





A DevOps Implementation Framework for Large Agile-Based Financial Organizations

Anitha Devi Nagarajan  and Sietse J. Overbeek ^(✉) 

Department of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands

{A.D.Nagarajan2, S.J.Overbeek}@uu.nl

Abstract. Modern large-scale financial organizations show an interest in embracing a DevOps way of working in addition to Agile adoption. Implementing DevOps next to Agile enhances certain Agile practices while extending other practices. Although there are quite some DevOps maturity models available in the literature, they are either not specific to large-scale financial organizations or do not include the Agile aspects within the desired scope. This study has been performed to identify why such organizations are interested in implementing DevOps and how this implementation can be guided by a conceptual framework. As a result, a list of drivers, a generic DevOps implementation framework and driver-dependent variations are presented. The development of these artifacts has been realized through a design science research method and they have been validated by practitioners from financial organizations in the Netherlands. The practitioners have identified the developed artifacts as useful, mainly to educate people within their organizations. Moreover, the artifacts have been applied to real organizational goals to demonstrate how they can be of help to identify the useful measurement units, which in turn can help to measure and achieve their DevOps transformation goals. Thus, the developed artifacts are not only serving as a baseline for future research but are also useful for existing financial organizations to commence and get ahead with their DevOps implementations.

Keywords: Agile · DevOps implementation framework
DevOps drivers · Large financial organizations
Transformation measurement

1 Introduction

Agile methodologies have gained a widespread acceptance due to advantages like faster software development with improved quality, and the ability to welcome changes throughout the project leading to improved customer satisfaction in comparison with traditional software development approaches such as the waterfall method [2] and the incremental method [34]. However, the structural

division between the functional departments such as development and operations remained, and it leads to delays in the deployment of the developed software in the actual production environment [9, 35]. As a solution to the latter, the DevOps movement has emerged with the purpose of closing the gap between development and operations [38, 40].

Although DevOps has emerged from cloud-based product organizations, the DevOps paradigm is not exclusive to organizations that are surrounded with cloud computing [38, 40]. Due to the DevOps adoption benefits reported by several other organizations such as better quality assurance and enhanced collaboration and communication [31], large financial organizations are also willing to embrace this new way of working. However, the objectives behind a DevOps implementation may differ considerably among different types of organizations and so does the corresponding means to measure their success [10].

DevOps is often coalesced with Agile principles and practices. Those studies that compared DevOps with other software development methodologies have identified that both Agile and DevOps have similar goals and values, but their scope varies. When DevOps is ‘laid over’ an Agile implementation, it enhances several Agile practices while extending others outside development activities [19, 20, 23]. There are several studies that list the factors required for a successful Agile transformation from various perspectives [4, 5]. The existing DevOps implementation models have focused only on DevOps and so little attention has been paid on its relationship with Agile principles and practices [24].

Motivated by these concerns and encouraged by the current needs of participating organizations, it was intended to develop a conceptual framework, which depicts the various aspects involved in the DevOps implementation of Agile-based financial organizations. We wanted to connect it to their drivers and the measurement units to make it more complete since the progress towards the DevOps transformation goals are measured by measurement units, which in turn help to get closer to the goals. This paper is organized as follows: Sect. 2 describes the research approach and the involved methods and techniques. Section 3 elaborates the developed artifacts and subsequently Sect. 4 discusses the evaluation of those artifacts. Finally, Sects. 5 and 6 reviews and concludes the study respectively.

2 Research Approach: The Design Cycle

This study design is inspired from the design science methodology by Wieringa [39] and so a brief overview of this study’s approach is shown in Fig. 1. Each of the phases mentioned in the figure is explained further.

Problem Investigation and Data Collection. First, the research objectives and research questions were formulated by understanding the interests and needs of the involved stakeholders, one of the participating financial organizations in the Netherlands and also by identifying a gap in the available literature in this context. Next to that, the literature study was carried out to review other relevant scientific studies to lay a stable theoretical foundation and to gather data

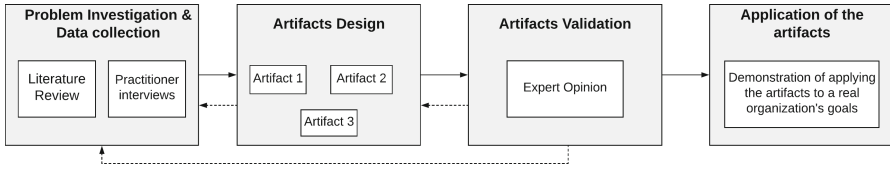


Fig. 1. Research approach inspired by a design science framework [39]

from them. Finally, semi-structured interviews were conducted with practitioners from different organizations in order to collect industry-specific data. Furthermore, the requirements about the prospective artifacts were also collected from the stakeholders.

Artifacts Design. In the second phase, the following results were established: (1) Drivers for Agile and DevOps implementation at the large financial organizations, (2) Generic DevOps implementation framework, and (3) Driver-dependent framework variations based on the relationship identified between the developed framework and the identified drivers. These results were realized by performing the directed content analysis on the data collected from the literature and practitioner interviews. In order to perform the content analysis, we used the tool named Nvivo. The resulted artifacts from this phase are elaborated in Sect. 3.

