Working with Agile and DevOps Together

Linnea Hagman

May 2022

1 Introduction

Since the 1960s, different methodologies and orientations have evolved in the software development industry [1]. The once popular linear Waterfall methodology divides the work into phases where each phase needs to be completed before pursuing the next. The methodology received critique for not being effective when unexpected changes occur. Extra time and money are needed if a problem about the design occurs once moved on the implementation phase.[2]

The glory days of Waterfall methods are over and has been replaced with Agile and DevOps. The two popular methodologies both have the same goal; enable teams to work faster, more efficiently, and produce better quality.[3]

With these two present methodology giants, it can be interesting to see how Agile and DevOps work together. How do Agile and DevOps complement each other? How can you use Agile and DevOps successfully together? What are the challenges of using Agile and DevOps together?

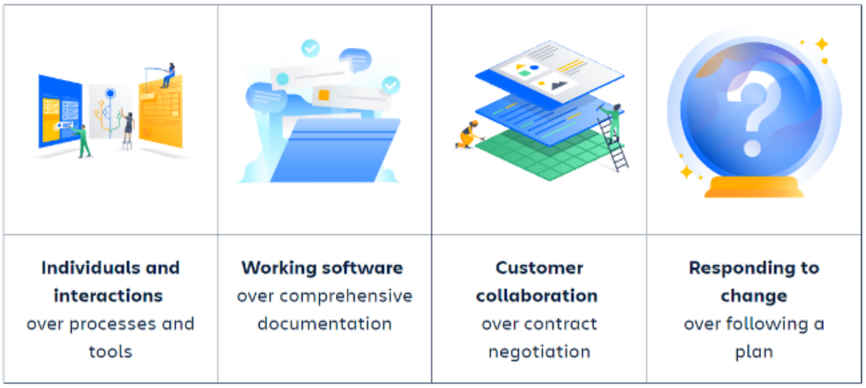
This paper finds clarity in these questions with the findings that DevOps can be seen as an extension of Agile and a way to try to handle the bottleneck of Agile methodologies. Furthermore, automation and soft skills need to be prioritized when using Agile and DevOps in conjunction, but that soft skills often are overlooked and not valued in the hiring or recruitment process.

2 Background

In this section some background information about Agile (2.1) and DevOps (2.2) is presented.

2.1 What is Agile?

Agile methods are meant to work in a complex and changeable environment [4]. Developed in the early 2000s, the now almost 20 years old methodology remains focused on having collaborative teams, visualizing projects through the use of boards, and avoiding unnecessary work. The visualizing process is often done at a Kanban board, with three different columns; Planned work, Work in progress and Finished. Tasks are written on notes, which then are moved from



Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Figure 1: The Agile Manifesto [5].

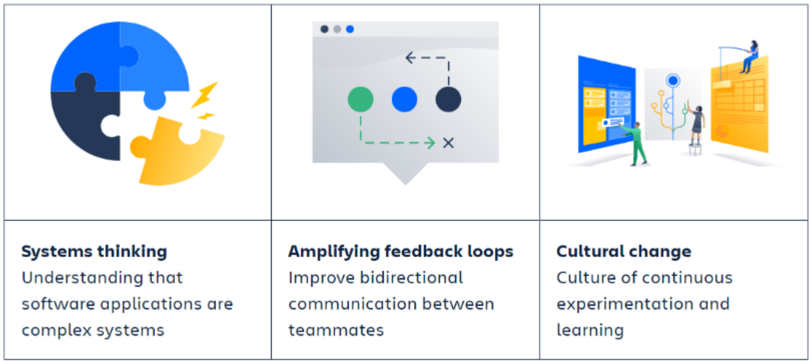
left to right on the board through the different stages of the work process. This is done in order to enhance members' focus on the project as well as individual tasks. In addition, Agile work promotes decision making that is decentralized, since that helps with the efficiency of teams. There are some guidelines for how Agile teams should be managed in the most efficient way:

1. Should have 3-9 people
2. Must have a clear and common goal
3. Must have necessary competence to carry out task
4. Members should spend most of their time on the tasks
5. The same team members should work throughout, and not be switched

One of the key characteristics of Agile work is to have a meeting once everyday with all members of the team. Each member reports the following: what did I do yesterday, what am I doing today, and are there any obstacles?.[4] Another key characteristic is that Agile is famous for having short iterative cycles, with a focus on customer feedback and quick releases. Although there are many different versions of Agile work, they share a common manifesto with four core values, see figure 1.[5]

