Quality assurance within the agile system development life-cycle

[1]Dovleac Raluca, [2]Suciu Cristina

Departament: [1]Management, [2]Industrial Engineering

University:University of Petroșani

Address: Petrosani, 1 December 1918, st.1

ROMÂNIA

email account: [1]PhD Student, Drd.ec. raluca.dovleac@yahoo.com, *[2]PhD Student, Drd.ec.* cristina_suciu27@yahoo.com

Abstract: Agile development methodologies have gained popularity as of lately due to their flexibility, continuous testing, customer involvement and short iterations. Companies using this approach have observed improvements in their product delivery time and customer satisfaction. A question that arises from these observations is: Does Agile incorporate Quality Assurance in its development practice, and could this be a reason behind its efficiency? Furthermore, if Agile does not integrate Quality Assurance practices in its methodologies, would the implementation of such practices increase the productivity of this development approach and provide a useful guideline for practitioners? In our paper we have analyzed these issues in order to better understand the relation between Quality Assurance and Agile as well as provide directions towards future integrations of Quality Assurance aspects within Agile development practices. We did this by first taking a look at what Agile means and what it stands for, its evolution in time and core principles, then analyzing the quality management process and the role of quality assurance within this discipline. We then looked at how quality is measured in Agile and at the success factors in the case of Agile methodologies and compared them to the success factors of Total Quality Management (TQM). By doing this, we highlighted the success factors which were present in both TQM and Agile methodologies and provided directions for further improvements.

Key-words: Agile, Quality, Quality management, Project management

1. Introduction

Given the complexity of the products offered in today’s markets, the booming number of software products available for almost any need and the large number of competitors, forced producers to make better products in terms of quality while maintaining a low price. These aspects led to a “revolution” in the field of product development, and especially software products development, which encouraged the development of an easy to use project management tool that would adapt per a company’s need and help develop better and faster.

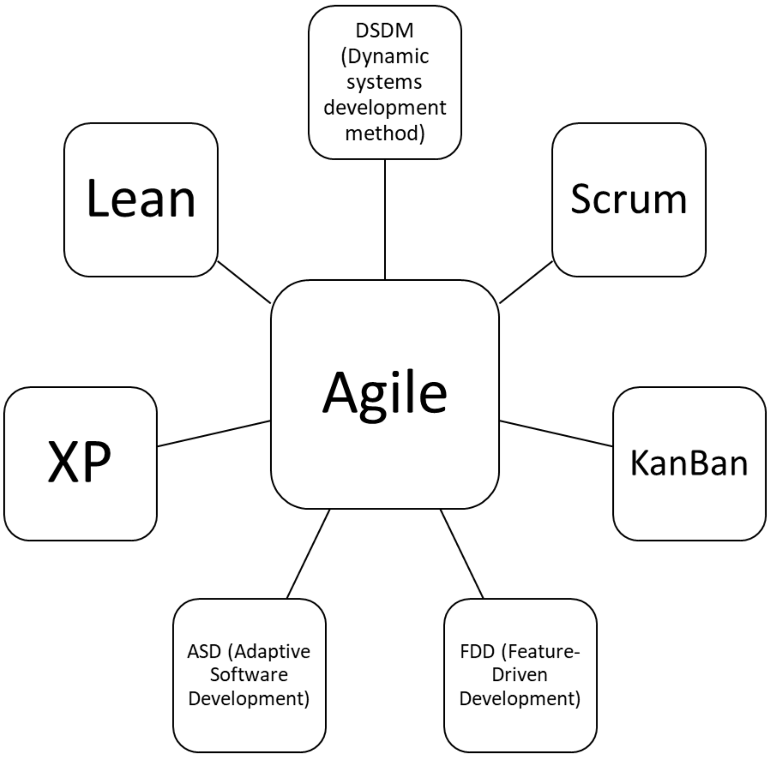
Agile was born in the mid 90’s as a response to these demands and as an alternative to the bureaucratic project management practices and traditional product life cycles (Beck, Extreme Programming Explained – Embrace Change 2000), by proposing an incremental design that would allow rapid prototyping and shorter development circles along with a solution for changing requirements (Boehm and Turner, Balancing Agility and Discipline: 2003). The proposed approach allows development phases to overlap during the development process rather than waiting for a step to be complete in order to move on to the next one.

Ever since its introduction to the software development community, Agile has been a controversial topic for practitioners and researchers altogether (Boehm and Turner, Using risk to balance agile and plan-driven methods 2003). However, Agile methodologies have gained acceptance in the commercial area due to their facilitation of customer integration in the production process and early delivery. Two of the most significant benefits of using Agile are: the management of unstable requirements throughout the development lifecycle and shorter delivery time while being able to work with budget constraints (Grenning 2001) (Murru, Deias and Mugheddue 2003) (Rasmussen 2003) (Schuh 2001).

