

The rise of DevQualOps and implications on software quality

RALUCA DOVLEAC

Department of Management and Industrial Engineering

University of Petrosani

No. 20 Street, Petrosani, Hunedoara

ROMANIA

Abstract: Quality of software products and development processes has been seen through the lens of customer oriented practices lately. We can observe an emphasis on agile development and development approaches that promise fast, iterative releases which are in accordance to the customer and/or stakeholder's requirements. While the delivery time has been significantly cut short by these approaches, time after time we witness several other drawbacks of these methodologies (costs raised by technical debt, Agile burnout, issues regarding scalability and so on) as well as new methodologies emerging in order to provide a fix for some of these issues. The latest approach in software development methodologies was the DevOps development model, a model that aimed to bridge the gap between Development and Operations personnel. Even so, issues regarding to software quality continued to emerge, which led to the birth of DevQualOps, a software development model that integrates more aspects and tools of quality management within the development process. The paper looks at the implications and possible benefits of implementing and using the DevQualOps development model as an alternative to those already used.

Key-Words: DevOps, DevQualOps, agile development, software development, quality, software quality, quality assurance, lean software

Received: March 30, 2022. Revised: February 12, 2023. Accepted: March 15, 2023. Published: April 11, 2023.

1 Introduction

The term "Devops" was first introduced in 2009, in the context of companies providing rapid releases [1], and has continued to steadily grown in popularity since, with an impressive number of papers covering the topic of DevOps and its implication in practice [2] [3].

Recent concerns however, in term of quality of software products coming from companies using the Agile or DevOps approach have shifted the focus to the efficiency of these development approaches and has led some researchers to consider possible alternatives. One such alternative, which evolved from DevOps is DevQualOps, a model meant to empower quality within the development process, with hopes of increasing quality of software products and software development processes and thus reducing the cost of poor quality and the satisfaction of both customers and development teams.

The current paper looks at the DevQualOps model from an evolutionary point, highlighting the course it took from software quality to the Agile manifesto and its shortcomings regarding quality, then at DevOps with its respective shortcomings regarding quality and then analyzing the

DevQualOps model as a possible alternative to those methodologies, an alternative which will hopefully help bridge the gap between software development and software quality once and for all, and if not, at least raise more awareness regarding the means by which quality is obtained, monitored, controlled and improved within the software development community.

2 Agile development and software quality

Although emerging software development methodologies such as the Agile development approach reduced the time to market of software products, improved communication within team members, enforced a flexibility of the design process and helped define more reasonable processes [4], this trend wasn't without repercussions.

Worldwide, for most organizations, software quality has been noticed to lag behind other targets and company objectives. The cost of poor software quality for the year 2020 was estimated at roughly 2.08 trillion dollars, while the cost of issues raised in relation to technical debt was estimated to be around 1.31 trillion dollars. [5]

In the next sections, we will look at software quality, in order to understand what it requires and how it is measured, at the Agile development approach in order to highlight the pros and cons of implementing it and the relationship between agile driven development and software quality and lastly, we will examine DevOps – a development approach that emerged as an improved alternative to agile development methodologies.

The current paper addresses the existing knowledge gap in literature regarding the DevQualOps approach to software development and how it can help companies align their quality goals with efficient software development.

2.1 Software quality

When talking about software quality, we must take into account the definition proposed by the IEEE Standard Glossary of Software Engineering Terminology according to which the quality of software is the degree to which a system, component or process meets the specified requirements and the needs or expectations of a user. [6]

An approach used in practice when trying to evaluate the quality of software products is the usage of quality models for highlighting the relationships between various factors taken into account when evaluating quality. Some of these models include: The McCall quality model, The ISO 9126 standard, Dromey's model and modelling quality as a process. But even so, with these models and more others, a problem still persists – the fact that quality remains a complex concept that is hard to measure given its highly context-dependent aspect. [7]

