity and deliver business value beyond customer expectations. The cost to build and time to market a software product or a software service can reduce by integrating the required software development practices. Today, staying relevant in the competitive market can be made easier by choosing the best available software development practices, frameworks, and models to fit an organization's value and knowledge system. The conceptual model aims to reduce the gap between various hybrid software development approaches and existing practices contributing to software development. The paper concludes by discussing the benefits of the hybrid software development approach and urge IT organizations to adopt it as per their software development requirements and customer requirements.

Keywords: Agile, Design Thinking, hybrid software development, Information technology, Application development.

1. Introduction
Today, the Information Technology (IT) industry primarily focuses on agile software development methodologies or agile methodologies for their software development practices. Agile methodologies constitute various practices like Scrum, Feature Driven Development, Test Driven Development, etc. The core principle of agile methodology is iterative software development in small and delivers rapid software builds [1]. It is important to know that applying Agile Methodology for building products and software services is not enough to remain competitive in the long run as business challenges continue to evolve. Today, the time to market a high-quality software product or a software service is important, and the essential factor in achieving this quality is to integrate a new methodology, "DevOps."
DevOps focuses on continuous integration and continuous testing of software built with various integration and testing tools. It focuses on automating the building and testing, and deployment of the developed software build [2]. Today, security is also one of the key concerns in the Information Technology sector. “DevSecOps” is a recent approach that focuses on continuous assessment of security risks and vulnerabilities present in software build with tools' help. It focuses on automating the security testing and monitoring process. Enterprise Architecture (EA) can help to improve an organization’s business and technological capabilities [3]. It can also be an effective IT investment and help an organization to reap business benefits [4].

Design thinking is a new problem-solving approach. It focuses on understanding the clients' problems and creating innovative solutions with a human-centric approach [5]. It involves empathizing with the client, understanding the problem, brainstorming for possible solutions to solve the problem. Later, the solution prototype is developed based on client feedback, and this process continues till a complete solution is designed for the client [6].

Today, research is focused on a particular aspect of software development methodologies. However, it is important to address the current market need by integrating design thinking approaches to the Agile development approach, DevOps, etc., to ensure that software product or service stands above the competition [7]. The research paper aims to address the gap between various hybrid software development approaches and existing practices that contribute to software development practices [8]. A conceptual framework designed aims to address the gap. A survey questionnaire has been sent to professionals to obtain their feedback on who is employed in the IT sector to assess the impact of existing software development practices. The data obtained from the survey questionnaire is analyzed qualitatively and quantitatively to understand the software development practices [9].

The research scope focuses on studying modern software development practices like agile methodologies and how DevOps, Design thinking, DevSecOps, and Enterprise Architecture can leverage the existing core principles of agile methodologies for improved productivity of organizations [10].

2. Literature Review

Agile software development (ASD) has gained significant importance in today’s software development practices. However, new practices and technologies continue to evolve, which allow software development companies to deliver business value to their customers in the face of increasing competition. Scrum is one of the most popularly used frameworks in agile software development [11]. Other frameworks in Agile software development include Kanban, Extreme Programming, Feature Driven Development, etc. Kanban plays an important role in visualizing and tracking various implementation tasks [12]. Agile software development practices possess the qualities of being nimble, flexible, quick, and adapt quickly to changes in requirements [13]. Agile-based software development teams are cross-functional. The agile practices have also helped the teams to collaborate closely. In Agile frameworks like Scrum, software development is done in sprints that are iterative and focus on delivering the product in small increments [14]. Product Backlog is a concept in Scrum which focuses on user stories which are requirements of a business solution. These user stories form a part of a smaller increment delivered as part of a business solution to a client in a sprint [15].

Productivity is of crucial importance for success in any project. In the Agile Software development approach, the important parameters that influence productivity are the team's composition, dependencies of external factors, and employee turnover. The Agile team needs to be designed to consider factors of the size of a team, skills of team members, the personality of team members, and time required to complete a project that will be allocated to an agile team [16]. Project factors like project constraints, development tools can also impact productivity [17].

