DevOps Enabled Agile: Combining Agile and DevOps Methodologies for Software Development

Shah Murtaza Rashid Al Masud, Md. Masnun, Mst. Afia Sultana
Anamika Sultana, Fahad Ahmed, Nasima Begum
Department of Computer Science and Engineering
University of Asia Pacific, Dhaka, Bangladesh

***Abstract*—The Agile and DevOps software development methodologies have made revolutionary advancements in software engineering. These methodologies vastly improve software quality and also speed up the process of developing software products. However, several limitations have been discovered in the practical implementation of Agile and DevOps, including the lack of collaboration between the development, testing and delivery sectors of different software projects and high skill requirements. This paper presents a solution to bridge the existing gaps between Agile and DevOps methodologies by integrating DevOps principles into Agile to devise a hybrid DevOps Enabled Agile for software development. This study includes the development of a small-scale, experimental pilot project to demonstrate how software development teams can combine the advantages of Agile and DevOps methodologies to fulfill the gaps and provide further improvements to the speed and quality of software development process while maintaining feasible skill requirements.**

***Keywords*—Agile; DevOps; gaps; collaboration; skill; DevOps Enabled Agile; software development**

I. Introduction

The field of software development is evolving continuously with new methodologies and techniques constantly being devised to bridge the gaps of previous ones. Our motivation is to propose and verify the DevOps Enabled Agile methodology, a methodology that will combine Agile with DevOps to fulfill the gaps of both methodologies in order to further advance the field of software development.

This study includes an in-depth analysis of Agile and DevOps methodologies, describing and comparing the different practices, requirements, challenges etc. and concluding that both methodologies have disadvantages and drawbacks that can reduce the effectiveness of individually implementing them for software development projects. On the basis of this analysis, the implementation of DevOps Enabled Agile has been justified as the solution to this problem.

For practical application of implementing DevOps Enabled Agile methodology, we propose combining standard Agile methods with the cross-functionality and collaboration principles of DevOps. We have developed an experimental pilot project using this procedure and conducted an in-depth comparison of the results gained by observing the performance of both Agile and DevOps Enabled Agile methodologies in order to demonstrate the benefits of DevOps Enabled Agile methodology.

The remaining parts of this paper are organized in the following ways; Section II presents an in-depth comparative analysis of Agile and DevOps methodologies and provides justification for combining these methodologies into DevOps Enabled Agile methodology. Section III contains the background and related works of our study in the form of a systematic literature review. Section IV describes the proposed methodology used in this study to test and verify the effectiveness of DevOps Enabled Agile for an experimental pilot project. This is followed by Section V which contains the results from the experimental pilot project and a discussion of the findings that justify the implementation of DevOps Enabled Agile. Lastly, in Section VI, the study is concluded with a brief analysis of the scope for future research.

II. Analysis of Agile and DevOps

Agile methodology is a highly efficient and effective software development methodology which is focused on four core principles [1]. These principles emphasize the importance of individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

There are many methods and techniques to implement Agile methodology depending on project scope, knowledge domain and other factors including DAD, SAFe, XP, Scrum etc. [2] Scrum in particular is an agile framework designed for small teams which is relatively simple to use and highly adaptable to changes [3].

Agile methodology provides many different benefits over traditional software development methodologies [4]. The main benefits include lower documentation requirements, strong collaboration with customers, rapid delivery and continuous alignment with business needs and goals.

However, we have discovered that Agile methodology has a number of major gaps including the separation of the development, testing and delivery sectors and lack of cross-functionality [5]. Agile methodology does not support continuous integration of feedback during development, thus limiting the speed and efficiency of software development. Additionally, the lack of cross-functionality within developer teams may result in disconnected and uncoordinated development, testing and delivery sectors which may lead to project completion and progression not aligning with planned estimates.

However, DevOps (Development-Operations) is an extension of Agile methodology which can harmonize the development and operation processes/sectors and provide cross-functionality within teams. Thus, DevOps can mitigate some of the drawbacks of Agile methodology.