Artifacts Validation. After identifying the mentioned results and documenting them as artifacts, they were iteratively validated with industrial experts by means of the expert opinion method [39]. The validation session results are detailed in Sect. 5. Based on the validation session outcome, the artifacts were refined and their updated versions were used for further validation sessions. These refinements included a missing driver and an incomplete focus area.

Application of the Artifacts. It is demonstrated how the developed artifacts can be applied to an organization's goals to identify the possible and useful measurement units, which in turn can help to measure their progress towards their DevOps implementation goals. For this, we have considered certain goals of a large Dutch bank that participated in the research and performed the demonstration. The purpose of this demonstration is to show how the developed framework can uncover the various possible obstacles towards the organizational goals and how to select the suitable measurement units based on that.

3 Solution Design : Development of the DevOps Implementation Framework

This chapter is presenting the results of the data analysis performed on the data collected from both practitioners and the available literature. Based on the results, the basic components namely, drivers, perspectives and focus areas are identified. Later by grouping the discerned perspectives and focus areas, a DevOps implementation framework is developed. Thereafter the relationship between the developed framework and the identified drivers are revealed.

3.1 Agile and DevOps Implementation Drivers at Financial Organizations

The list of drivers provide the reasons why large financial organizations are interested in implementing Agile and DevOps. There are six such drivers identified namely, (1) agility and customer-centricity, (2) efficient value delivery to customers, (3) cooperative culture, (4) empowered people, (5) focus on continuous improvement and, (6) process and stakeholder alignment as shown in Fig. 2. These drivers have been identified based on the data collected from the practitioners and their importance has been legitimized by cross-checking with the literature.

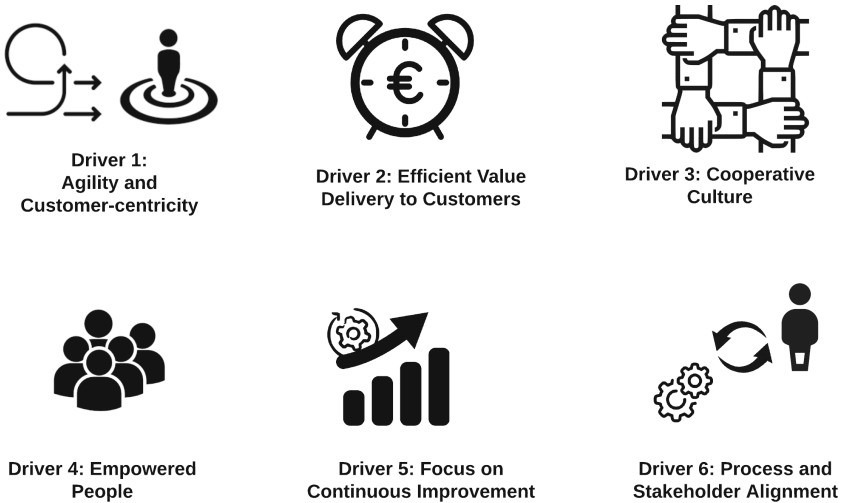


Fig. 2. DevOps and Agile implementation drivers for large financial organizations

Agility and Customer-Centricity. Agility is an ability of organizations to respond faster to changes as such from customer and market [36]. Customer-centricity is the ability of the organizations to develop systems according to customer preferences and needs [21]. The need for these abilities are found to be the main drivers for large financial organizations to embrace DevOps and Agile.

Efficient Value Delivery to Customers. Agile has proven to be increasing the speed of the upstream processes of software development such as, identifying business needs and developing software accordingly. On the other hand, the speed of downstream processes like verification, validation and delivery of the software can be improved with the support of DevOps [15].

Cooperative Culture. Thanks to the Agile software development process, the wall between the customers and development team was brought down as a consequence of the frequent communication possibilities and smaller iterations [27].

DevOps has enhanced this scenario further by allowing all possible roles within system development, operations and maintenance to work closely with each other, which has largely enriched the communication among the involved IT stakeholders [20,31].

Empowered People. Organizations want their people to be more empowered by taking ownership on their tasks and be capable of doing more than what is described in their job descriptions. People are expected to focus on achieving the group goal instead of focusing only on their individual achievements.

Focus on Continuous Improvement. Traditionally, organizations were interested mostly in the improvement of their delivery. However, Agile promoted the incremental software development and delivery (Principle 3) which allowed teams to learn from their past and to become better progressively (Principle 12) [13]. DevOps drives organizations to concentrate not only on the delivery part but also on the improvement of people and processes [19].

Process and Stakeholder Alignment. The alignment among various stakeholders such as business teams, IT teams and end users is identified as the last important driver for Agile and DevOps implementations of the target organizations. Such an alignment requires collaboration and communication among IT teams and customers, and between organizational units themselves.