A project’s required time to complete is usually estimated and tracked on designing, development, and testing activities. However, it is also necessary to consider all activities undertaken to complete a project, like a consultancy, training, and any other activities required to complete a project [18]. The quality of a software product has gained prominent importance to stay competitive in the IT market.
Major factors contributing to IT projects' success are based on effective practices in project management, systematic development, and testing processes [20].

The software's maintenance mechanism is considered a separate process built is not clearly defined in the Agile software development approach [21]. DevOps is a practice that can help Agile Software Teams overcome the gap between the Development Team and Operations Team, which can help make the Maintenance mechanism or maintenance process of software easier [22]. DevOps' focus is also to focus on Continuous Integration (CI) and Continuous Delivery (CD). DevOps can enable collaboration between the teams working in silos for the smooth delivery of business solutions. It can help to reduce lack of communication and help to resolve problems quickly [23]. DevOps facilitates rapid delivery of software builds, which is quite important in today's highly competitive world. DevOps can integrate easily with the popular Scrum Framework. Various tools are required to ensure smooth integration [24].

Tools like GIT, Jenkins, and Maven help for continuous integration and continuous delivery [25]. Logging tools like Log4j play an important role in keeping a log of errors generated for DevOps [26]. A hybrid approach of Agile software development practices like Scrum and DevOps' practice has shown a positive effect of improving deployment frequency, reducing the number of defects, and improved the number of user stories completed by a team in a sprint [27]. It is also necessary to have an infrastructure for an organization to support DevOps practice. DevOps has helped to achieve a cycle time of minutes for companies like Google and Amazon [28]. DevOps practices are specially implemented for Cloud-based software where frequent release practice is quite common. Today, it is important to measure the impact of software development practices to improve continuously. Metrics like MTTR (Mean Time to Repair) can help an organization check if it has improved over its past practices [29].

Security aspects of DevOps have gained attention in the past few years. This practice is known as DevSecOps. The focus is on ensuring security is emphasized from the initial phase of solution development and also focus on security aspects of CI and CD practices. Security must be addressed in the initial phase. It can be costly to ensure that all security aspects are focused when the solution has been developed. An organization also needs to ensure security aspects are covered for non-functional requirements. The buzzword “Shifting security left ” pays attention to security aspects of non-functional requirements in the initial phase of solution development. The objective of DevSecOps is to break the silos between the development team, security team, and operations team and be an enabler where all these teams can work in tandem. Tools like Sonar Cube, which is a source code quality management tool, can help for DevSecOps practices. Building automation test scripts that can check security aspects of a business solution considered a DevSecOps practice. The aim is to reduce security vulnerabilities from the initial phase of development [30], which can save huge costs associated with fixing security vulnerabilities later.

Practices and technology continue to evolve. The other recent trend has been Micro services, an architectural style that focuses on building a service (set of software functionalities) focused on serving a typical business function. Micro services have proven quite beneficial for software systems that are built frequently. This architecture has also benefitted the cloud-based software system. DevOps and Micro services provide an abundance of new concepts and technological trends that can pave exciting software development pathways. DevOps and Micro services' approach can prove quite helpful for an organization since the core principles of frequent deployment are found in DevOps practice and used for deploying Micro services.

Software solutions can be complex and interface with many other business components, legacy software components, or modern software components. Enterprise Architecture (EA) serves as a practical guide for an organization to efficiently manage software solutions and help for an organization's business transformation. An organization's enterprise architecture should enable an organization to elaborate on its strategy and ensure coordination among its projects and programs and achieve its long-term business vision.
EA artifacts can help a company make investment Information Technology (IT) decisions and gain key strategic insights. EA is a crucial link between IT and business and for an organization. The more mature an organization uses EA, the better it can reap the benefits and be successful with its IT investments. EA can help streamline the organization's business processes, efficiently use IT infrastructure, efficiently integrate business units, and promote rapid software solutions delivery. IT Governance is an important aspect of EA. IT Governance and EA can assist an organization to achieve its various enterprise-wide goals. It also facilitates gap analysis and helps an organization identify legacy systems and improvise on its software solutions, and microservices are of key importance in today's world as they can accelerate digital transformation for a business [31].