DevOps is a software development methodology focused on the collaboration of developers and operators, but it lacks concrete definition and is referred to as a skillset, a set of practices and even as a job description [6]. DevOps focuses on cross-functionality between different sectors of development, operations and quality assurance which would result in highly coordinated development, testing and delivery which would lead towards smoother project progression [7].

But, pure DevOps certainly cannot focus on fulfilling continuously changing user requirements by receiving and incorporating feedback in incremental cycles. Additionally, pure DevOps has high skill requirements that cannot be fulfilled by software engineers who lack extensive training and experience on DevOps culture and practices. Thus fully incorporating DevOps into public sectors such as mid and low level software development projects would result in difficulties and drawbacks that would be unacceptable for efficient and effective software development process [8] [9] [10].

Hence, to mitigate the existing problems, we propose combining Agile and DevOps methodologies into the hybrid DevOps Enabled Agile methodology for software development. Our research and analysis have revealed that DevOps and Agile methodologies can be combined together to produce the best results and mitigate the existing problems of both methodologies. Our method will follow the examples set by related works that have suggested combining Agile with DevOps to utilize the benefits of both methodologies [11] [12] [13] [14].

Our proposed methodology, DevOps Enabled Agile has been designed to combine the collaborative and cross-functional aspects of DevOps with the simplicity and low skill requirements of Agile to bridge the gaps of both methodologies, thus ensuring rapid progression and maximum effectiveness of the development, testing and delivery sectors of software development projects. Therefore, this analysis justifies the implementation of DevOps Enabled Agile for software development.

III. Related Work

Our study includes a systematic literature review of different related works. We have screened ten existing papers in particular as the basis of our research and analyzed the problem statement, research method and limitations of these papers as presented below in order to conduct the literature review.

Marius Andersen Bjørni and Simen Haugen have conducted interviews and observations to discover the major challenges of Agile methodology which consist of high sprint workloads, lack of testing in sprints before release, poor PBI descriptions, lack of business agility, lack of documentation, inadequate PBI grooming, lack of team improvement, inefficient release process and not holding sprint reviews. However, this study has not considered work estimation during research, leaving this factor as a research gap [5].

Koi-Akrofi et al. have also conducted a study on Agile methodology based on literature review. They have pointed out the notable advantages of Agile over traditional methodologies, which include adaptability towards changing customer requirements, simple method of updating, priority assessment, feedback integration, continuous testing and motivation for developers. They have also pointed out the notable challenges of Agile methodology which are the unpredictability of development, requirement of high dedication, time and effort, greater customer demands, lack of documentation and projects going off-track. However, this study is purely a literature review and does not use any additional method of research, thus showing that fully transitioning to DevOps also has major difficulties and drawbacks [15].

Farid et al. have conducted a study on combining DevOps with Lean Software Development. This study has presented Lean Software Development as a software development methodology focusing on waste reduction and continuous improvement while presenting DevOps as a set of practices targeted towards improvement of the overall development life cycle by integrating development and operations together in cross-functional teams. The research method involves implementing DevOps practices to bridge discovered gaps in Lean Software Development. This method was able to utilize DevOps practices to overcome issues of Lean Software Development in a number of different areas including Delays, Defects, Extra Features, Task Switching etc. The limitation of this work is that the application of DevOps principles to overcome existing issues has only been tested with Lean Software Development and the exploration of similar cases of applying DevOps principles for Scrum, XP and other methods have been left as prospects for future research [16].

Banica et al. presented DevOps as a Project Management methodology for software development projects. It utilizes a pilot project as the method of research, incorporating DevOps facilities into the development process of a website using VersionOne tool which is highly suitable for project management systems supporting DevOps methodology. The results of this project show that DevOps speeds up and improves the development process. However, due to the inexperienced teams used, some problems were faced during development, including delays and missing certain development goals, which classifies as the limitation of this research [17].