2.2 What is DevOps?

The name DevOps gives a hint about what it is; development and operations, working together, in one continuous process. DevOps is a collection of tools, practices and cultural philosophy. It promotes communication and collaboration between different teams, team empowerment as well as automation of technology. Both developers and IT-operations are involved in a DevOps-team. The



Systems thinking
Understanding that software applications are complex systems

Amplifying feedback loops
Improve bidirectional communication between teammates

Cultural change
Culture of continuous experimentation and learning

Figure 2: The DevOps Three Ways framework [5]

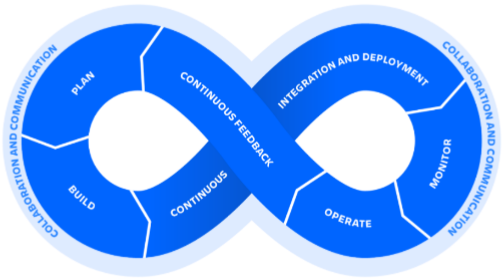

COLLABORATION AND COMMUNICATION
PLAN
BUILD
CONTINUOUS
CONTINUOUS FEEDBACK
INTEGRATION AND DEPLOYMENT
OPERATE
MONITOR
COLLABORATION AND COMMUNICATION

Figure 3: The six phases of DevOps [6].

team should work throughout the product life-cycle process; development, testing, deployment and operations. This is done to gain the benefits of higher quality and faster software deployment. Working with DevOps implies a cultural shift and a new way of working, with significant implications.[6] A common framework for understanding DevOps is “The DevOps Three Ways”, seen in figure 2.

The fundamentals in DevOps are automation, collaboration, continuous integration and continuous delivery. DevOps teams often use tools to automate these processes, which increases the speed as well as the reliability. DevOps is a continuous process, and the different stages are shown in an infinite loop in figure 3. The left hand side is connected to development, while the right side has to do with operations.[6]

3 How do Agile and DevOps Complement Each Other?

Once introduced, Agile working methods were a response to the problems of traditional working methods, such as the Waterfall method had. One of these shortcomings with traditional methods is their inability to handle a changing environment. One of the reasons for using Agile methods is to be able to deliver on time while doing it with high quality.[7] Many software developers are using Agile methods in their development process, and gain the benefits of increased productivity and being able to meet their customers' demands [8]. DevOps was created to handle the bottleneck of Agile; limitations for Development teams to deliver to Operations rapidly and frequently. However, the Agile methods are the foundation for DevOps, and in order to have a successful DevOps deployment, an Agile approach for the software development is required.[3][9]

A major reason for the Agile bottleneck mentioned above is the fact that some departments of the organizations work in silos [3]. As mentioned in 2.2, DevOps requires developers and operations to work together. Organizations are now shifting to DevOps to eliminate the silos [3]. To elaborate, Agile planning can have disadvantages including missed deadlines and completed elements that are incompatible with each other due to them being created by different groups [10]. It becomes clear that the weak link here in Agile working is insufficient communication [3]. DevOps is suggested as a solution for the problem. In DevOps settings, the communication happens internally in the department as well as with other departments, because it is mandatory for developers and operators to coordinate.[10]

It is always a good idea to work with Agile and Devops in tandem in a team [3]. Promoters of using the two styles together highlight that DevOps can be viewed as an extension of Agile. In Agile, cross-functional teams consisting of people with different expertise are important. These teams typically include designers, developers and testers.[11] By installing an operator in charge of supporting the shift from software to implementation, DevOps enables more cross-functionality and elimination of groups working in silos. By having rigorous communication between DevOps and other teams, automation of processes and increased openness are enabled.[12]

4 How Can Agile and DevOps Be Used Successfully Together?

As mentioned in section 2.2, automation is one of the key parts in DevOps. The different levels of automation can be described in the following way [9]:

Level 1- Agile way of working: More frequent releases and iteration. Development and Operations work in silos.
Level 2- Continuous integration: Development and Operations work together.