In today's world, the choice of software development methodologies and practices is an important concern as a single approach may not be a solution for creating today's business solutions where technologies keep on evolving, and project requirements can change dynamically. Organizations tend to use multiple methodologies for a project. Such multiple approaches can be termed as a hybrid approach for software development. The other reasons for choosing a hybrid approach depending on the project budget and the project's criticality (high, medium, or low). Another reason for selecting a hybrid approach can also depend on an organization's experience with various IT projects and its team members' work experience working in the IT industry. Business constraints, organization constraints, client constraints, improving flexibility of operations and evolving existing practices, and being pragmatic to manage software complexities are reasons for choosing a hybrid software development approach. The hybrid approach has been a natural evolution over some time. The hybrid software development approach has been gaining popularity in the IT industry; it can help manage different stakeholder groups' requirements in an organization.

Design Thinking (DT) is an approach or a process to empathize with a customer, define insights gained by empathizing with a customer, brainstorm and develop creative solutions, create a prototype of the solution and test the prototype by gaining feedback from the customer. This process can be iterative. Design thinking is a human-centric approach to creating a solution. The customer point of view helps an organization to generate ideas and build possible prototype solutions to solve customer's problems. To understand problems and designing solutions, integrating Design thinking and agile software development practices can prove beneficial for organizations. The design thinking approach is flexible and iterative, helping an organization integrate with agile framework practices like Scrum and extreme programming. Design thinking integrated with agile software development methodologies can also contribute to the digital transformation of a company. Design thinking supports the requirements gathering phase of agile practices as it can help gather requirements more efficiently from a customer. The design thinking approach's benefit is not just limited to the initial stages of a software project, which is the requirement gathering and design stage; it can also be integrated with the entire software lifecycle by using Agile-based software practices. Design thinking cannot be considered an alternative to existing software development practices. Rather, it is an approach or practice that, when integrated with software development practices, can help an organization be competitive in the market and deliver business solutions that are customer-centric and based on perceptions of a customer.

3. Methodology
The literature review of research papers and articles helped to understand the importance and influence of the various practices, approaches on agile software development methodology.

The focus was on qualitative and quantitative assessments to assess the impact of the hybrid software development practices. The quantitative assessment was done with an unpaired t-test to evaluate the hypothesis, "Organization's comfort level to adopt and utilize hybrid - a hybrid approach affects its productivity." The research was done with a questionnaire-based survey, which helped qualitative as well as quantitative analysis. The questionnaire was sent to professionals in the Information Technology sector with the help of Google Forms. The survey link was sent shared with the help of social media channels. The focus was on the professionals who utilize the Agile Software
Development approach in their organization. The questionnaire focused on various aspects like the participants’ designation and project domain. The focus was also on the convenience, benefit, and perception with agile methodologies, DevOps, DevSecOps, Enterprise Architecture and Design thinking.

Participants were also asked about the practices which foster continuous improvement of their organization and adaption of new technologies. Participants were also asked to provide information about the frameworks utilized within agile software development methodologies and DevOps tools in their organization. The convenience reception about productivity benefit from hybrid software development and triggers for hybrid software development was also taken under consideration with the questionnaire's help. A criterion was decided about the participants’ perception of applying technologies and techniques with the Five-Point Likert Scale's help. The scale created is as follows: Strongly Disagree (1), Disagree (2), Neutral (3), Agree (4), Strongly Agree (5). Also, utilizing Design Thinking was rated on a scale of Five: Never (1), Occasionally (2), Sometimes (3), Often (4), Always (5).