Wiedemann et al. provides an in-depth analysis on the application of DevOps methodology for IT projects. A qualitative multiple case study was conducted in eight different industries that utilize DevOps to achieve intra-IT alignment, justifying the use of DevOps in IT project management in order to orchestrate development and operations within IT functions. However, this study only proposes using DevOps methodology for this purpose, neglecting the potential of combining it with Agile methodology to gain further advantages and overcome existing disadvantages, thus leaving a gap in the research [18].

Kuusinen et al. observed and analyzed the progression of two groups of students working on software projects transitioning from Agile to DevOps methodology. This research discovered several challenges and problems in the transition process, including legacy systems, issue of rights and slow testing. Applying continuous testing and the collaboration of cross-functional teams have been suggested as solutions to solve these problems. However, a more complete solution involving the utilization of Agile methodology in coordination with DevOps methodology instead of fully transitioning to DevOps methodology has been left unexplored, which is a limitation to be addressed by future research [19].

Sikender Mohsienuddin Mohammad briefly compares Agile and DevOps Methodologies by analyzing their similarities and differences. The similarities of Agile and DevOps are that both are focused on quick and efficient iterative feedback-based development of software. They can be used together in a coordinated way to speed up and improve the software development process. On the other hand, there are also some major differences between the two methodologies. These differences include Agile only taking feedback from customers while DevOps additionally takes feedback from internal team and Agile minimizing the gap between developers and customers while DevOps minimizes the gap between developers, testers and operators. However, this paper neglects to mention the challenges of DevOps methodology, leaving it as a gap in the research [20].

Hemon et al. have proposed a maturity model for transitioning from Agile to DevOps, utilizing three stages of progression; Agile, Continuous Integration and Continuous Delivery. This model focuses on the alignment of development and operations while aiming to achieve greater smartness in IS function. However, the scope of this research does not extend towards the field of software development despite Agile and DevOps having a high degree of application in this field. Thus, it can be considered as a research gap [8].

Hemon et al. conducted an additional study on the subject of Agile to DevOps transition, focusing on individual roles, collaboration and skills. The research method applied data collected from a case study in an organization with years of experience in DevOps to form the basis of the research. The results of the research conclude that DevOps allows for balanced collaboration between individuals but also requires skilled developers and operators to be used efficiently. However, the extent of more diverse collaboration has not been fully demonstrated in this study, which could be explored in future research [10].

IV. Proposed Methodology

Our study has utilized the development of a small scale, experimental pilot project using Agile and DevOps Enabled Agile methodologies followed by a systematic comparison of the results in order to justify and validate our proposed methodology for software development.

The development process consists of a three-phase method followed by the developer teams:

1) The first phase consists of the identification of customer requirements before converting them into a sprint backlog.

2) The second phase consists of developing the components of the project and testing them to resolve any issues.

3) The third phase consists of delivering each component to the customer and gaining feedback to keep the project on track and ensure fulfillment of requirements.

We have selected a simple web application for the students of a university to perform course selection and manage payments as our pilot project. In the initialization process, we have formed a team consisting of three members, the developer, the tester and the customer representative.

A use case diagram illustrating the team members, their contributions to the pilot project and the information flow between them is given in Fig. 1:

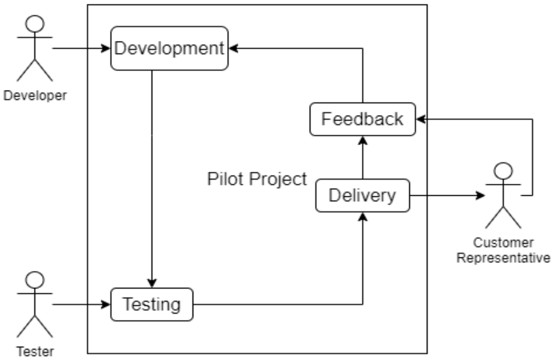

Fig. 1. Use Case Diagram of the Pilot Project.

