Faculty of Informatics
Masaryk University

IV064
Information Society

Modern Agile Software Engineering

Adrián Tóth (491322) November 16, 2019

Contents

1 Software Engineering

From time to time, software engineering become the most important part of software development. There are many different definitions of software engineering which are sometimes misleading and incorrect. The *IEEE*[1] defines the software engineering, which clarifies its true meaning in the software development.

Definition 1.1. Software engineering

> The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software [1].

Up to the present time, software engineering has evolved and adapted by the time, introducing new methodologies for software development, which lead to an increase in the number of successful projects. Software development methodologies provide a framework for planning, executing, and managing the process of developing software systems [7]. The importance of software engineering has resulted in a research of software engineering itself in a form of empirical, contemplative, case and other studies, which delivered new approaches, attitudes and practices. The speed of information technology development is massively progressive and incredibly fast and it is also part of our daily life.

Definition 1.2. Delivery

> Release of a system or component to its customer or intended user [1].

Nowadays requirements for the software are unbearably high. Developing a software present days means, to deliver the product as soon as possible and simultaneously provide the best possible product quality, which is classified by product metrics. Quality plays a vital role for the software users [4]. This article deals with the software delivery time from the view of the modern software engineering.

Developing a quality software product is an essential need for the software industry [4]. To deliver a quality product under short time, it requires development agility, cross-functional teams and collaborative effort of self-organization. To deliver a quality product as fast as possible, it is needed to meet the above mentioned requirements and use a proper framework and methodology. The most common development methodologies is agile as well as predictive methodology. In addition to agile development methodology, we will discuss and compare the newly integrated methodologies and practices.

1.1 Time to Market

Time to the market is the key to success in the field of information technology. Businesses must be prepared and shaped to adapt and evolve the modern breakthrough technologies. It is not

[1] Institute of Electrical and Electronics Engineers

so easy to be up to date, especially not in the information technology business. From time to time, as every manual tasks were transformed into a fully automated hands–off processes, the software engineered was also influenced by this automation impact. Automation is limitless because of the fact that there can be any task transfered from manual to automated. If the automated process is configured properly, it may save a huge amount of time.

The software development must be also faster, the automation of the software delivery pipeline is not the only one required for a quick software delivery. Time needed for software product creation is the most influencing part of the delivery time. Based on the development progress the time may be also prolonged or shortened.

2 Agile Development

Delivering a product in the shortest possible time requires good software development methodology, which is possible with agile development methodology. Agile methodology provides all the necessary approaches, practices and disciplines to the software development, which allow fast product delivery and high quality of the product. The main and principal approaches to software development are: flexible and adaptable, focus on people (customers, users, developers), regularly updated requirements. In a comparison to predictive methodology, agile provides less (minimal) effort to upfront planning and the product is delivered after small iterations in a form of product increments (releases). Agile methods are based on small development cycles, continuous integration of software versions, adaptive planning, team collaboration, customer involvement, and feedback [9].

Definition 2.1. Scrum

> A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value [6].

SCRUM is the most well known and common agile software development framework. It is simple to understand but difficult to master. The framework describes and defines the whole development structure – events, roles and artifacts. The scrum framework poster in Figure 1 provides a graphical view of how scrum is implemented at a team level within an organization [5].

It is difficult to state precisely the advantages and disadvantages against agile alternatives. Both agile and predictive have pros and cons which can differ by the project requirements. Predictive approach is greatly used for a project where we exactly know the product requirements and specifications. Agile approach is useful while it is needed to deliver the product (or maybe its prototype) in the quickest time while it might be later enhanced by additional patches via updates. The manner of time and product specifications (requirements definition) are crucial to state the shortcomings of each methodologies.

2.1 DevOps

A new development approach has been created to reduce the time costs for software product delivery. You likely have heard of the word DevOps, so let us clarify its true meaning. The modern software development involves an immense effort to the product deployment because of integration and connection conflicts between each product parts. DevOps efficiently integrates development, delivery, and operations, thus facilitating a lean, fluid connection of these traditionally separated silos [3]. It came to address this conflict and bridge the gap between developers and operations staff [8].

Definition 2.2. DevOps

> DevOps is an organizational approach that stresses empathy and cross-functional collaboration within and between teams – especially development and IT operations – in software development organizations, in order to operate resilient systems and accelerate delivery of changes. [2].

The acronym consists from two parts Dev and Ops. DevOps is an abbreviation of development and IT operations, it is not limited to those two teams; thus, there is no need for "extending" acronyms like DevSecOps or DevNetOps [2].