4. Results and Discussion
1. Diversity of participants' design criterion: Responses were received from 54 IT professionals with diverse designations within the IT sector. Figure 1 shows the Diversity of participants’ designations in the IT sector.

![Figure 1: Diversity of participants’ designations in the IT sector](image)

![Figure 2: Diversity of participants’ project domain](image)
2. Diversity of participants' project domain: Our surveyed participants work in diverse project domain as illustrated in Figure 2.

3. Agile Software Development Approaches: Our survey shows Scrum is the most popularly used agile software development methodology. This result strengthens the fact that Scrum is the most popular used framework under agile software development methodology. Also, it is important to note that a hybrid approach is also used within the agile software development methodologies. It can be seen that an organization uses practices like Scrum and Kanban together. Figure 3 shows the Agile Software Development Approaches used by participants’ organizations.

![Figure 3: Agile Software Development Approaches used by participants’ organizations](image)

4. Convenience with Agile Software Development Methodology for software development: 81.49% of surveyed participants agreed with agile software development methodology for their projects, which shows that Agile Software Development Methodology has reached a mature level in today's organizations. It is important to know that the percentage of neutral or disagreed participants use agile-based software development practices, which is evident from our survey of Approaches for Agile Software Development Approaches. However, this can be a possible indicator that some organizations still need to enhance their processes to utilize agile software development effectively.

5. Convenience with DevOps practice for software development: 79.63% surveyed participants agreed with DevOps practice for their projects, which possibly shows that DevOps also has reached a mature level in today's organizations. The remaining percentage of neutral or disagreed participants indicates that some organizations still need to undergo a DevOps maturity assessment to yield better results.

6. Tools utilized for DevOps: Out of the 54 participants surveyed, 51 participants' organizations used DevOps tools for their software development process. The participants were given an option to choose one or more tools or add any tool used in their organization. The graph showed that Git one of the popularly used DevOps tools. It is also important to note that the participants' surveyed organizations used a combination of tools. Figure 4 shows the Tools utilized for DevOps by participants’ organizations.

7. Concern of DevSecOps approach or Security approach for DevOps: In the survey, it was seen that 50% participants’ organizations followed the DevSecOps approach, which shows that organizations need to catch up with the DevSecOps approach.
Figure 4: Tools utilized for DevOps by participants’ organizations

8. Implementation of enterprise architecture: In the survey, it was found that 59.26% of our participants’ organizations focus on utilizing enterprise architecture mostly for their larger projects. The project size and budget may influence in decision-making of whether enterprise architecture is required.

9. Metrics to improve processes: Continuous improvement is important for any organization to succeed in today’s competitive world. Various metrics help an organization to improve continuously. It was no surprise to see that 85.18% of our participants’ organizations focus on metrics as a part of their continuous improvement in the survey.

10. Benefits of design thinking approach for an organization: 61.11% of surveyed participants agreed that their design thinking approach benefited the organization. 33.33% of participants were neutral about the benefit of this approach. It is possible that formal training in design thinking can help to increase benefit design thinking for an organization.

11. New technologies like Micro-Services or Artificial Intelligence, Data Analytics, etc., are used by organizations to leverage their business: 75.93% surveyed participants agreed with new technologies being utilized to leverage their organizations’ business, which may indicate that organizations are actively using new technologies to leverage their business today. It is also possible that these technologies can give rise to new hybrid software development methodologies in the future.

12. Organization actively supports research, proof of concepts to harness the power of new technologies and new software development practices: 81.49% surveyed participants’ organization follow this practice, which indicates a positive practice amongst IT organizations to harness new technologies.

13. Participants utilizing design thinking concepts in professional work: In the survey, it was seen that 90.74% of participants followed the design thinking approach for their professional work approach, which shows that organizations are readily utilizing the Design Thinking approach for their professional work.