2.2 Agile manifesto

Since the core values and principle of the agile development approach have been stated in the “Manifesto for Agile Software Development” in 2001 [8], a large number of companies worldwide implemented it, in part due to its advertised flexibility, iterative approach to software development and culture of frequent releases of products that met customer requirements. This large-scale adoption has been achieved despite emerging various issues related to the implementation of agile methodologies. [9] [10]

Recent years have brought to surface an increase dissatisfaction with agile methodologies, while some of the very creators of the Agile Manifesto for Software Development even going as far as saying that “Agile is dead”. [11]

Some of the dissatisfaction with the agile approach to software development came from the rigid implementation of agile methodologies, the transition of focus from better software development practices to certification concerns, misuse of agile principles and practice, a sense of micromanagement on the Scrum master's side, unrealistic time goals for sprints and unrealistic expectations of daily standups amongst others. [12] [13] [14]

Even more worryingly, recent talks regarding the effectiveness of agile development focused on the long-term effects of agile development on the developer's well-being, highlighting thus the risk of “Agile burnout”, a state of burnout that developers working in agile development are more likely to develop. [15] [16]

2.3 DevOps

DevOps was born out of the desire to bridge the gap between the development and operations teams [17], with an emphasis on bringing software changes into production more frequently and with faster feedback cycles [18]. Some of the most significant aspects regarding DevOps include: a culture of collaboration, a focus on automation, clear metrics and measurements, embracing sharing of information within the software process, structuring the organization around services rather than discipline, a different approach to organization governance and an increase attention to quality assurance. [2]

Summarizing, DevOps can be defined as a development methodology which focuses on core principles such as: communication and collaboration, continuous integration, quality assurance, delivery and automated deployment for delivering products while also bridging the gap between development and operations. [3]

An important aspect covered by DevOps is the concern for the quality of software products released and frequency of these releases. [19] [20] Quality assurance, in the context of DevOps has manifested in scheduling, feedback to design decisions, prioritization of bugs and decisions regarding releases of software with bugs as well as risk assessments. [21]

Although beneficial in some cases in practice, DevOps has some core problems that still prevail such as: issues regarding scaling across the organization due to manual processes, issues regarding the coordination between the development and operation teams and overlapping tools that hinder agility. [22] Another important issue identified was the lack of robust quality concerns.

The focus on quality concerns or robust testing emerged only at the end of the development cycle, meaning that code was in most cases built to be functional and not necessarily of high quality. [5]

3 DevQualOps – bridging the gap between software development and software quality

DevQualOps emerged as a solution to address issues related to quality of software products, acknowledging that software products have evolved in complexity and time to market, and with it, the customer expectations evolved as well [23].

DevQualOps aims at integrating quality within the process of software creation and delivery, in order to achieve customer excitement. Some propose that this can be achieved by addressing four main concerns, as shown in Figure 1.



Fig.1 DevQualOps core principles

As it can be observed from Figure 1, the core principles of DevQualOps include: a shift towards a culture of quality where all personnel work to achieve the quality goals rather than relying and placing the responsibility on the Quality Assurance (QA) team; an emphasis on understanding the customer and its needs; investing time and resources in the development of the team and empowering them to become the change agents within the company and informing others of the company's products roadmap. Once feedback is available, it's recommended to use approaches such as Behavior-Driven Development (BDD) to optimize development cycles. [23]

A recent report from CISQ (CONSORTIUM FOR INFORMATION & SOFTWARE QUALITY) highlighted the key enablers for achieving the highest levels of costs, schedule and quality performance in software development presented in Figure 2. [5]