3.2 DevOps Implementation Framework for Agile-Based Large Financial Organizations

The developed generic DevOps implementation framework suitable for Agile-based financial organizations is shown in Fig. 3. This framework has been developed to serve as a guideline to all types of employees involved in a DevOps implementation of such organizations. This framework is suitable to be used for those that have already implemented Agile or that are interested in implementing Agile along with a DevOps implementation. The framework has been developed with two levels of constructs namely, perspectives and focus areas.

Perspectives are the dimensions that the corresponding implementation factors belong to and there are four of them namely, (1) organizational perspective, (2) people perspective, (3) process perspective and (4) technology perspective. These perspectives are identified from the agile software development literature and they are maintained here since the interviewees agreed that these perspectives are appropriate to this context.

The focus areas are the principal sections that require attention within every perspective regarding the Agile and DevOps adoption. These focus areas are patterns identified mainly from the interviews and they are further explained with the corresponding literature studies. Every focus area is related to one of the given perspectives.

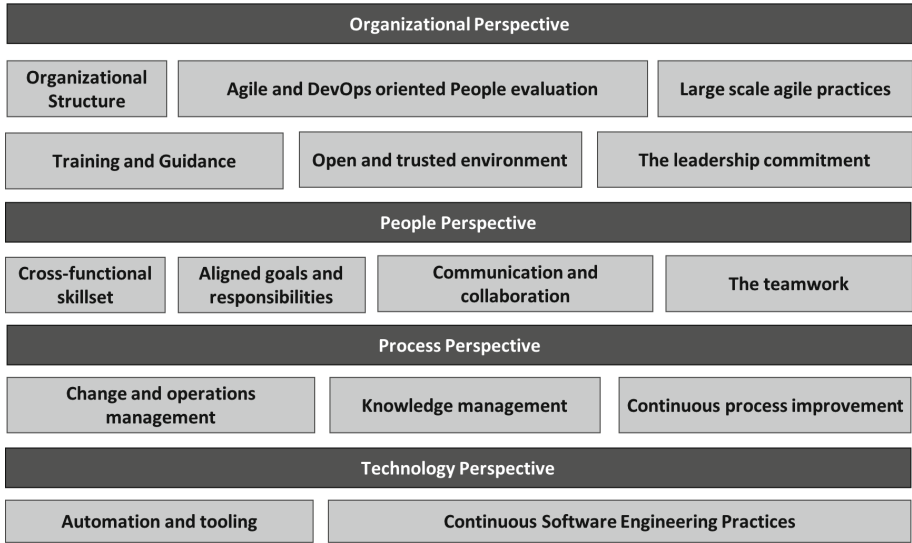


Fig. 3. DevOps implementation framework for Agile-based large financial organizations

Organizational Perspective. The organizational perspective includes all the focus areas that the management of a typical non-DevOps organization should consider and facilitate in order to constitute the landscape within their organization to foster a DevOps mentality. The following focus areas are identified as the significant ones within this perspective.

(Sub-)Organizational Structure. The first focus area is about the structure of the organization and the sub-organizations. DevOps leads to teams that bring together experts such as software development professionals and operations professionals enabling them to share their skills and experiences [19]. The team structure should allow for live and peer-to-peer communication within the team but not via other means such as through management or tickets [32, 33].

Agile and DevOps Oriented People Evaluation. The next one is regarding the people evaluation and performance reviews that are commonly conducted within an organization. It is imperative for the organization to make sure that the method of evaluation is team-based, encourages collaboration over competition among team members and teams, and is not conflicting with the behavioral needs of Agile and DevOps [6].

Large-Scale Agile Practices. This focus area emphasizes on tailoring the agile practices specific to the organization and following them throughout the organization, not only at the team levels but also at the project and portfolio levels across the enterprise [17].

Open and Trusted Environment. Having an open and transparent environment is an important characteristic not only for the teams but for the entire

DevOps enterprise. Therefore, the management should be clearly communicating the goals and objectives of the decisions that involve teams, and keep the metrics visible for everyone so that they can share the responsibility to achieve it together [6]. Moreover, the organization should give a safe environment for people to give their honest opinion and feedback without being afraid of fear or abuse [33].

Training and Guidance. The human impediments towards organizational change such as lack of knowledge, cultural issues, resistance to change, wrong mindset and lack of collaboration can be handled by coaching and guiding them properly and by stimulating their growth mindset. This is possible with the help of training and human facilitators such as coaches and champions [14, 25].

The Leadership Commitment. The leaders in the Agile and DevOps environment should not support but also practice the agile methods to perform their leadership activities wherever applicable. Moreover, the managers in such environment should practice ‘leadership and collaboration’ but not ‘command and control’. The Agile leaders provide guidance, take risks, should be committed to their people, and collaborate with various levels of stakeholders [30].

People Perspective. This perspective identifies the most important people characteristics required for the effective working at Agile and DevOps based organizational environments.

Cross-Functional Skillset. The cross-functional teams are an important component of DevOps environment which in general is formed initially by involving experts from various functional domains such as programmers, functional testers, performance testers and operations personnel. Ideally, this can lead to a situation in which these experts communicate and collaborate to become cross-functional team members who are multi-skilled and flexible [1, 7, 19].