14. Triggers for a hybrid approach for software development (Participants’ organizations utilize a hybrid approach for software development which is not limited to using a hybrid approach of Agile and DevOps or Agile and Design Thinking): In the survey, participants were asked for various reasons...
which triggered a hybrid approach for software development. Participants were given certain predefined options and also a choice to enter their reason to choose. The results were as follows:

A. Need to evolve existing practices for creating high-quality business solutions: 37.04% of participants agreed to this reason.

B. Project Requirements: 29.63% of participants agreed to this reason.

C. Team Members’ Experience: 12.96% of participants agreed to this reason.

D. Organizational Factors: 9.26% of participants agreed to this reason.

11.11% of our participants’ organizations do not use a hybrid approach. This possible shows that there can be various reasons for the hybrid approach for software development and the evolution of practices, one of the major reasons to adopt a hybrid approach for software development in an organization.

15. Assessing the impact of hybrid approaches on productivity:

A) Assessing project productivity improvement from using a hybrid approach of DevOps and Agile Methodology: Majority (70.37%) of our participants agreed that using a hybrid approach of DevOps and Agile methodology improved their project's productivity. With adequate training and creating proof of sample project concepts using this hybrid approach, it can benefit organizations. Figure 5 shows assessing project's productivity improvement from using a hybrid approach of DevOps and Agile Methodology.

Figure 5: Assessing Project’s productivity improvement from using a hybrid approach of DevOps and Agile Methodology

Figure 6: Assessing Project’s productivity improvement from using a hybrid approach of Design Thinking and Agile Methodology
B) Assessing project's productivity improvement from using a hybrid approach of Design Thinking and Agile Methodology: The majority of our 61.11% of our participants agreed that using a hybrid approach of Design Thinking and Agile methodology has improved their project's productivity. It is possible that formal training in the Design Thinking approach could improvise the benefit of Design thinking. Project constraints, too, can affect the selection of this hybrid approach. Figure 6 shows the Assessing Project's productivity improvement from using a hybrid approach of Design Thinking and Agile Methodology.

5. Statistical Analysis
The hypothesis is “Organization's comfort level to adopt and utilize hybrid approach affects its productivity.” To evaluate the hypothesis, “Organization’s comfort to adopt and utilize hybrid software development” and “organization has improved its productivity concerning software development after adopting and utilizing hybrid software development” were analyzed by unpaired t-test, were which used as an inferential statistic. If, the p-value is < 0.05 for the test, then it could be considered statistically significant. Out of 54 participants surveyed, 47 participants utilize a hybrid approach for software development which is not limited to using Agile and DevOps or Agile and Design Thinking. The p-value was found to be less than 0.05; as per the result of the t-test, the organization's comfort level to adopt and utilize hybrid a hybrid approach affects its productivity. A correlation test was also done for the “Organization’s comfort to adopt and utilize hybrid software development” and “organization has improved its productivity concerning software development after adopting and utilizing hybrid software development.” The correlation coefficient from the test was found to be 0.745, which indicates a strongly positive relation.

Therefore, from both the tests, it can be said that if an organization is comfortable adopting and utilizing hybrid software development, it will improve the productivity of an organization concerning its software development practice. Figure 7 shows the Conceptual framework for a modern hybrid approach for software development using Scrum.

A conceptual model of a hybrid software development framework

![Figure 7: Conceptual framework for a modern hybrid approach for software development using Scrum](image-url)
With the help of our research survey and existing literature review, the hybrid approaches and the various technical practices of implementing an enterprise architecture, DevOps, DevSecOps, Scrum, and using a practice approach of Design Thinking, the new conceptual model provides a framework to reduce the gap between various hybrid software development approaches and existing practices which contribute for software development. It may help overcome some of the challenges an organization can face to create a customer-centric business solution quickly and effectively and be a strong market competitor.