The customer representative provided a list of requirements for the project, illustrated in Fig. 2:

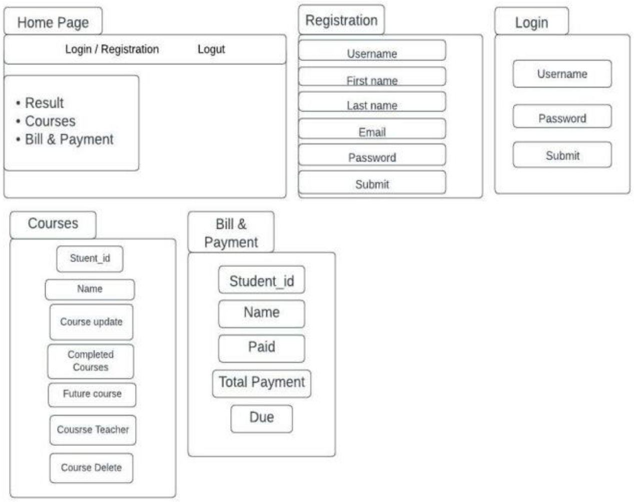

Fig. 2. Requirements Diagram of the Pilot Project.

We have analyzed these requirements describing the structure of the project and converted the results into a sprint backlog consisting of four sprints for each module of the web application with two to three story points denoting module features adding up to a total of 10 story points, as shown in Table I.

TABLE I. SPRINT BACKLOG

Sprint No.	Sprint Description	Story Point No.	Story Point Description
1	Front End	1	Structure of front end system
		2	Design of front end system
		3	Navigation through front end system
2	Authentication	4	Registration feature
		5	Login feature
		6	Log out feature
3	Course Selection	7	Course Selection feature
		8	Course Viewing feature
4	Payment	9	Payment Status feature
		10	Make Payment feature

In the first iteration, we used Agile methodology by assigning the team members to work separately on the development, testing and delivery sectors over a five-day period. We have catalogued the observed data on performance and progression in the form of a series of Burndown Charts plotting actual remaining story points and estimated remaining story points against the number of days passed. The Burndown Charts of this iteration have tracked the performance of the development, testing and delivery sectors using Agile methodology.

For the next iteration we switched to DevOps Enabled Agile methodology with cross-functional team members working together over a five-day period. This led to continuous collaboration and cooperation among the team members, allowing for the development, testing and delivery sectors to become coordinated without having to fulfill the high skill requirements of fully transitioning to DevOps. We have observed the performance of this methodology by generating another series of Burndown Charts tracking the performance of the development, testing and delivery sectors using DevOps Enabled Agile methodology.

Thus, the implementation of our pilot project has been completed using both Agile methodology and DevOps Enabled Agile methodology with the Burndown Charts generated for both iterations allowing for in-depth comparisons of their performances to determine that DevOps Enabled Agile is the superior methodology in terms of project progression. The findings and discussion of our methodology have been presented and further discussed in the next section of our study.

V. FINDINGS AND DISCUSSION

The results of our pilot project have been observed and catalogued in the form of two separate series of Burndown Charts that denote the performance of Agile methodology and DevOps Enabled Agile methodology for the development and delivery of a simple pilot project by a small team of inexperienced developers.

The Burndown Charts for the first iteration (Agile) are given below:

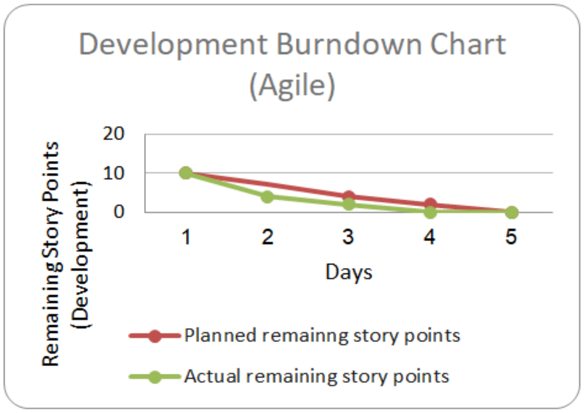

Fig. 3. Development Burndown Chart (Agile).