Both make tests (unit and non-regression), and the process is as automated as possible.
Level 3- Continuous delivery: Development and Operations work together on performance tests, end-to-end tests, user acceptance tests.[9]

Automation Level 3 requires some different skills of the personnel compared to Level 1. In this case, more so called soft skills are required in Level 3.[9] Soft skills are non-technical and can determine a person's strength in for instance leadership or mediating. Some examples of soft skills are *communication*, *courtesy*, *interpersonal skills*, *team-work*, and *work ethic.* Hard skills on the other hand have to do with using technical knowledge and the ability to perform. Some examples of hard skills are *coding*, *testing*, and *release management.*[13] The soft skill *communication* has been found to be more common in organizations with Level 3 automation. This is also true for *flexibility*, *interpersonal skills*, and *team-work.* The latter three soft skills are also more important in DevOps compared to Agile. However, the soft skill *responsibility* tends to go down as the automation level increases. One reason for this can be that *responsibility* in an Agile setting is divided in a formal fashion, while in a continuous delivery setting, there is more of a shared sense of *responsibility.* Collaboration has been mentioned a lot throughout this essay, and it was found that collaboration increases when automation increases. This included a higher frequency of collaboration as well as collaboration being more efficient.[9] Another enabler for collaboration is sharing transparency. Additionally, deployment agility and resilience are outcomes from having a collaborative culture.[14]

With all of this together, one can probably say that in order for having a successful collaboration of Agile and DevOps, an organization should aim for Level 3 in automation, encourage sharing transparency and focus on soft skills.

5 What Are the Challenges of Using DevOps and Agile Together?

As mentioned in section 3, it is a good idea to use Agile and DevOps together. However, one needs to understand that these theories change and that it cannot be seen as a solution that can solve all problems of the organization.[3] Section 4 described that Agile and DevOps has an impact on collaboration. Thus, it is highly important for organizations to grasp the requirement about a new set of skills in the workplace and educators to understand these new requirements when forming curriculums for IT-students [15][16]. It is common for companies to neglect the soft skills in DevOps job-advertisements, and only focus on the hard skills, which has led to a lack of people having soft skills being recruited [17][18]. This can potentially lead to organizations not optimizing their chances for a successful Agile and DevOps way of working.

6 Discussion and Conclusion

Reflecting on the fact that DevOps uses Agile as a foundation, this paper was built around the idea that Agile is essential for DevOps to work successfully. With that in mind, this paper explored how DevOps can be extended from Agile, rather than merging the two theories together from two different directions. This may not come as a surprise since DevOps tries to solve Agile methods' shortcomings. The paper has not covered any material that implies a true conflict between the two working methods, but rather that an organization needs to "step up" when building DevOps on top of Agile. It is clear from this paper that a higher level of automation actually changes the collaborative patterns. This is of course great news and a clear message to organizations. However, one can still think that organizations need to be mindful with this and not see it as the ultimate solution. If the organization lacks collaborative personnel, the level of collaboration might not reach a desired level, even when Level 3 in automation is reached. Therefore, it could be a good idea for organizations to focus on building collaboration, while increasing the level of automation.

To conclude, in this essay it has become clear that DevOps is something that is a great extension of Agile, but that an Agile approach is necessary to successfully implement DevOps. While Agile has been a breakthrough in the industry, organizations still experience missed deadlines and incompatible work. DevOps has tried to solve this by having close collaboration between development and operations, while having automation as one of the key pillars. This paper also made it clear that a higher level of automation promotes collaboration. However, soft skills are important when the level of automation increases, and the three soft skills of *flexibility*, *interpersonal skills*, and *team-wor*k are more important in DevOps compared to Agile. Despite this fact, soft skills are often not promoted in job advertisements. Therefore, it is probably not far-fetched to encourage organizations that want to combine Agile and DevOps to make sure that people with soft skills are recruited or kept, while also aiming for a Level 3 in automation.

To round off, if only one take-at-home message can be remembered from this essay it should be: *Using an Agile way of working together with DevOps is a good idea for organizations.*

7 References

[1] H. Yuan, Y. Han, and J. Hu, ”Research on agile development methodology of service-oriented personalized software” International Conference on Computer Science and Software Engineering , March. 7, 2011. Accessed May. 16, 2022 [Online] Available: https://www.semanticscholar.org/paper/Research-on-Agile-Development-Methodology-of-Yuan-Han/8a4a321719aad26ba3ccf3e0f4309c4ae21d85c8

[2] L. Hoory and C. Bottorff, ”What Is Waterfall Methodology? Here's How It Can Help Your Project Management Strategy”. Forbes.com. https://www.forbes.com/advisor/business/what-is-waterfall-methodology/ (accessed May. 18, 2022)

[3] M. M. Sikender. “DevOps automation and Agile methodology”. International Journal of Creative Research Thoughts (IJCRT), Wilmington, Canada, Aug. 3, 2017. Accessed May. 5, 2022 [Online] Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3655581

[4] Tonnquist. B. “Agile Methods” in Project Management, 4th ed. Stockholm, Sweden: Sanoma Utbildning AB, 2018.