The framework can be helpful for organizations that are using agile methodologies like Scrum for software development. The conceptual model uses Scrum, as is the most popular agile practice.

A design thinking approach begins from the start of a project. A team responsible for gathering requirements and analyzing them and should utilize a design thinking approach that can help the team empathize with the customer, understand the customer's points of view, and ideate a solution that can satisfy the customer's requirements. The sprint backlog focuses on the user stories which need to be taken up for a project.

A design thinking perspective can help a team think from a user stories' implementation perspective and a customer perspective. In the development phase, a team can utilize the design thinking approach to build a more customer-centric system that is easy to use and visually attractive. The team responsible for testing can utilize a design thinking approach by utilizing and broadening the customer perspective and collaborate with the team for a more resilient, robust, and effective system. The team responsible for deployment and maintenance can utilize a design thinking approach to get a customer perspective as the favorable time to deploy a business solution, deploy a business solution, and be a part of a feedback loop from the customer.

In today's world, enterprise architecture can help streamline the organization's business processes, efficiently use IT infrastructure, efficiently integrate business units, and promote rapid software solutions delivery. Hence, a team needs to design an enterprise architecture to support the development, testing, deployment, and maintenance of the business, by focusing on the enterprise architecture during the sprint backlog creation and beginning with the design phase. However, it is also essential that the enterprise architecture evolves according to customer and project team's feedback and facilitates ease of solution delivery.

DevOps can be utilized at the start of a sprint at creating a sprint backlog stage by working on breaking the silos between the development and operations and focusing on collaboration. The collaboration efforts should be focused on steps and processes needed for a smooth solution delivery required for a particular sprint. The DevSecOps approach at the stage of creation of sprint backlog can help a team focus on the required security aspects of the solution delivered in the sprint. Continuous integration aspects of DevOps would be crucial during the development and testing phases. DevOps' continuous delivery aspect would play a major role in the deployment and maintenance of the solution. The DevSecOps approach would help the team responsible for development to add the required security to the business solution from the development phase. The team responsible for testing can verify and validate the security features added to the solution delivered in the sprint. The team responsible for deployment can verify and validate the security features required before the solution is deployed and monitor the solution during the maintenance phase for any security issues. However, here, all the teams must work in tandem to complete their responsibilities.

Sprints in Scrum are iterative. Hence, the practices of Design Thinking, DevOps, DevSecOps, and Enterprise Architecture need to adapt to the sprint requirements.

6. Conclusion
The study showed that hybrid software development approaches could benefit an organization in terms of improving productivity. An organization must continue to evolve its business processes and technology to be at ease to adapt to hybrid software development practices to achieve its business vision and goals. The conceptual framework combines the strength of Agile software development
methodology, DevOps, DevSecOps, Enterprise Architecture, and Design thinking to leverage an organization's existing software development practices.

The hybrid approach's power can benefit the organizations as it can add more flexibility for business processes and ensure success and customer satisfaction for the business solution delivered. IT professionals and their organizations are urged to adopt hybrid software development practices according to their business vision and goals, software development practices, and client requirements.

