

# The Impact of Agile Methods on Software Project Management

Mahdad Khelghatdost<sup>\*</sup>, Ali Mohsenzadeh<sup>2</sup>, Hesamoldin Pourkarim Dodangeh<sup>3</sup>

Department of Computer Engineering, Mazandaran University of science and Technology

Received: June 10 2013

Accepted: July 11 2013

---

## ABSTRACT

As more and more software projects engage Agile Methods, there are emerging patterns of success and failure. With growing adoption of Agile Methods, project managers increasingly need to understand the applicability to their projects and factors that drive key project performance characteristics. While some organizations affirm that Agile Methods solve all their problems; few have shown consistent success over a range of typical software projects. Agile Methods have advantages, especially in accommodating change due to volatile requirements. This paper summarizes several paper examines the impact of agile methods on software project management, particularly on the subjects of projects will be discussed.

**KEYWORDS:** Software project management, Agile methods, XP, Scrum

---

## 1. INTRODUCTION

Software engineering as a discipline, against there are two key challenges that separates it from other engineering disciplines. Software, a product concept and is often palpable. Although the software can be changed, but these changes will lead to increased costs in the project lifecycle.[1]

A lot of agile methodologies have been introduced in the past decade ,Such as XP and Scrum. While these methods have their own characteristics, but have a common goal of rapid response to change. When responding to rapid changes in the end of project, has many cost, Ability to respond quickly to changes, reduce project costs and risks. The challenge for managers is to determine whether a method is faster for a set of project activities. All software development methodologies are risk and understand the risks and find ways to monitor, reduce and manage those risks is an important aspect of project management software. In this paper, the impact of agile methods on software project management review is to identify some of the strengths and weaknesses.

## 2. The Agile Methodology Statement

In 2001, a group of 17 members who works on the agile approaches provide a statement known as the agile statement. Its concepts are in order:

- 1- The superiority of the individualism and interaction to processes and tools.
- 2- The superiority of the executable software to conceptual documents.
- 3-The superiority of the collaboration with clients to contract-based negotiations.
- 4- The superiority of the variations responding to following an initial plan.[2]

And some explanations are followed:

1-The superiority of the individualism and interaction to processes and tools: A group of average programmers with good relationships usually do much better than a group of professional ones without any good relationships.

2- The superiority of the executable software to conceptual documents: Software has no limited documents because it is meaningless to produce such documents. Also, excessive documents are worse than limited ones as it is time consuming to produce them.

3- The superiority of the collaboration with clients to contract-based negotiations: A successful project is a project that frequently receives clients' responses.

4- The superiority of the variations responding to following an initial plan: It is not possible to produce the final product considering a preliminary framework and design, because minor and major requirements of the design often become visible during the development and so the developer team should make required changes in designing with respect to these requirements.[3]

## 3. The Definition of Agile Software Development

A framework is a concept that deals with the accomplishment of software engineering projects. Thus, the agile is a combination of a philosophy with a series of software developing instructions. [4]

### **3.1 The Overall Philosophy of the Agile Designing**

The philosophy could be summarized in the form of below concepts which in turn result to the agile popularity:

1. Encouraging customers.
2. The incremental and early delivery.
3. A small but full of motivations team.
4. Informal methods.
5. The least activity.
6. Simplicity.

### **3.2 The Instructions of Agile Designing**

Below, there are two important instructions in agile designing:

1. Enforcing in order to deliver products early instead of analysis and designing;
2. A continuous contact between the developer and customer.

### **3.3 The Agile Methodology Projects' Characteristics**

- 1- Being modular at the development process level;
- 2- Being repeatable by short repetition periods,
- 3- Scheduling by 1-6 week repetition periods,
- 4- Scrounging in the development process by abandoning all of the unnecessary activities,
- 5- Being adaptive by new probable risks,
- 6- The incremental process trend which leads to the functionality of the project and making it in small steps,
- 7- The incremental and convergence trend which decreases risks.[5]

## **4. Agile Methods**

Nearly after 2000, various methods in agile categories have been introduced. They often seek to be committed to agile concepts. Two methods have been suggested in this section:

### **4.1 The XP Approach**

XP is one of the most famous agile methods. It uses simultaneously some programming concepts such as designing and implementation, software testing and the client participation in the development team. Team members including the user set a meeting and define project priorities. They are usually some programmers, software testers and analysts who collaborate in the whole project. Programmers work dual and write some tests before writing the program. While one of them writes a part of program, another one controls the accuracy of codes and fixes the probable problems. This is one of the XP advantages which enhance the product software quality. [6][7]