[5] T. Hall. “Agile vs. DevOps”. Atlassian.com. https://www.atlassian.com/devops/what-is-devops/agile-vs-devops (accessed May. 4, 2022)

[6] Atlassian. “What is DevOps”. Atlassian.com https://www.atlassian.com/devops (accessed May. 5, 2022)

[7] J. Highsmith and A. Cockburn, Agile software development: The business of innovation. IEEE, 2001. Accessed: May. 10, 2022. [Online]. Available: /urlhttps://ieeexplore.ieee.org/document/947100

[8] M. Laanti, O. Salo, and P. Abrahamsson, ”Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation”, Information and Software Technology, 2011. Accessed May. 5, 2022 [Online] Available: https://www.sciencedirect.com/science/article/abs/pii/S0950584910002119

[9] B. Fitzgerald, A. Hemon, B. Lyonnet, and F. Rowe, “From Agile to DevOps: Smart Skills and Collaborations”, Accessed: May. 10, 2022. [Online]. Available: https://www.researchgate.net/profile/Frantz-Rowe/publication/3315 76124_From_Agile_to_DevOps_Smart_Skills_and_Collaborations/links/5f56909792851c250b9cea14/From-Agile-to-DevOps-Smart-Skills-and-Collaborations.pdf

[10] N. Forsgren, and J. Humble, ”DevOps: Profiles in ITSM Performance and Contributing Factors”, SSRN Electronic Journal, 2015. Accessed: May. 10,

2022. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2681906

[11] N. Dzamashvili Fogelström, T. Gorschek, M. Svahnberg, and P. Olsson, "The impact of agile principles on market-driven software product development", Journal of Software Maintenance and Evolution: Research and Practice. Accessed: May. 11, 2022. [Online]. Available: https://www.researchgate.net/publication/220674020_The_impact_of_agile_principles_on_market-driven_software_product_development

[12] M. Drury, K. Conboy, and K. Power, "Obstacles to decision making in Agile software development teams", Journal of Systems and Software, 2012. Accessed: May. 11, 2022. [Online]. Available: https://www.researchgate.net/publ ication/256991595_Obstacles_to_decision_making_in_Agile_software_development _teams

[13] M. Maram, P. Prabhakaran, S. Murthy, and N. Domala, "Sixteen roles performed by software engineers in first one year. IEEE, 2009. Accessed: May. 11, 2022. [Online]. Available: https://www.researchgate.net/publication/224400633_Sixteen_Roles_Performed_by_Software_Engineers_in_First_One_Year

[14] W. Pinheiro Luz, G. Pinto, and R. Bonifácio, "Adopting DevOps in the real world: A theory, a model, and a case study", Journal of Systems and Software, 2019. Accessed: May. 18, 2022. [Online]. Available: https://www.science direct.com/science/article/pii/S0164121219301517

[15] C.H. Tan, and H.-H. Teo, "Training future software developers to acquire agile development skills", Communications of the ACM, 2007. Accessed: May. 12, 2022. [Online]. Available: https://www.researchgate.net/publication/220421 $392_{T}raining_{f}uture_{s}oftware_{d}evelopers_{t}o_{a}cquire_{a}gile_{d}evelopment_{s}kills$

[16] S. K. Bang, S. Chung, Y. Choh, and M. Dupuis, "A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps". In Proceedings of the RIIT, ACM, Orlando, FL, USA. 2013. Accessed: May. 12, 2022. [Online]. Available: https://www.researchgate.net/publication/266654602_A_grounded_theory_analysis_of_modern_web_applications_knowledge_skills_and_abilities_for_DevOps

[17]W. Hussain, T. Clear, and S. MacDonell, "Emerging trends for global DevOps: a New Zealand perspective". IEEE Press. Buenos Aires, Argentina, 2017. Accessed: May. 12, 2022. [Online]. Available: https://www.researchgate.net/publication/318474630_Emerging_Trends_for_Global_DevOps_A_New_Zealand_Perspective

[18] E. M. Trauth, D. W. Farwell, and D. Lee, "The IS expectation gap: Industry expectations versus academic preparation". Mis Quarterly, 1993. Accessed:

May. 12, 2022. [Online]. Available: https://www.researchgate.net/publ ication/220260197_The_IS_Expectation_Gap_Industry_Expectations_Versus_Aca demic_Preparation