References
[1] Allshuqayrran, N., Ali, N., & Evaans, R. (2016). In microservice architecture, a systematic mapping study was conducted. Proceedings - 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications, SOCA 2016, 44–51. https://doi.org/10.1109/SOCA.2016.155
[2] Armmour, F. J., & Kaaisler, S. H. (2001). Transition and deployment of enterprise architecture in an agile manner. IT Professional, 3(6), 31–36. https://doi.org/10.1109/6294.9777769
[3] Bohh, W. F., & Yeelin, D. (2006). Managing information infrastructure using enterprise architecture principles. Journal of Management Information Systems, 24(3), 161–207. https://doi.org/10.2753/MIS0742-12222303307
[4] Carteer, K. (2017). Faranconis Raynada on DsevSecOps. IEEE Software, 33(5), 92–96. https://doi.org/10.1109/MS.2017.32571578
[5] Cesaar, J., & Reusso, R. D. F. S. M. (2018). ScienceDirect is a website dedicated to science. Design Thinking in Agile Software Development: A Systematic Literature Review Design Thinking in Agile Software Development: A Systematic Literature Review. Procedia Computer Science, 138, 775–782. https://doi.org/10.1016/j.procs.2018.10.1001
[6] Coohen, D., Lindvall, M., & Cossta, P. (2004). Introduction to Agile Method. Advances in Computers, 62(2), 1–66. https://doi.org/10.1016/S0065-2458(03)62001-2
[7] Diaaz, J., Peerez, J. E., , M. A., Meena, G. A., & Yagae, A. (2019). Self-Service Cybersecurity Monitorings as Enabler for DeevSecOps. IEEE Access, 7, 100283–100295. https://doi.org/10.1109/access.2019.29300000
[8] Dingssöyrr, T., Nerrur, S., Balljepally, V., & Moe, N. B. (2012). Towards describing agile software development after a decade of agile methodologies. Journal of Systems and Software, 85(6), 1213–1221. https://doi.org/10.1016/j.jss.2012.02.0033
[9] Doobrigkeit, F., & Dea Paaula, D. (2019). Understanding the manifestations of design thinking in software engineering is essential to understanding design thinking in practice. ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 1059–1069. https://doi.org/10.1145/3338906.33400451
[10] Doőležel, M. (2018). Potential applications of institutional theory to the study of hybrid software development principles and practices. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11271 LNCS, 441–448. https://doi.org/10.1007/978-3-030-03673-7_35
[11] Dummitriu, D., & Poopescu, M. A. (2020). IT Management Enterprise Architecture Framework Design In IT management, Enterprise Architecture Framework Design is a must. Procedia Manufacturing, 46, 932–940. https://doi.org/10.1016/j.promfg.2020.05.011
[12] Ebbert, C., Galardo, G., Herenantes, J., & Serraano, N. (2016). DevOps. IEEE Software, 33(34), 944–1000. https://doi.org/10.1109/MS.2016.688
[13] Hägerr, F., Kowaark, T., Krügers, J., Vedetterli, C., Übefnickel, F., & Uflacker, M. (2015). DT@Scrum: Integrating Design Thinking with Software Development Process. In Design Thinking Research: Building Innovators (Vol. 2, Issue 2, pp. 263–289). https://doi.org/10.1007/978-3-3109-06823-7_014
[14] Hidalgo, E. S. (2019). Adapting the scrum frameworks for agile projectsand management system in science : a case study of a distributed research initiative. Heliyon, July 2018, e01447.
https://doi.org/10.1016/j.heliyon.2019.e01447

[15] Jaabbari, R., Alli, N., & Peetersen, K. (2016). What is DevOps? A Systematic Mapping Study on Definition and Practice. ACM XP ’16 Workshops: Proceedings of the Scientific Workshop Proceedings of XP2016. https://doi.org/https://doi.org/10.1145/2962695.2962707

[16] Kuhrmann, M., Diebold, P., Müñch, J., Tesll, P., Garorusi, V., Feldderer, M., Trektser, K., McCaffery, F., Linsssen, O., shanser, E., & Prause, C. R. (2017). Hybrid software system development: Waterfall, Scrum, and beyond. ACM International Conference Proceeding Series, Part F1287, 30–39. https://doi.org/10.1145/3084100.3084104

[17] Kuhrmann, M., Diebold, P., Munvch, J., Tesll, P., Trektser, K., McwCaffery, F., Gwarounsi, V., Feledderer, M., Leinssen, O., Haanser, E., & Pruse, C. R. (2019). Hybrid Software Development Approach in Practice: A European Perspective. IEEE Software, 36(4), 20–31. https://doi.org/10.1109/MS.2018.1210161245