Aligned Goals and Responsibilities. Hutterman defines a team as a group of people working together to achieve a shared group goal [19]. Within a DevOps organization, the team goals should not conflict with each other but focus on achieving a common goal that is beneficial to an user group. Based on our understanding of the collected data, we say that the sub-organizational goal should be aligned to the main goals of the enterprise and in the same way, an (agile) team’s goals should be in line with the corresponding sub-organizational goals. Thus the people goals and their responsibilities should be driven by the shared team goals which are associated with them. Figure 4 depicts this.

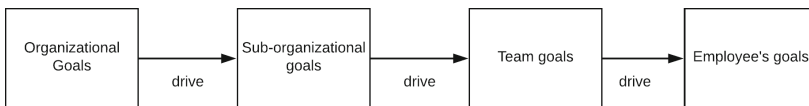


Fig. 4. Emphasizing aligned goals within DevOps environment

Communication and Collaboration. The next focus area accentuates the need for effective communication and intense collaboration among the team members, IT management and business. With the act of communication, people do exchange knowledge, influence each other, recognize each other's work and build a community. By working collaboratively within the community, people build trust and empathy for each other [6].

The Teamwork. The next focus area draws attention to the teamwork aspect of people working in Agile and DevOps organizations. Teamwork boosts not only the performance of the team but also the individual performance and together it contributes to the overall performance of the organization [22].

Process Perspective. This perspective includes the important process areas that the agile organizations need to consider within the context of DevOps.

Change and Operations Management. This focus area insists on developing a change and operations management plan and integrating it with the project management method. DevOps practices intend to reduce the time between the code commits of a change in the development system and placing the change in the production system [31]. Involving the operations group in the Change Advisory Board and by coordinating with the operations maintenance personnel will help to make sure current operations will not be negatively impacted [26].

Knowledge Management. Thanks to Agile and also DevOps, the functional groups of people are disseminated and restructured into cross-functional DevOps teams, which are formed around value streams. This brings in a clear need for effective knowledge management processes and activities so that continuous learning and coordination can happen within the enterprise. The organization and the people need to identify suitable knowledge management processes that work for them and support them with relevant tools and infrastructure. People should be aware of the advantages and importance of knowledge management practices and so are encouraged to share knowledge with each other.

Continuous Process Improvement. The Agile and DevOps adoption by large complex organizations require experimentation and adaptation of the methods and processes to the organization's structure, culture, product/service strategy, human resource management policies, customer interfaces, project roles and governance structures, including program and project portfolio management [17].

Technology Perspective. This perspective identifies and describes the focus areas which require attention from the technological standpoint within the DevOps implementation.

Automation and Tooling. According to several studies, automation is found to be the technological enabler of DevOps [19,31,40]. Being aware of both the

benefits and possible pitfalls, organizations should perform effective automation so that it can be an advantage but not an impediment.

Continuous Software Engineering Practices. Continuous Software Engineering practices help to eliminate waste in the context of lean software development. Some examples of waste are, (a) delays, due to lack of communication and understanding; (b) unnecessary additional work that does not yield expected business value; (c) defects due to poor execution of tasks; (d) partial completion of work. These wastes can possibly be removed or reduced with the implementation of continuous software engineering practices such as continuous integration, continuous testing, continuous monitoring, continuous delivery and other such practices [12]. Because of their contribution to the faster value delivery, they are here involved in the context of DevOps implementation.

3.3 Relationship Matrix Between Drivers and Focus Areas of the Developed Framework

The final research outcome is developed to understand which among the fifteen focus areas should be first aimed at, based on what has driven an organization to go with a DevOps implementation. Based on these relationships, we have developed the variations of the presented framework for every driver, which highlights the related focus areas from Table 1. However, they are not shown here due to the space restrictions. Also the rationale behind the given relationships are not described here for the same reason but, those justifications are either based on literature or from the collected interview data, or both. For example, the second focus area ‘Agile and DevOps oriented people evaluation’ can influence the cooperative work culture (driver 3) if the reward structure is utilized appropriately [3]; the empowered people (driver 4), since people’s self-development is encouraged and self-confidence is improved when the feedback is constructive [19, 29]; the focus on continuous improvement (driver 5), because rewarding the whole team can encourage them to achieve more together [37] and giving them regular feedback help them to refine themselves progressively [19]. However, this focus area’s direct influence on other drivers were not found from the collected data sources.

From Table 1, we infer that the focus areas namely, the leadership commitment from organizational perspective, aligned goals and responsibilities, team work from the people perspective and the continuous software engineering practices from the technology perspective are significant for all identified DevOps drivers. In addition to that, from Table 1 it can be deduced that the people perspective is the most contributing one in the case of a DevOps transformation as this is influencing most of the drivers and the DevOps implementation goals.