Agile software development is a collection of software development methodologies which all inherit the iterative development characteristic, while valuing self-organizing, cross-functional teams. Agile methodologies provide a disciplined project management process through the help of constant inspection and adaptation to customer demands, and the promotion of rapid delivery of quality software through the help of short iterations (CPRIME n.d.). The term “Agile” was born in 2001 when a group of software development practitioners and experts met to discuss the ways to improve the software development process (Beedle, et al. 2001).

The collection of software development methodologies that are part of the Agile development approach can be observed in Figure 1.

Fig.1 Agile development methodologies


DSDM (Dynamic systems development method)
Lean
Scrum
Agile
XP
KanBan
ASD (Adaptive Software Development)
FDD (Feature-Driven Development)

The Agile approach is based on four foundational values and twelve principles to help ensure high performing, cross-functional and self-organizing teams deliver quality products in a short time span. The four values can be seen in Figure 2 while the twelve principles can be observed in Figure 3.

Fig.2 The four Agile values

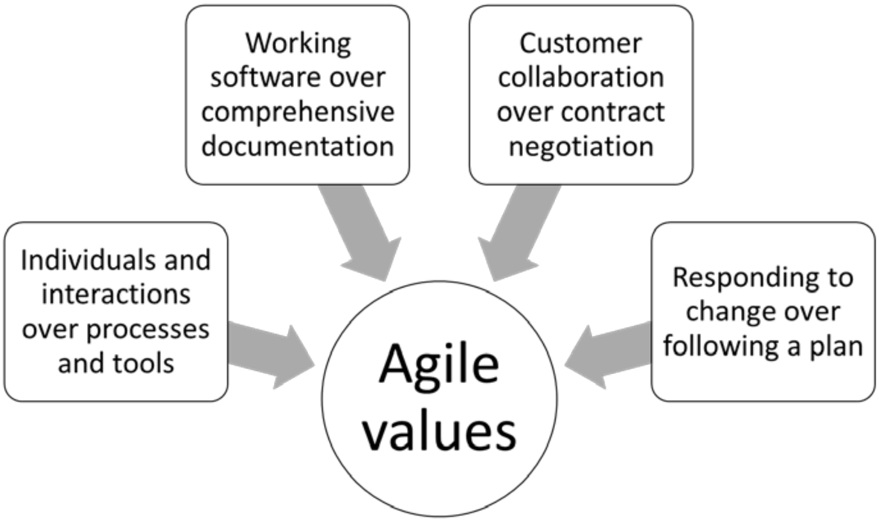
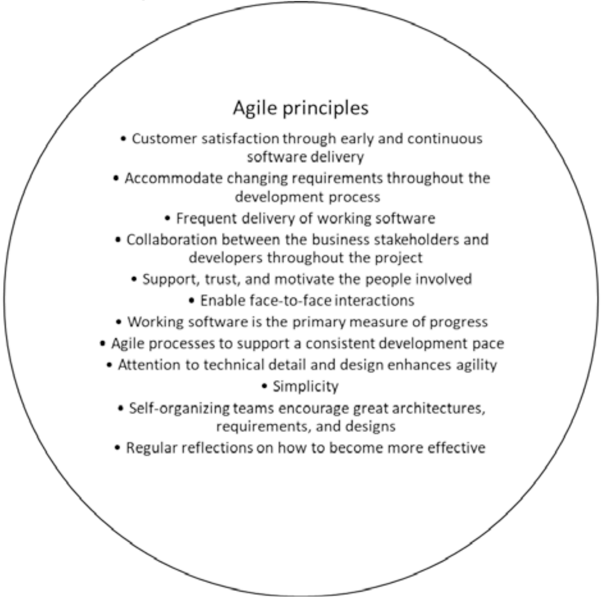

Working software over comprehensive documentation
Customer collaboration over contract negotiation
Individuals and interactions over processes and tools
Agile values
Responding to change over following a plan

Fig.3 Twelve Agile principles



Agile principles

- Customer satisfaction through early and continuous software delivery
- Accommodate changing requirements throughout the development process
- Frequent delivery of working software
- Collaboration between the business stakeholders and developers throughout the project
- Support, trust, and motivate the people involved
- Enable face-to-face interactions
- Working software is the primary measure of progress
- Agile processes to support a consistent development pace
- Attention to technical detail and design enhances agility
- Simplicity
- Self-organizing teams encourage great architectures, requirements, and designs
- Regular reflections on how to become more effective

Quality management is a process which ensures that a company, product or service is consistent and meets the established quality standards imposed and the ways in which it achieves and maintains it through the help of its four main components: quality planning, quality assurance, quality control and quality improvement (Wikipedia n.d.).