### **4.2 The Scrum Approach**

Nowadays, the Scrum is one of the software production ways especially for small ones. Small software could be made using this method. This way is mainly emphasizes the flexibility, compatibility and usefulness. Each member should understand his/her task in the project and follow a certain goal in all of the operational and executive phases. As Scrum is a framework, it does not give complete and accurate details about how works should be done in the project and so assigns these affairs to the team. Actually, the Scrum manager is responsible to identify members' tasks, investigate the evolutionary process of software construction and abilities of members, and work in order to reduce the risks of project. It is one of the Scrum advantages that it could be divided a big project to small and understandable parts which could be simultaneously accomplished. [8][9][10]

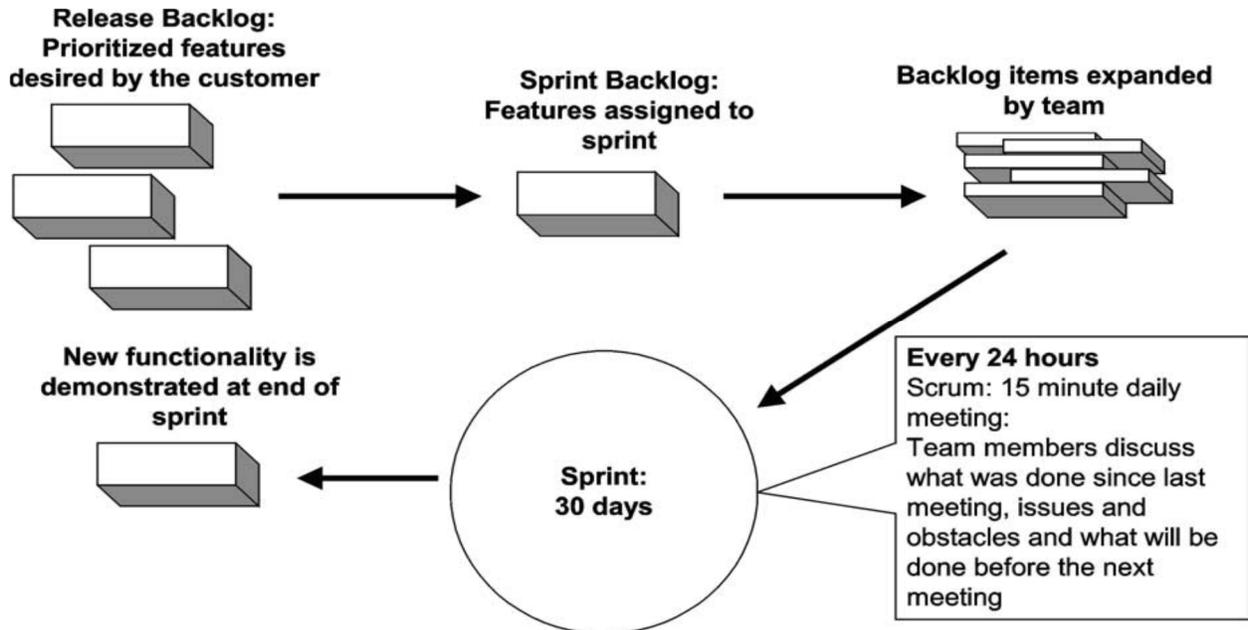


FIG1. The Scrum Lifecycle [11]

**5. The impact of agile methods on project management**

While the ideas and intent behind Agile Methods are by and large good, they have impacts on the people, process, and project elements of an effort. We examine some of these impacts to determine whether an Agile Method can and should be applied to a project, given its requirements, available staff, and external factors such as business and legal constraints.

**5.1 People**

There can be a range of people involved in a software effort – developers, testers, project leaders to name a few. There is often a customer and an end user who wants the resulting product. There are also executive managers (business executives and directors of the development shop), who are interested in budgets and returns on investment, and human resources. Each of these has a stake in an agile project.

**5.1.1 Developers**

Perhaps the largest impact of Agile Methods is on the Developers. Agile Methods depend on strong developers – they must be amicable, talented, skilled, and able to communicate well [1]. Developers must be willing to work as a team, able to handle constant change, and resourceful enough to solve problems. Agile Methods are very lightweight methods, not affording strict guidelines and processes for developers to follow. Hence, they do not accommodate weaker developers well. Yet, skilled technology workers are often a rare commodity. This is a management risk as some developers may not fit in this agile environment.