Table 1. Relationship matrix between the identified drivers and the focus areas

Focus Areas Drivers	Driver 1: Agility and customer-centricity	Driver 2: Efficient value delivery to customers	Driver 3: Cooperative culture	Driver 4: Empowered people	Driver 5: Focus on continuous improvement	Driver 6: Process and stakeholder alignment
Focus area 1: (Sub-)organizational structure	X	X	X	X		X
Focus area 2: Agile and DevOps oriented people evaluation			X	X	X	
Focus area 3: Large-scale agile practices	X	X	X			X
Focus area 4: Open and trusted environment			X	X		
Focus area 5: Training and guidance			X	X	X	
Focus area 6: The leadership commitment	X	X	X	X	X	X
Focus area 7: Cross-functional skillset	X	X	X	X		X
Focus area 8: Aligned goals and responsibilities	X	X	X	X	X	X
Focus area 9: Communication and collaboration	X	X	X		X	X
Focus area 10: Teamwork	X	X	X	X	X	X
Focus area 11: Change and operations management	X	X				X
Focus area 12: Knowledge management		X	X		X	X
Focus area 13: Continuous process improvement			X	X	X	X
Focus area 14: Automation and tooling	X	X			X	X
Focus area 15: Continuous software engineering practices	X	X	X	X	X	X

4 Application of Drivers and Framework to Identify Measurement Units

This section demonstrates how the developed framework and identified drivers can be used to achieve the DevOps transformation goals of an organization with the help of an example. For this, we have considered one of the goals of a Dutch financial organization, which is a multinational banking and financial services

company and it is one of the largest banks in the Netherlands. Their goal is about expediting their delivery so that their customers can enjoy their service and products earlier than before. Since this goal is suitable to be analyzed from the perspectives of both people and process, we have chosen to present it here.

As mentioned above, in order to explain the derivation of suitable measurement units, the goal has been analyzed in two ways: (a) from the people and organizational perspective and (b) from the process and technology perspective as shown in Fig. 5. In order to improve the time taken for delivery, it is important to first know how much time it currently takes for any requirement including new feature related requirements and change requests. Thus it is relevant to measure the **(1) time passing between the initiation and the actual delivery of those requirements**. However, it might not really be enough to keep looking at the overall time that is being taken for the life-cycle of a requirement when the time stays indifferent. In that case, we can have a deeper look into the time by checking it in two different ways, which makes the given Fig. 5 to get separate branches into people and process perspectives.

From the people perspective, the requirements can be checked to see **(2) the time period that a requirement stayed with different roles such as tester or operations**. It is useful to link the organizational perspective with the people perspective here. Based on the identified time taken by different roles to handle requirements, the following questions may arise:

1. Why does a specific role keep these requirements longer?
2. What is the average time spent by that role on other requirements?
3. What can be done to reduce the time spent by that role?
4. Is this a common scenario with anyone taking that role or is it something specific to the person who took that role?

The above questions are formulated to get a deeper understanding on the source of the problem (i.e., longer processing time of requirements) based on the identified perspectives. For example, the above questions may help to reveal the existing issues such as communication issues among roles, specific role's inability in taking up other role's tasks, management interference or less commitment of the involved people. One or more of these issues may be identified as the obstacle towards achieving the goal and so they need to be paid attention to.

Similarly, from the process and technology perspective, the requirements can be checked to see **(3) the time periods that a requirement stayed with involved processes such as development or functional testing**. This helps to identify which process takes longer which in turn initiates an analysis, such as:

1. Why that specific process takes longer than others for a requirement?
2. What is the average time spent on that process for other requirements?
3. How can that process be improved to reduce time?
4. Is the improvement required on the identified process or any other dependent process?
5. Is it really the process that needs improvement or the people who are involved in it?