In the context of quality management, Quality Assurance (QA) is the systematic process of determining if a product and/or service meets the quality requirements by establishing and maintaining a set of requirements to be followed in order to produce reliable products. A QA system has the purpose of improving the work processes while also providing a trust reason to customers. The QA process has three components: measurement of performance; comparison of registered performance with existing standards and providing adequate actions for maintaining or achieving quality as suggested by the standard (Dyk , et al. 1993) (Rouse n.d.).

2. Materials and methods

Given the fact that Agile has become increasingly popular in the last years, with more practitioners integrating one or more Agile methodologies into their development process, the need for quality assurance and finding new ways of achieving quality assurance within these projects has grown as well.

Many of today's organization face the challenge of ensuring quality within the Agile methodologies. This is due to the fact that QA practices which proved useful and efficient in the past are no longer fitted for the Agile environment, given the agility of the environment and the fact that many formal processes have turned informal and furthermore make it impossible to impose rules and documentation in order to achieve quality assurance. It can be concluded that in the case of Agile development, QA activities have to be integrated within the team daily activities, while maintaining flexibility, transparency and collaboration (Bhasin 2012).

While some Agile methodologies help complement the practice of Total Quality Management (TQM) such as the customer focus, the employee commitment and satisfaction, there are also features which come in contrast to TQM practices. Such features refer to things such as: the plan driven approach versus agility, process emphasis versus product emphasis, much documentation versus executable lean documentation (Siakas and Siakas 2007;).

To better understand these aspects, one must first look at the way quality has been defined from an Agile perspective. McBreen defines agile quality as the development of software that can respond to chance as the customer requires it to change (McBreen 2002). By looking at this definition we can deduce that the frequent delivery of working software in accordance to customer demands constitutes an important aspect of agile quality assurance.

Another definition considers agile quality to be a result of practices such as effective collaborative work, incremental development, and iterative development as implemented through techniques such as refactoring, test-driven development, modelling and effective communication techniques (Stamelos and Sfetsos 2007).

The evaluation of quality assurance in Agile processes has been done through a comparative analysis between quality in the waterfall development model and quality in the Agile group of methodologies. The results of this study showed that quality assurance is present in the case of Agile development processes but it is achieved differently from the traditional processes (Huo, et al. 2004).

A correlation between the quality management parameters and the features available in Agile Techniques can be observed in Table 1. The purpose of the table is to highlight where quality and Agile meet and the common features they inherit in order to better evaluate the QA process within Agile methodologies.

Table 1. Software Quality parameters within Agile Techniques

Software Quality Parameters	Agile Techniques
Correctness	Write code from minimal requirements. Specification is obtained by direct communication with the customer. Customer is allowed to change specification. Test-driven development.
Robustness	Not directly addressed in agile development.
Extendibility	A general feature of all OO developed applications. Emphasis is on technical excellence and good design. Emphasis also on achieving best architecture.

Reusability	A general feature of all OO developed applications. There are some arguments against reusability of agile products (Turk, France, & Rumpe, 2002; Weisert, 2002).
Compatibility	A general feature of all OO developed applications.
Efficiency	Apply good coding standards.
Portability	Practice of continuous integration in extreme programming.
Timeliness	Strongest point of agility, Short cycles, quick delivery, etc.
Integrity	Not directly addressed in agile development.
Verifiability	Test-driven development is another strength of agility.
Ease of use	Since the customer is part of the team, and customers give feedback frequently, they will most likely recommend a system that is easy to use.

Source: Stamelos and Sfetsos 2007

As previously debated, Agile methodologies can incorporate quality assurance within their processes but require some changes to help fit the QA practices. For instance, in the case of Waterfall development process, the matter of quality and stability are usually dealt with in the later phases of the release, when changes are costlier to fix. On the opposite side, in the case of Agile development, smaller builds that provide increasing functionality are presented early in the development process to customers in order to achieve validation, and furthermore are often updated through their iterative process. The role of QA in Agile methodologies therefore is to ensure quality and stability in each iteration of the development process (Bhasin 2012).

An interesting aspect of quality management and its aid in the case of Agile has been also covered in a study which looked for the successful implementation of quality management tools in the Agile development process in order to aid Agile processes and methodologies (Dovleac and Ionica 2017).

Another interesting aspect to factor in, is the similarities between success factors in the practice of Total Quality Management (TQM) and Agile development. The success factors can be observed in Table 2.