Level	Characteristics
3	able to produce solutions in unprecedented situations
2	able to tailor solutions to fit new, but preceded situation
1A	solid developer able to implement functionality, estimate effort, & refactor code
1B	able to implement simple functionality, execute tests, & follow directions
-1	unwilling or unable to work in a collaborative environment

Table 1. Boehm & Turner’s developer levels [13]

The “-1” level of developer depicted in Table 1, would be challenged in an agile environment. Even “1B” developers consume resources in “hand-holding” [14]. Hence, the top three levels make up the core of the agile development team. Boehm and Turner suggest level “3” developers may not be needed for all projects, depending on how unprecedented it might be. Given the need for a high level of expertise, Agile Methods may be difficult to employ in a traditionally staffed organization. Highly skilled staffs are always in demand, and without accommodating 1B developers, it may be difficult to build a long term human capital strategy. This is just one reason that long term projects present a significant risk for Agile Methods.

### 5.1.2 Testers

The impact of using an Agile Method on the testing (or quality assurance) organization hinges on the shorter development cycles where testing occurs throughout the development process [14]. Testers must work closely with the developers as code is being written. In Agile Methods such as XP, tests are changed before code is modified by the developers and the role of a tester is significantly reduced [7]. Testers focus on system and functional tests as more of an independent validation and verification role. Testers may need to be more capable as programmers to automate their system and functional tests and incorporate them into the automated testing framework. This may represent a different skill set. The project management challenge is to reallocate testers that no longer fit into the agile group and find testers with appropriate development/testing skills. This represents an opportunity for novice developers (level “1B”) to start, and gain system and Agile Method expertise. Such an approach requires one more experienced developer (or experienced test manager).

### 5.1.3 Project Leaders

There are two key Project Leader roles in software development – project managers and team leads. Each has a diverse set of challenges as management under an Agile Method differs from other methodologies. This distinction is well characterized as leading people and managing process resources. Since Agile teams involve experienced staff with sizeable responsibility, a mentor or coach leadership approach is most effective. Team leads must be willing to enable members to take initiative. Leadership is done via collaboration rather than command and control type leadership [1]. This can represent a cultural shift for some as they must be willing to share decision making authority [15]. The job of a team lead is to facilitate the team into making decisions [7]. In contrast, project managers in agile processes are responsible for tracking progress and making business decisions. Project managers have a larger adjustment than team leads since schedules and plans are far less important under agile methodologies. The emphasis is placed on responding to change rather than following a specific plan. This presents a challenge as they are usually called upon to detail the status of the project. Project managers also have a much more involved role. In SCRUM, for example, the project manager meets with the team daily and leads the daily SCRUM [1]. Frequent/short meetings with the team are the norm for the agile team [14]. Project managers are also more involved with the customer collaboration, instead of usual focusing on defining deliverables and contracts. If the project manager considering an Agile Method is not capable or does not want such a role, selecting an Agile Method may not be appropriate.

### 5.1.4 Customers

In the agile, as the customer is more involved in producing the software, he/she has one of the most valuable roles in comparison with other software production ways [16]. For instance in traditional methods like the cascade method, clients just interact with developers to determine requirements at the first stage as well as they are involved at the end to receive alpha and beta versions of the project while in agile methods it is recommended that customers stay with the production and development team during the construction of the software product. In this method, it is necessary to become aware of the availability of customer representatives and their awareness of final requirements before starting. [17][18] In agile methods, there is usually a beneficiary one as a candidate showing all who are involved in the project. He/she should be an expert and be able to make important decisions such as the acceptance of the product, prioritization, etc. [18]

### 5.1.5 The Team

Since Agile Methods rely substantially on collaboration and communication, the team is key for success. A single strong-willed developer, developers who do not work well together, a customer who doesn't engage with the team, each could destroy the collaborative nature of a group. The team chemistry is of represents a significant risk for the agile project. Turnover is another significant personnel factor to be considered with an agile team. Without formal documentation, high turnover on a project can lead to loss of critical knowledge. While this can be mitigated by code reviews and having developers rotate working on different functional areas, the loss of a significant member

of a team can still be catastrophic. The project manager must consider this risk when examining whether the team (and the organization) is right for an Agile Method. Recognizing a key tenant for XP is retaining relevant knowledge by retaining good people.