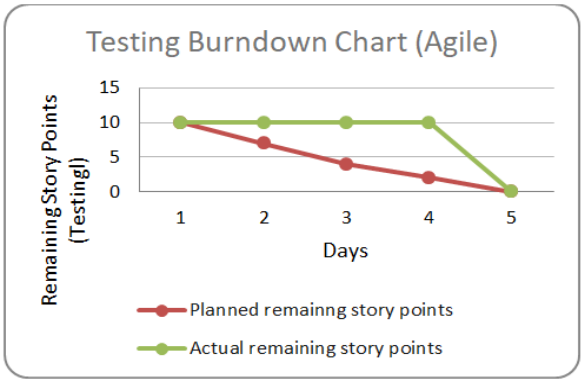

Fig. 4. Testing Burndown Chart (Agile).

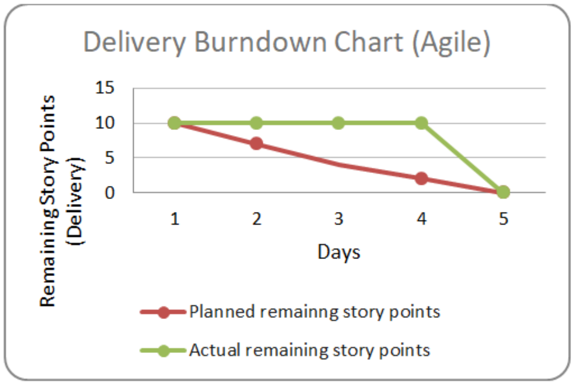

Fig. 5. Delivery Burndown Chart (Agile).

In the first chart of this iteration shown in Fig. 3, we can see that the number of actual remaining story points consistently stayed below the number of estimated remaining story points throughout the project and development reached completion before the estimated deadline on day five, meaning that the development sector performed better than estimated, progressing faster and finishing earlier than planned.

In the second and third charts of this iteration shown in Fig. 4 and Fig. 5 respectively, we can see that the actual remaining story points for testing and delivery have not decreased at all until the last day. This means that testing and delivery sectors were unable to make any progress before the completion of the development sector on day four, which shows that the observed performance is not up to estimated standards.

Therefore, Agile methodology has not performed as well as estimated due to the lack of consistent progression in the testing and delivery sectors. Our proposed reasoning behind this is that the lack of cross-functionality or collaboration between team members lead to the testing and delivery sectors heavily depending on the completion of the development sector, resulting in overall project progression lagging behind planned estimates until the end of the project timeline.

The Burndown Charts for the next iteration of the project using DevOps Enabled Agile are illustrated below:

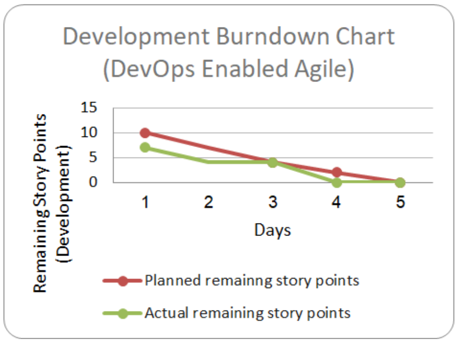

Fig. 6. Development Burndown Chart (DevOps Enabled Agile).

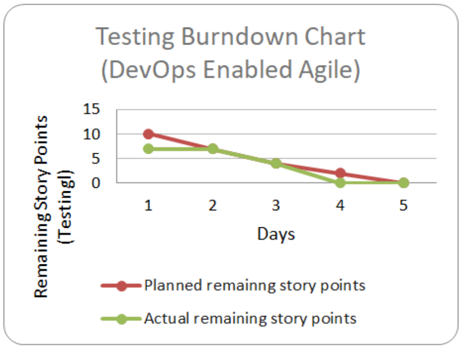

Fig. 7. Testing Burndown Chart (DevOps Enabled Agile).