Table 2. Success factors in TQM and Agile development

Success factors	TQM	Agile development
Leadership	Emphasis on management response and the intrinsic link between leadership and the quality of processes	Management and software developers have an equal role in the leadership of projects
Customer	Strong customer/user emphasis	Customers on board
Focus	Customer satisfaction focus	Increase customer satisfaction
Flexibility	Predictive approach	Adaptive approach
	Long term plans	No detailed plans, short term plans
	Emphasis on processes	Flexibility to changing requirements
	Compliance to standards	High iteration frequency
	Bureaucracy	Test-driven development
	Extended documentation	Executable lean documentation

Motivation	Management commitment and leadership are the driving factors for motivating employees to strive for continuous process improvement	Self-directed teams consisting of highly skilled, motivated and innovative software engineers
Job satisfaction	Replaceable roles	Skilled individuals
Empowerment commitment	Empowerment emphasizes a cultural change by highlighting commitment at all levels in the organization	Empowerment of software developers, communication and end-user participation supported
Collaboration	Collaboration horizontally and vertically between and within departments	Small teams working in co-located development environment reflecting on efficiency and tuning accordingly

Source: Siakas and Siakas 2007

3. Results and discussion

The results suggest that the usage of Agile methodologies in correlation with quality assurance practices lead to a high success rate (Abbas, Gravell and Wills 2010). When Agile methodologies have a good impact on the overall quality of the product and / or service it turns out that they also impact the productivity and costs generated by the development of the product, which can lead to an increased business value. Furthermore, by analyzing the relationship between various success factors in Agile development and their impact on achieving and maintaining high quality, a QA framework for Agile development methodologies can be proposed to help companies deliver high quality products and ensure quality throughout the development lifecycle (Bhasin 2012).

It is therefore recommended that Agile practitioners pay attention to the QA process throughout the development cycle as well as look for ways to improve the integration of these processes in the Agile development practices.

Bibliography

1. Abbas, Noura , Andrew M Gravell, and Gary B Wills. 2010. "Using Factor Analysis to Generate Clusters of Agile." *AGILE Conference.* Nashville: IEEE.
2. Beck, K. 2000. *Extreme Programming Explained – Embrace Change.* Reading: Addison Wesley Longman.
3. Beck, K. 2003. *Test Driven Development: By Example.* Boston,: Addison Wesley.
4. Beedle, Mike , Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, Jim Highsmith , Andrew Hunt, et al. 2001. *Manifesto for Agile Software Development.* Accessed 04 23, 2018. http://agilemanifesto.org/.
5. Bhasin, Sonali. 2012. "Quality Assurance in Agile –A study towards achieving excellence."
6. Boehm , B., and R. Turner. 2003. *Balancing Agility and Discipline:.* Boston: Addison Wesley Professionals.
7. Boehm , B., and R. Turner. 2003. "Using risk to balance agile and plan-driven methods." *Computer* 36: 57-66.
8. CPRIME. n.d. *WHAT IS AGILE? WHAT IS SCRUM?* Accessed 04 23, 2018. https://www.cprime.com/resources/what-is-agile-what-is-scrum/.
9. Dovleac, R., and A. Ionica. 2017. “Quality Management techniques embedded in Agile Project Development” *MATEC Web Conf.* 121.
10. Dyk , J. van, R. B. Barnett, J. E. Cygler, and P. C. Shragge . 1993.
11. “COMMISSIONING AND QUALITY ASSURANCE OF TREATMENT PLANNING COMPUTERS*.” *Int.J.Radiation Oncology Biol. Phys.* 26: 261-271.
12. Grenning, J. . 2001. "Launching extreme programming at a process-intensive company,." *Software, IEEE* 18: 27-33.
13. Huo, M., J. Verner, L. Zhu , and M. A. Babar. 2004. "Software quality and agile methods." *IEEE Computer.*
14. McBreen, P. 2002. "Quality assurance on agile." Software Craftsmanship Inc.
15. Murru, O., R. Deias, and G. Mugheddue. 2003. "Assessing XP at a European Internet company." *Software, IEEE* 20: 37-43.
16. Rasmussen, J. 2003. "Introducing XP into Greenfield Projects lessons learned." *Software, IEEE* 20: 21-28.
17. Rouse, Margaret . n.d. *quality assurance (QA).* Accessed 04 23, 2018. https://searchsoftwarequality.techtarget.com/definition/quality-assurance.
18. Schuh, P. . 2001. "Recovery, redemption, and extreme programming." *Software, IEEE* 18: 34-41.
19. Siakas, Kerstin V. , and Errikos Siakas. 2007;. "The Agile Professional Culture: A Source of Agile Quality." *Softw. Process Improve. Pract* 12: 597–610. doi:10.1002/spip.344.
20. Stamelos, Ioannis G. , and Panagiotis Sfetsos. 2007. *Agile Software Development Quality Assurance.* IGI Global.
21. Wikipedia. n.d. *Quality management.* Accessed 04 23, 2018. https://en.wikipedia.org/wiki/Quality_management.