3 Continuous Delivery

3.1 Continuous Deployment

3.2 Continuous Integration

3.3 Deployment Pipeline

4 References

[1] *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE Std 610*. 1991. doi:10.1109/IEEESTD.1991.106963. [Online; Accessed: 2019-11-12].
Retrieved from: https://ieeexplore.ieee.org/document/182763

[2] Dyck, A.; Penners, R.; Lichter, H.: *Towards Definitions for Release Engineering and DevOps*. In *2015 IEEE/ACM 3rd International Workshop on Release Engineering*. 2015. doi:10.1109/RELENG.2015.10. [Online; Accessed: 2019-11-16].
Retrieved from: https://ieeexplore.ieee.org/document/7169442

[3] Ebert, C.; Gallardo, G.; Hernantes, J.; et al.: *DevOps*. *IEEE Software*. 2016. doi:10.1109/MS.2016.68. [Online; Accessed: 2019-11-16].
Retrieved from: https://ieeexplore.ieee.org/abstract/document/7458761

[4] Jain, P.; Sharma, A.; Ahuja, L.: *The Impact of Agile Software Development Process on the Quality of Software Product*. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. 2018. doi:10.1109/ICRITO.2018.8748529. [Online; Accessed: 2019-11-12].
Retrieved from: https://ieeexplore.ieee.org/document/8748529

[5] Scrum.org: *The Scrum Framework Poster*. 2019. [Online; Accessed: 2019-11-12].
Retrieved from: https://www.scrum.org/resources/scrum-framework-poster

[6] Scrum.org: *The Scrum Guide*. 2019. [Online; Accessed: 2019-11-12].
Retrieved from: https://www.scrum.org/resources/scrum-guide

[7] Vijayasarathy, L. R.; Butler, C. W.: *Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? IEEE Software*. 2016. doi:10.1109/MS.2015.26. [Online; Accessed: 2019-11-12].
Retrieved from: https://ieeexplore.ieee.org/document/7006383

[8] Wahaballa, A.; Wahballa, O.; Abdellatief, M.; et al.: *Toward unified DevOps model*. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. 2015. doi:10.1109/ICSESS.2015.7339039. [Online; Accessed: 2019-11-16].
Retrieved from: https://ieeexplore.ieee.org/document/7339039

[9] Younas, M.; Jawawi, D. N.; Ghani, I.; et al.: *Agile development in the cloud computing environment: A systematic review*. *Information and Software Technology*. 2018: pp. 142 – 158. doi:https://doi.org/10.1016/j.infsof.2018.06.014. [Online; Accessed: 2019-11-12].
Retrieved from:
https://www.sciencedirect.com/science/article/pii/S0950584918301319

5 Appendix

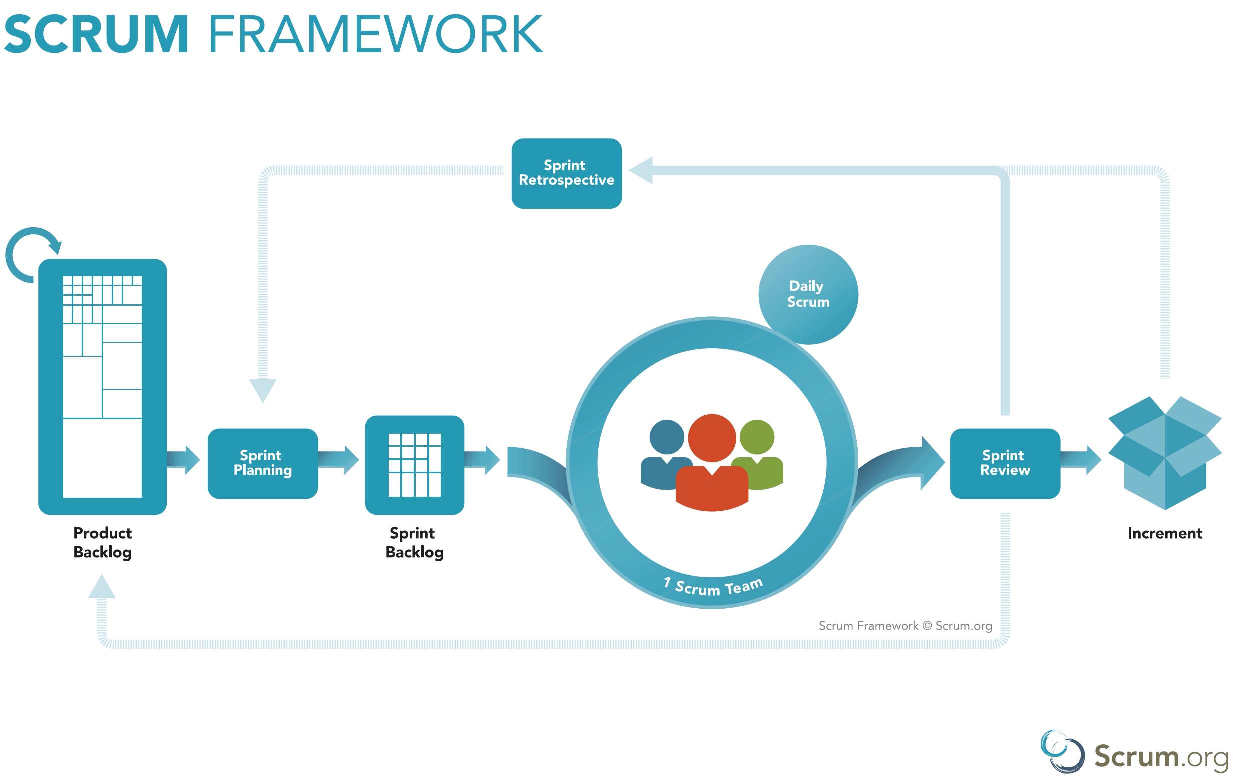

Figure 1: The Scrum Framework Poster [5].