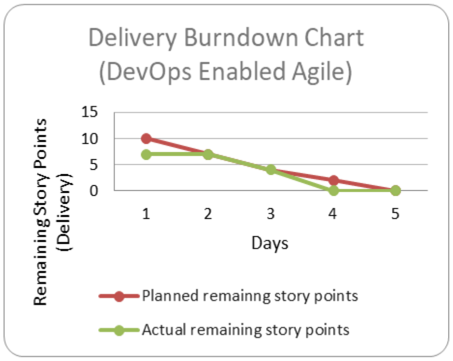

Fig. 8. Delivery Burndown Chart (DevOps Enabled Agile).

In the first chart of this iteration shown in Fig. 6, we can see that the development sector performed better than estimated similar to the previous iteration, showing faster progression throughout the project and reaching completion before the estimated deadline.

In the second and third charts of this iteration which are shown in Fig. 7 and Fig. 8 respectively, we can see that the actual remaining story points for testing and delivery have mostly remained below the estimated remaining story points and reached completion earlier than planned. This means that the testing and delivery sectors have performed better than planned estimates, leading to consistent and efficient progression of project goals and the completion of the testing and delivery sectors before the estimated deadline.

By comparing this data with the data gained from testing Agile methodology in the previous iteration, we can conclude that our proposed methodology, DevOps Enabled Agile methodology has performed far better in terms of consistent project progression and completion of project goals. DevOps Enabled Agile methodology utilized the cross-functionality aspect of DevOps to ensure that all three sectors would progress in a coordinated way leading to all three sectors gaining significant progress throughout the project timeline as opposed to Agile where testing and delivery could not progress until development reached completion. Therefore, DevOps Enabled Agile has been validated as a superior software development methodology when compared to Agile.

VI. Conclusion and Future Work

Our research has been focused on discovering the different gaps of the existing DevOps and Agile Methodologies and combining them to devise a new hybrid methodology, DevOps Enabled Agile, designed to bridge these gaps. From our findings, we have verified the effectiveness and integrity of DevOps Enabled Agile methodology in fulfilling the gaps of Agile methodology while also avoiding the drawbacks of fully transitioning to DevOps methodology and thus we have validated the accomplishment of our research objectives.

The limitation of our research is that we have only tested our proposed DevOps Enabled Agile methodology on a small scale, experimental project. Thus, its effectiveness has not been verified for large scale industrial software development projects. Additionally, our methodology is more focused on the cross-functionality and collaboration principles of DevOps methodology without utilizing any advanced DevOps tools. Application of DevOps Enabled Agile using DevOps tools has been left unexplored as a prospect for future research.

In the near future, we are planning on further optimizing and enhancing DevOps Enabled Agile methodology to make it suitable to be used universally for software development in both small and large scale projects. Additionally, we plan to test the application of DevOps Enabled Agile using different tools and techniques that have not been used in our experiment in order to verify whether this methodology is able to effectively adapt to them. This concludes our current study with a brief description of our future plans for further research.

ACKNOWLEDGMENT

We appreciate the financial support provided by the Institute of Energy, Environment, Research, and Development (IEERD, UAP) and the University of Asia Pacific.

REFERENCES

[1] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland and Dave Thomas, "Manifesto for Agile Software Development", Agile Alliance, 2001.

[2] M. K. Alqudah, R. Razali and M. K. Alqudah, "Agile Methods Selection Model: A Grounded Theory Study" International Journal of Advanced Computer Science and Applications (IJACSA), Volume 10, No. 7, 2019.