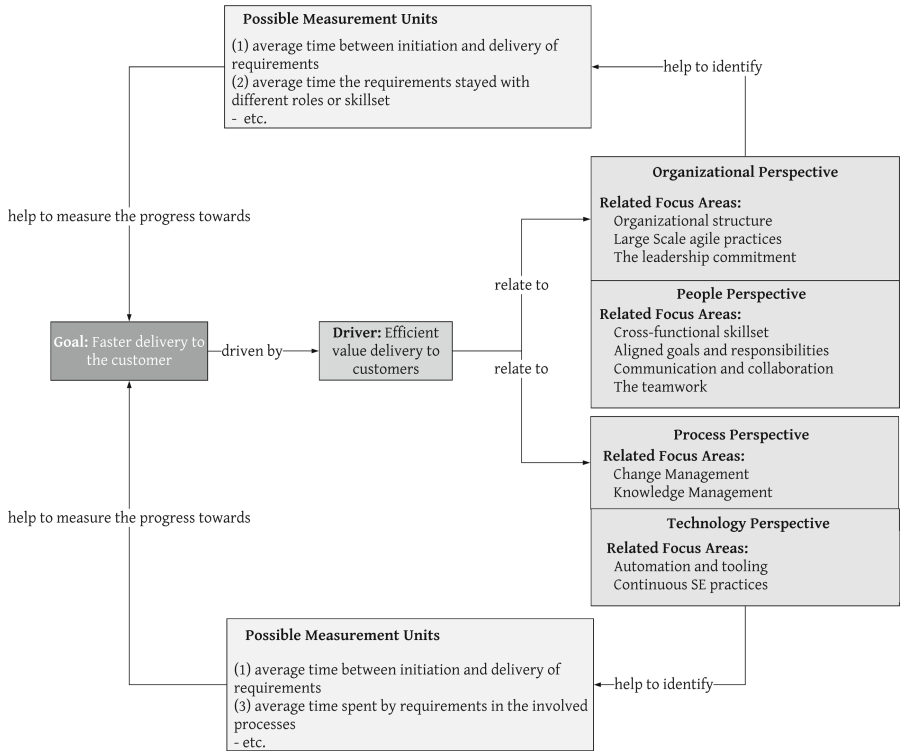


Fig. 5. Application of drivers and the developed framework on an organizational goal

As explained, asking such relevant questions helps to identify where the obstacle is and how that obstacle can be removed. The more important note is that these analyses should always lead to the identification of metrics that provoke the discussion of improvement points in terms of people, process or technology but not blaming each other.

Knowing the relevant focus areas which are related to the corresponding driver helps to ask the relevant questions. Moreover, they can be of help to go on and check the next relevant focus area from different perspectives so that the obstacles indirectly related to the goal can be identified and so the measurement units can be adjusted to measure the right focal point that needs attention.

5 Validation of the Artifacts

The expert opinion sessions were conducted with five experts who have various levels of experience working at financial organizations to evaluate the mentioned artifacts. These experts fulfilled the roles of DevOps engineer, DevOps consultant, DevOps architect and Delivery Manager in their respective organizations. All the experts have been part of one or more DevOps implementations within

their organizations or other organizations for which they have given consultations. During the validation session, the evaluators have been presented with the results one after the other and they have been asked the criteria-based questions related to it. Moreover, they have been allowed to go through the printed documentation in order to get more details whenever required. For the evaluation, we have considered several criteria namely, completeness, fit with organization, understandability, usefulness and accuracy. These criteria have been identified from the hierarchy of IS artifact evaluation criteria developed by Prat [28]. The evaluation results of the drivers, framework and their relationship can be found in Table 2.

Table 2. Evaluation results of artifacts

Criterion	Session 1		Session 2		Session 3
	Evaluator 1	Evaluator 2	Evaluator 3	Evaluator 4	Evaluator 5
Drivers					
Completeness	++	++	++	++	++
Fit with organization	+ -	+ -	++	++	++
Understandability	++	++	++	++	++
DevOps implementation framework for large financial organizations					
Completeness	++	++	++	+ -	++
Fit with organization	++	++	++	+ -	++
Understandability	++	++	++	++	++
Usefulness	++	++	++	+ -	++
Relationship drawn between artifacts 1 and 2					
Accuracy	N/A	N/A	++	++	++
Usefulness	+ -	++	++	+ -	++

++ Fully Agreed +- Partially Agreed - - Rejected N/A Not Assessed

According to the validation results, the evaluators agreed that the identified list of drivers is complete and the developed DevOps implementation framework includes all the required perspectives and focus areas. They confirmed that the results are easy to understand and most of them agreed that they are suitable to their organization. The evaluators agreed that the developed framework is suitable for the organizations who are yet to implement DevOps or those who are at the initial and immature stages of DevOps implementation. On the other hand, they mentioned that the framework may not help for those who are already mature with their DevOps implementations.

As it can be noted with the given results in Table 2, none of our results have been completely rejected by the evaluators. It could be because of one of the limitations of the study i.e. both data collection and evaluation was performed by companies based in one country (Netherlands) and the number of participated companies is limited to three.

6 Discussion

The current study has developed a high-level framework that encompasses the various perspectives and the focus areas that are relevant for the successful DevOps implementation of a large-scale organization. To maximize the usefulness of the framework and to make it specific for financial organizations, from where the practitioners are selected to participate in this study, we have also identified the drivers and we have demonstrated the derivation of measurement units based on the DevOps implementation goals.

Implications. In this research, several factors in terms of perspectives and focus areas as part of a DevOps implementation have been introduced. As can be seen in Table 1, the identified focus areas have many-to-many relationships with the collected drivers. Overall, a DevOps implementation is a collective effort of people working at different levels within an organization. For a successful implementation, an organization should discover which areas of the organization need what kind of changes and how to proceed from there. The developed conceptual framework is beneficial to realize such an implementation. In short, an exemplary DevOps organization underlines the need for people development and for process improvement. Moreover, it has a culture in which competition is of less importance compared to the importance of learning.