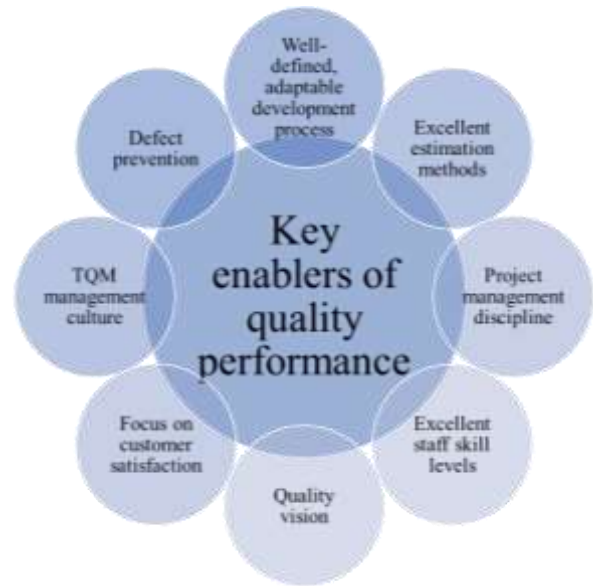


Fig.2 DevQualOps key enablers for quality

We can observe that the enablers focus on one hand on TQM (Total Quality Management) aspects such as: defect prevention, focus on customer satisfaction, establishing a quality vision and quality culture and using well defined processes, as well on the project management aspect highlighting thus the need for a project management discipline, excellent estimation methods and excellent staff skill levels.

Still, the same report proposes a set of techniques for ensuring higher quality software and shorter development cycles, as shown in Figure 3. [5] These techniques have been grouped by the author of the paper according to their contribution in a PDCA (Plan, Do, Check, Act) cycle.

The PDCA cycle is a simple yet powerful quality management tool that enables the visualization of the key activities that are part of the cycle and their contribution to the overall established objectives. The cycle starts with the planning phase, similar to any other management approach, then continues with the execution phase called “Do” in this case, followed by checkups and verification activities in the “Check” phase and lastly places an emphasis on making changes and a decision making process in the last stage, the “Act” one. The cycle can then be repeated until the company or individual is happy with the final result and the established objectives have been met.

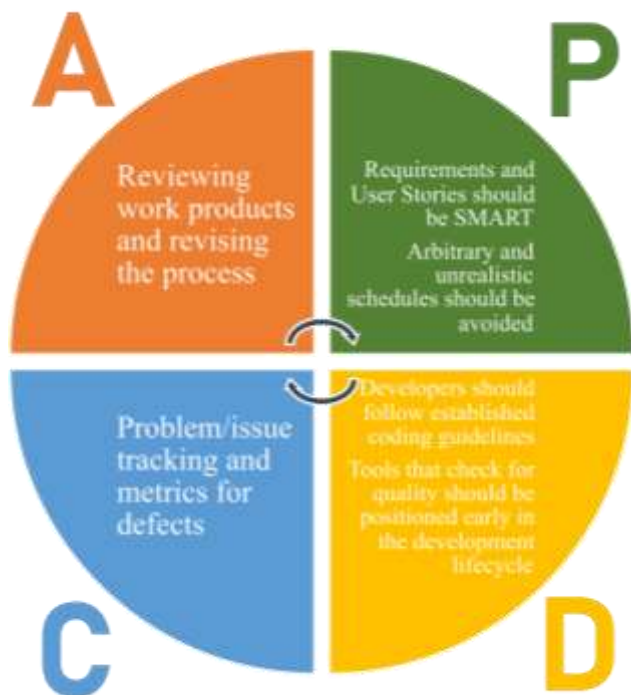


Fig.3 PDCA approach to techniques used for higher quality software

It is the opinion of the author that grouping the proposed tools according to their contribution to each stage of the PDCA cycle can facilitate the integration of the latter in the development process and can highlight the most essential aspects regarding quality for each stage of the development process. As it can be observed from Figure 3, a qualitative approach to software development begins in the “Plan” phase, with an emphasis on establishing SMART requirements, user stories, use cases and user scenarios along with avoiding arbitrary and unrealistic schedules. During the “Do” phase of the PDCA cycle the development team should focus on following well established coding guidelines in order to ensure that code is written in a consistent style and the number of defects is lower on one hand, while also aiming to integrate tools that check for quality as quickly as possible in the development process. During the “Check” phase, an emphasis is placed on problem tracking, on establishing and checking metrics for defects. During the last phase of the cycle, named the “Act” phase, activities regarding the review of the obtained work products are undertaken and the process is revised in order to understand what worked well and what didn’t and where the development team could bring improvements.