[3] Muhammad Asaad Subih, Babur Hayat Malik, Imran Mazhar, Izaz-ul-Hassan, Usman Sabir, Tamoor Wakeel, Wajid Ali, Amina Yousaf, Bilal-bin-Ijaz, Hadiqa Nawaz and Muhammad Suleman, "Comparison of Agile Method and Scrum Method with Software Quality Affecting Factors" International Journal of Advanced Computer Science and Applications(IJACSA), Volume 10, No. 5, 2019.

[4] R. Shankarmani, R. Pawar, S. S. Mantha and V. Babu, "Agile Methodology Adoption: Benefits and Constraints", International Journal of Computer Applications, Volume 58, 2012.

[5] M. A. Bjørni and S. Haugen, "Challenges with Agile in a System Development Department: A Case Study", IEEE Software, Volume 35, Issue 1, 2019. pp. 78 – 83.

[6] F. Erich, C. Amrit, and M. Daneva, "A Mapping Study on Cooperation between Information System Development and Operations", International Conference on Product-Focused Software Process Improvement, 2014. pp. 277–280.

[7] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps", IEEE Software, Volume 33, Issue 3, 2016. pp. 94 – 100.

[8] A. Hemon, B. Lyonnet, F. Rowe and B. Fitzgerald, "Conceptualizing the Transition from Agile to DevOps: A Maturity Model for a Smarter IS Function", IFIP International Federation for Information Processing, 2019. pp. 209–223.

[9] R. T. Yarlagadda, "How Public Sectors Can Adopt the DevOps Practices to Enhance the System", International Journal of Emerging Technologies and Innovative Research, ISSN, 2018. pp. 82-91.

[10] A. Hemon, B. Lyonnet, F. Rowe and B. Fitzgerald, "From Agile to DevOps: Smart Skills and Collaborations", Information Systems Frontiers, Volume 22, 2020.

[11] F. K. Kagai, "Adapting Agile DevOps for Strategic Information Systems Development", TechRxiv Preprint Server, Version 2, 2019.

[12] T. Cardoso, R. Chanin, A. R. Santos and A. Sales, "Combining Agile and DevOps to Improve Students' Tech and Non-tech Skills", CSEDU, 2021. pp. 299-306.

[13] N. Govil, M. Saurakhia, P. Agnihotri, S. Shukla and S. Agarwal, "Analyzing the Behaviour of Applying Agile Methodologies & DevOps Culture in e-Commerce Web Application", 2020 4th International Conference on Trends in Electronics and Informatics, 2020. pp. 899-902.

[14] B. Snyder and B. Curtis, "Using Analytics to Guide Improvement during an Agile–DevOps Transformation", 2017.

[15] G. Y. Koi-Akrofi, J. Koi-Akrofi and H. A. Matey, "Understanding the Characteristics, Benefits and Challenges of Agile it Project Management: A Literature Based Perspective", arXiv e-Print Archive, Volume 10, 2019. pp. 25-44.

[16] A. B. Farid, Y. M. Helmy and M/ M/ Bahloul, "Enhancing Lean Software Development by using Devops Practices" International Journal of Advanced Computer Science and Applications (IJACSA), Volume 8, No. 7, 2017.

[17] L. Banica, M. Radulescu, D. Rosca and A. Hagiu, "Is DevOps another Project Management Methodology?", Informatica Economica, Volume 21, 2017. pp. 39-51.

[18] A. Wiedemann, M. Wiesche, H. Gewald and H. Krcmar, "Understanding How DevOps Aligns Development and Operations: A Tripartite Model of Intra-IT Alignment", European Journal of Information Systems, 2020. pp. 1-16.

[19] K. Kuusinen, V. Balakumar, S. C. Jepsen, S. H. Larsen, T. A. Lemqvist, A. Muric, A. Ø. Nielsen, and O. Vestergaard, "A Large Agile Organization on its Journey towards DevOps", 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2018. pp. 60-63.

[20] S. M. Mohammad, "DevOps automation and Agile Methodology", International Journal of Creative Research Thoughts (IJCRT), Volume 5, Issue 3, 2017. pp. 946-949.