Comparison with Related Studies. There are several studies which identified the success factors of an Agile implementation from different perspectives [4, 5, 11]. The current study is also inspired on those studies and followed the list of perspectives taken from those studies. However, this one is different from them since the other mentioned studies concentrated on Agile whereas, the current one has concentrated on a DevOps implementation where Agile is also followed. This study expects the involved organization to already follow Agile or to implement Agile together with a DevOps implementation. Next to that, there are some DevOps maturity models available in the literature [8, 24]. This study is different from those studies in the following aspects: the current study focused on large organizations and is specific for the finance industry; the current study has the possibility to be expanded to an ‘organization specific maturity model’ in which every focus area is defined with the list of capabilities that the involving organization wants to progressively reach. This can be achieved by analyzing the organization’s situation and identifying the specific capabilities based on where they are and what areas they want to reach with DevOps. This process of developing an organization-specific maturity model using the developed conceptual framework is comparable to Situational Method Engineering [16]. On the contrary, the other mentioned studies focused on developing a generic maturity model which may not be suited to every organization and also they may not be suitable for tailoring.

Limitations. The current study considered several sources of data which came from both practitioners and other scientific studies. Although the participated practitioners are originally from various geographical areas, they all currently belong to a few financial organizations in the Netherlands. Moreover, the

evaluation part is performed by means of expert opinion, which focused on evaluating the artifacts against the given criteria. These evaluation results are not enough to quantitatively prove the usefulness of the developed artifacts in a real DevOps implementation scenario.

7 Conclusions and Future Research

As like with any other industry, DevOps is becoming popular among financial software organizations. Because of the advantages observed with Agile software development methods such as faster development time, improved quality and high customer-satisfaction, several large-scale financial organizations prefer Agile methods over traditional software development methods like waterfall software development method. However, before the start of this study it was still not clear why they are interested in implementing DevOps along with or on-top of an Agile implementation. Thus, this paper concentrated on identifying the drivers for large-scale financial organizations to 'go for' DevOps along with an Agile software development method. Nevertheless, with a DevOps adoption, several existing factors get affected and many other new factors need attention. Thus, the current study brings up a framework based on high-level factors from different perspectives that are required for the DevOps implementation in such organizations and develops the variations of the framework based on its relationship with the identified drivers. To justify the usage of the developed DevOps implementation framework along with identified drivers, an application scenario with a real financial organization's goal has been presented.

This study provides quite some future research opportunities. Since the data for the current study were collected mostly from banks in the Netherlands, the future studies can concentrate on performing similar research by taking other financial institutions into account, such as insurance companies and possibly organizations from various geographical locations. Subsequently, comparing the current study with studies in those different but comparable domains may even bring interesting results. As suggested by one of the evaluators, a useful note for similar research is to develop more specific focus areas that reduce the overlap between the driver-dependent variations of the developed framework. Furthermore, the current study has established the relationship between drivers and the focus areas and it was mostly based on the theoretical data found from the available literature. Every driver - focus area pair can be empirically researched to identify the concrete relationships between them. Since the developed conceptual framework has not been applied to a full fledged DevOps implementation, the framework itself can be revised and improved after being utilized in its entirety.

References

1. Abidin, Z.F.A., Jawawi, D., Ghani, I.: Agile transition model based on human factors. *Int. J. Innov. Comput.* **7**(1), 23–32 (2017)
2. Bhadoriya, N., Mishra, N., Malviya, A.: Agile software development methods, comparison with traditional methods and implementation in software firm. *Int. J. Eng. Res. Technol.* **3**(7) (2014)