[18] Kuhrmenan, M., & Fwernandez, D. M. (2015). Systematic Software Development: A State of the Practice Report from Germany. Proceeding - 2015 IEEE 10th International Conference on Global Software Engineering. ICGSE 2015, 51–60. https://doi.org/https://doi.org/10.1109/ICGSE.2015.11

[19]Lee, J. S. (2018). The DevOps and Agency Theory and revolution. Proceedings - 29th IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2018, 243–244. https://doi.org/10.1109/ISSREW.2018.00013

[20] Lindbeerg, T., Köppen, E., Rauth, I., & Meeinel, C. (2012). On the Perception, Adoption, and providing of Thinking in the IT Industry. In Design Thinking Research(pp. 229–240). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-21643-5_13

[21] Luccena, P., Branz, A., Checioria, A., & Tizezei, L. (2017). IBM's software development platform for design thinking. Communications in Computer and Information Science, 680, 98–109. https://doi.org/10.1007/978-3-319-55907-0_9

[22] Meelo, C. D. O., Cruzes, D. S., Kone, F., & Conradi, R. (2013). Case studies on agile team effectiveness and management that are interpretive. Information and Software Technology, 55(2), 412–428. https://doi.org/10.1016/j.infsof.2012.09.0004

[23] Papadopooulos, G. (2015). Traditional Software Development Methodologies to Agile Software Development Methodologies On large, distributed projects, as well.Proceeding - Social and Behavioral Sciences, 175, 455–463. https://doi.org/10.1016/j.sbspro.2015.01.12233

[24] Praddhan, S., Nannaiyur, V., Mellanahalli, P., Pallaa, M., & Chullani, S. (2019). A Case Study of Quality Metrics for Hybrid Software Development Organizations. Proceedings - Companion of the 19th IEEE International Conference on Software Quality, Reliability and Security, QRS-C 2019, 505–506. https://doi.org/10.1109/QRS-C.2019.00097

[25] Sammarawickrama, S. S., & Perrera, I. (2017). Continuous Scrum is a method for incorporating DevOps into Scrum. 17th International Conference on Advances in ICT for Emerging Regions, ICTOR 2017 - Proceedings, 2018-Janua, 18–25. https://doi.org/10.1109/ICTER.2017.82578808

[26] Shoukoourian, A., & Danielely, E. (2011). How to Measure “ Soft ” Things? 2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR), 7–10. https://doi.org/10.1109/CEE-SECR.2011.61888466

[27] Terrán, C., Torrej, J., & Filores, P. (2018). Under an Agile Methodological Approach, a Productivity Model for Software Development Factories is developed. Proceedings - 2017 International Conference on Computational Science and Computational Intelligence, CSCL 2017, 1039–1042. https://doi.org/10.1109/CSCLC.2017.1880

[28] vaan dene Besrg, M., Slot, R., van Stenbergen, M., Faase, P., & van aVllet, H. (2019). How does enterprise architecture help you make better IT investment decisions?. Journal of Systems and Software, 152, 134–150. https://doi.org/10.1016/j.jss.2019.02.053

[29] Vijayaassarathy, L. R., & Buttlar, C. W. (2016). Can Organizational, Project, and Team Characteristics Influence Software Development Methodologies? IEEE Software, 33(5), 86–94. https://doi.org/10.1109/MS.2015.26

[30] Zddun, U., Wittern, E., & Letner, P. (2020). Emerging Trends, Challenges, and Experiences in
DevOps and Microservice APIs. *IEEE Software*, 37(1), 87–91. https://doi.org/10.1109/MS.2019.2947982

[31] Zimmermsaann, A., Schmidt, R., Sandkuhl, K., Jugel, D., Bogner, J., & Möhring, M. (2018). Evolution of Enterprise Architecture for Digital Transformation. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW, 2018-Octob*, 87–96. https://doi.org/10.1109/EDOCW.2018.00023