The DevQualOps process model presented in Figure 4 highlights the relationship between Agile development process and DevOps processes while ensuring an appropriate level of quality across the Agile and DevOps development lifecycle. [5]

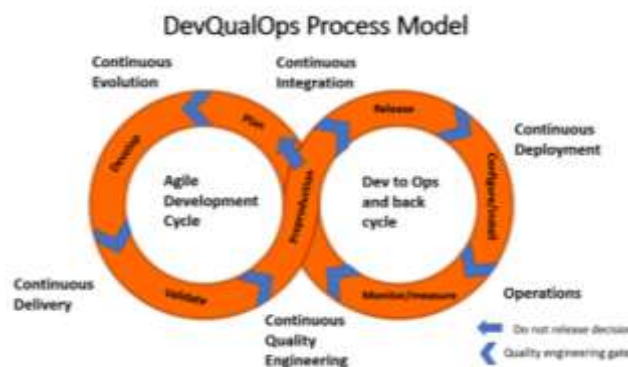


Fig.4 DevQualOps process model [5]

As it can be observed from Figure 4, the DevQualOps model proposed by CISQ facilitates the integration and/or transition from Agile development cycle to a DevOps development cycle while maintaining an emphasis on continuous evolution, integration and delivery and introducing the concept of Continuous Quality Engineering in order to raise awareness on the importance of quality in software development and establish its place and role within the development process. We can observe that the two development processes follow a PDCA cycle approach with activities such as Planning, Production, Validation, Development, Release, Configuration and Monitoring. We can also observe a shift toward empowering quality within the development process, with quality engineering gates becoming an integral part of the process and being placed at key points in the process.

The authors of the model also highlight the necessity for establishing measurable quality objectives and conducting trend analysis along with defect prevention practices implemented as part of continuous process improvement and making use of automated tools for every aspect of Software Quality Engineering (SQE) and not just testing. [5]

4 Conclusion

DevQualOps, the intersection of the DevOps development methodology and software quality has emerged as a possible solution to the poor quality of software products that use Agile or DevOps development methodologies for their development process.

DevQualOps encompasses both Agile and DevOps methodologies key principles such as

shortening the development times and creating more responsive development cycles but also addresses issues in the quality of the resulting product and the development process itself.

In the current paper, the author analyzed the role that DevQualOps can play within a development process and proposed a PDCA based approach for encompassing the quality tools suggested for implementation in the development process of companies looking to obtain higher quality results and using Agile or DevOps software development methodologies.

The contributions and benefits of adopting a DevQualOps model will be more visible and relevant once feedback from companies using this model is gathered and that is why it is the opinion of the author that further research directions should focus on case studies and observations regarding the usage of DevQualOps within the development process and noting the results obtained, both in term of quality of final product as well as quality of the development process and overall satisfaction of the development team and stakeholders.

References:

- [1] J. Humble and J. Molesky, "Why enterprises must adopt DevOps to enable continuous delivery," *Cutt. IT J.*, vol. 24, no. 8, pp. 6-12, 2011.
- [2] F. Erich, C. Amrit and M. Daneva, "A Qualitative Study of DevOps Usage in Practice," *JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS*, pp. 1-47, 2017.
- [3] R. Jabbari, N. b. Ali, B. Tanveer and K. Petersen, "What is DevOps? A Systematic Mapping Study on Definitions and Practices," in *XP '16 Workshops: Proceedings of the Scientific Workshop*, New York, 2016.
- [4] B. Choudhary and S. Rakesh, "An approach using agile method for software development," in *International Conference on Innovation and Challenges in Cyber Security*, 2016.
- [5] H. KRASNER, "The Cost of Poor Software Quality in the US: A 2020 Report," *CONSORTIUM FOR INFORMATION & SOFTWARE QUALITY*, 2020.
- [6] J. P. Miguel, D. Mauricio and G. Rodríguez, "A Review of Software Quality Models for the Evaluation of Software Products," *International Journal of Software Engineering & Applications*, vol. 5, no. 6, pp. 31-53, 2014.
- [7] B. KITCHENHAM and S. L. PFLEEGER, "SOFTWARE QUALITY: THE ELUSIVE TARGET," *IEEE SOFTWARE*, vol. 13, no. 1, pp. 12-21, 1996.
- [8] M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, K. Schwaber, J. Sutherland and D. Thomas, "Manifesto for Agile Software Development," 2001. [Online]. Available: <https://agilemanifesto.org/>. [Accessed 01 03 2023].
- [9] T. Dybå and T. Dingsøyr, "What do we know about agile software development?," *IEEE Software*, vol. 26, no. 5, pp. 6-9, 2009.
- [10] K. Dikert, M. Paasivaara and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review.," *Journal of Systems and Software*, vol. 119, pp. 87-108, 2016.
- [11] D. Thomas, Writer, *Agile is Dead*. [Performance]. GOTO Conferences, 2015.
- [12] K. Cagle, "The End of Agile," *Forbes*, 23 08 2019. [Online]. Available: <https://www.forbes.com/sites/cognitiveworld/2019/08/23/the-end-of-agile/>. [Accessed 2023 01 03].
- [13] S. Middleton, "Agile is Dead, McKinsey Just Killed It," *Medium.com*, 14 07 2021. [Online]. Available: <https://medium.com/agileinsider/agile-is-dead-mckinsey-just-killed-it-547df8c6c9df>. [Accessed 01 03 2023].
- [14] R. Jeffries, "Developers Should Abandon Agile," *ronjeffries.com*, 10 05 2018. [Online]. Available: <https://ronjeffries.com/articles/018-01ff/abandon-1/>. [Accessed 01 03 2023].
- [15] Wrike.com, "What Is Agile Burnout?," *Wrike.com*, [Online]. Available: <https://www.wrike.com/agile-guide/faq/what-is-agile-burnout/>. [Accessed 01 03 2023].
- [16] S. Sharma, "How To Eliminate Burnout And Make Agile Teams More Productive," *knowledgehut*, 21 02 2023. [Online]. Available: <https://www.knowledgehut.com/blog/agile/how-to-eliminate-burnout-and-make-agile-teams-more-productive>. [Accessed 01 03 2023].
- [17] A. Brunnert, A. van Hoorn, F. Willnecker, A. Danciu, W. Hasselbring, C. Heger, N. Herbst, P. Jamshidi, R. Jung, J. von Kistowski, A. Koziol, J. Kroß, S. Spinner, C. Vögele, J. Walter and A. Wert, "Performance-oriented DevOps: A Research Agenda," *research.spec.org*, 2015.
- [18] C.-P. Bezemer, S. Eismann, V. Ferme, J. Grohmann, R. Heinrich, P. Jamshidi, W. Shang, A. van Hoorn, M. Villavicencio, J. Walter and F. Willnecker, "How is

Performance Addressed in DevOps? A Survey on Industrial Practices," in ICPE '19, Mumbai, 2019.

- [19] L. Riungu-Kalliosaari, S. Mäkinen, L. E. Lwakatare, J. Tiihonen and T. Männistö, "DevOps Adoption Benefits and Challenges in Practice: A Case Study," in Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trodheim, 2016.
- [20] C. Ebert, G. Gallardo, J. Hernantes and N. Serrano, DevOps, *IEEE COMPUTER SOCIETY*, 2016.
- [21] J. Roche, Adopting DevOps Practices in Quality Assurance, *ACM*, 2013.
- [22] S. M. Mohammad, "An exploratory study of DevOps and it's future in the United States," *IJCRT*, vol. 4, no. 1, pp. 114-117, 2016.
- [23] J. Schloss, "The Real-Life Steps to Making DevQualOps Happen," *cmswire*, 07 05 2021. [Online]. Available: <https://www.cmswire.com/digital-experience/the-real-life-steps-to-making-devqualops-happen/>. [Accessed 08 03 2023].