3. Chatman, J.A., Barsade, S.G.: Personality, organizational culture, and cooperation: evidence from a business simulation. *Adm. Sci. Q.* **40**, 423–443 (1995)
4. Chow, T., Cao, D.B.: A survey study of critical success factors in agile software projects. *J. Syst. Softw.* **81**(6), 961–971 (2008)
5. Darwish, N.R., Rizk, N.M. : Multi-dimensional success factors of agile software development projects. *Int. J. Comput. Appl.* **118**(15) (2015)
6. Davis, J., Daniels, R.: *Effective DevOps: building a culture of collaboration, affinity, and tooling at scale.* O'Reilly Media Inc., Sebastopol (2016)
7. Demirkan, H., Spohrer, J.: T-shaped innovators: identifying the right talent to support service innovation. *Res.-Technol. Manag.* **58**(5), 12–15 (2015)
8. de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., Brinkkemper, S.: DevOps competences and maturity for software producing organizations. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) *BPMDS/EMMSAD -2018. LNBIP*, vol. 318, pp. 244–259. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_16
9. de FranSa, B.B.N., Jeronimo Jr., H., Travassos, G.H.: Characterizing DevOps by hearing multiple voices. In: *Proceedings of the 30th Brazilian Symposium on Software Engineering*, pp. 53–62 (2016)
10. Elberzhager, F., Arif, T., Naab, M., Süß, I., Koban, S.: From agile development to DevOps: going towards faster releases at high quality – experiences from an industrial context. In: Winkler, D., Biffi, S., Bergsmann, J. (eds.) *SWQD 2017. LNBIP*, vol. 269, pp. 33–44. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49421-0_3
11. El Hameed, T.A., Latif, M.A.E., Kholief, S.: Identify and classify critical success factor of agile software development methodology using mind map. *Int. J. Adv. Comput. Sci. Appl.* **7**(5), 85–92 (2016)
12. Farid, A.B., Helmy, Y.M., Bahloul, M.M.: Enhancing lean software development by using DevOps practices. *Int. J. Adv. Comput. Sci. Appl.* **8**(7), 267–277 (2017)
13. Fowler, M., Highsmith, J.: The agile manifesto. *Softw. Dev.* **9**(8), 28–35 (2001)
14. Gandomani, T.J., Zulzalil, H., Ghani, A.A.A., Sultan, A.B.M., Nafchi, M.Z.: Obstacles in moving to agile software development methods; at a glance. *J. Comput. Sci.* **9**(5), 620–625 (2013)
15. Geurts, W.J.: Faster is better and cheaper. In: *INCOSE International Symposium*, vol. 26, no. 1, pp. 1002–1015 (2016)
16. Harmsen, A.F., Brinkkemper, J.N., Oei, J.H.: Situational method engineering for information system project approaches, pp. 169–194, Department of Computer Science, University of Twente, pp. 1–32 (1994)
17. Hobbs, B., Petit, Y.: Agile methods on large projects in large organizations. *Proj. Manag. J.* **48**(3), 3–19 (2017)
18. Horney, N., Pasmore, B., O'Shea, T.: Leadership agility: a business imperative for a VUCA world. *Hum. Resour. Plan.* **33**(4), 32–38 (2010)
19. Huttermann, M.: *DevOps for Developers.* Apress, New York City (2012)
20. Jabbari, R., bin Ali, N., Petersen, K., Tanveer, B.: What is DevOps?: A systematic mapping study on definitions and practices. In: *Proceedings of the Scientific Workshop Proceedings of XP2016*, p. 12. ACM (2016)
21. Liang, T.P., Tanniru, M.: Special section: customer-centric information systems. *J. Manag. Inf. Syst.* **23**(3), 9–15 (2006)
22. Lindsjörn, Y., Sjöberg, D.I., Dingsøy, T., Bergersen, G.R., Dybå, T.: Teamwork quality and project success in software development: a survey of agile development teams. *J. Syst. Softw.* **122**, 274–286 (2016)

23. Lwakatare, L.E., Kuvaja, P., Oivo, M.: Relationship of DevOps to agile, lean and continuous deployment. In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T. (eds.) PROFES 2016. LNCS, vol. 10027, pp. 399–415. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49094-6_27
24. Mohamed, S.: DevOps maturity calculator DOMC-value oriented approach. *Int. J. Eng. Sci. Res.* **2**(2), 25–35 (2016)
25. Parizi, R.M., Gandomani, T.J., Nafchi, M.Z.: Hidden facilitators of agile transition: agile coaches and agile champions. In: 8th Malaysian Software Engineering Conference (MySEC 2014), pp. 246–250. IEEE (2014)
26. Phifer, B.: Next-generation process integration: CMMI and ITIL do devops. *Cut. IT J.* **24**(8), 28 (2011)
27. Pikkarainen, M., Haikara, J., Salo, O.: The impact of agile practices on communication in software development. *Empir. Softw. Eng* **13**(3), 303–337 (2008)
28. Prat, N., Comyn-Wattiau, I., Akoka, J.: Artifact evaluation in information systems design-science research-a holistic view. In: PACIS, pp. 23–39 (2014)
29. Quinn, R.E., Spreitzer, G.M.: Seven questions every leader should consider. *Organ. Dyn.* **26**(2), 37–49 (1997)
30. Rigby, D.K., Sutherland, J., Takeuchi, H.: Embracing agile. *Harv. Bus. Rev.* **94**(5), 40–50 (2016)
31. Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L.E., Tiihonen, J., Männistö, T.: DevOps adoption benefits and challenges in practice: a case study. In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T. (eds.) PROFES 2016. LNCS, vol. 10027, pp. 590–597. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49094-6_44
32. Sharma, S.: *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise*. Wiley, Hoboken (2017)
33. Swartout, P.: *Continuous Delivery and DevOps: A Quickstart Guide*. Packt Publishing Ltd., Birmingham (2014)
34. Tarhan, A., Yilmaz, S.G.: Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process. *Inf. Softw. Technol.* **56**(5), 477–494 (2014)
35. Tessem, B., Iden, J.: Cooperation between developers and operations in software engineering projects. In: Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering, pp. 105–108 (2008)
36. Tseng, Y.H., Lin, C.T.: Enhancing enterprise agility by deploying agile drivers, capabilities and providers. *Inf. Sci.* **181**(17), 3693–3708 (2011)
37. Walls, M.: *Building a DevOps Culture*, 1st edn. O’Reilly Media Inc., Sebastopol (2013)
38. Wettinger, J., Breitenbcher, U., Leymann, F.: Standards-based DevOps automation and integration using TOSCA. In: Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 59–68 (2014)
39. Wieringa, R.J.: *Design Science Methodology for Information Systems and Software Engineering*. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-662-43839-8>
40. Zhu, L., Bass, L., Champlin-Scharff, G.: Devops and its practices. *IEEE Softw.* **33**(3), 32–34 (2016)