## **5.2 Process**

Since Agile Methods represent a new principles, processes activities, and sub-goals, they have an impact on many of an organization's processes. Old processes (e.g., planning, development, delivery, operations) must be replaced by agile ones. Cultural shifts in the organization towards Agile Methods turn old ways of thinking on their end, inducing resistance.

### **5.2.1 Planning**

Agile processes are characterized by placing less emphasis on formal planning. This is not to say that planning does not occur. With so many small tasks, it is argued that agile processes require more planning, but unlike other methodologies, planning is not upfront followed by micro adjustments. Rather it is constant task to ensure optimal delivery results [14]. Agile planning is a relatively informal process. For example, deciding what will go into each time-box is accomplished through the daily SCRUM meeting by discussing pending problems, prioritizing work, and assigning resources to the problems [16]. In other agile methods, even this level of planning may not be considered [18]. It is important to factor in informality.

### **5.2.2 Documentation**

In Agile Methods, documentation is sparse – often limited to source code and a set of user stories as in XP[2]. Most Agile Methods do allow for an optional architecture to be developed, and in DSDM it is even mandatory [19]. The driving factor for documentation in Agile Methods is how often it is going to change and need to be updated. A vision statement for a project might be extremely beneficial and never change recording it warranted. Conversely, a low-level component design would be more likely to change, inducing some redundant document changes as well. This documentation-light process avoids wasted effort where documents are written once and then become obsolete as they are not updated to reflect the changes. With Agile processes, information is communicated informally and is simply kept as part of the collective knowledge of the organization. While reducing the amount of documentation can increase productivity, it does come at some risk and cost. Documentation serves as a way to bring new members up to speed. It is useful when transitioning the project to a maintenance team. From a business perspective, documents form the basis for audits assuring proper quality procedures are followed. Documentation serves as a domain knowledge repository. If the organization changes dramatically, this knowledge can be lost.

### **5.2.3 Development Processes**

Agile processes often encourage principles that dramatically change the process. While many of these are not limited to Agile Methods, Agile development encourages if not require their usage. Key development processes of interest are refactoring, minimalist development, code reviews, and continuous integration. Refactoring is the process of taking code and improving it without losing any functionality. Code might be improved for readability, maintainability, or performance. In refactoring, the code must pass all tests and abide by all defined contracts after it is rewritten. The development process question here is “when is refactoring prioritized over adding new functionality?” “Minimalist development within the Agile Methods community it is known as the YAGNI precept an acronym for “You Aren’t Going to Need It.” Under YAGNI, features not needed for the current functional product are stripped out to keep the implementation simple. This reduces effort as well as “gold-plating “where unneeded functionality trickles in. The risk with YAGNI is that sometimes future requirements are known and building the system to support these requirements can lead to less effort down the road by eliminating major refactoring [20][21]. Projects that have well-defined future directions may not benefit from this aspect of the agile development process. Code reviews are the process whereby one or more developers examines the code written by another. This could be continuous as in the pair programming aspect of XP [2] or be periodic as in the peer reviews incorporated in DSDM [19]. A key advantage of code reviews in this context is that they serve as a method of communication. Developers become familiar with the inner workings, design tradeoffs, and open issues with areas of the code they may be required to work with later. This can offset the risk of losing a member of the team, either temporarily on vacation or other leave or permanently due to a change in employment. Continuous integration is the process whereby the system is tested often, usually nightly if not even more frequently [2]. Developers integrate their code into a base line and run a set of regression tests on it. Continuous integration increases quality as side-effects of a change are quickly uncovered. Since finding defects early reduces the effort of fixing them, this aspect of the agile processes can have a significant impact on quality and schedules. However, developers must write a comprehensive set of tests

to be used as regression tests and must take the time to integrate and test their code. This may require a shift in developer perspective if the developer is accustomed to simply writing code which is then tested by a different group. Many development groups already practice these principles. However, many developers are prickly about the notions of peer programming and may chafe at having to write a significant number of tests. It may be necessary for the project manager to incorporate these processes slowly and with incentives to increase the chances of their acceptance.

## 6. Conclusions

The selection of a method for a specific project must be very careful, taking into consideration many different factors, including those mentioned above. In many cases, being both Agile and stable at the same time will be necessary. A contradictory combination, it seems, and therefore extra challenging, but not impossible. As Siemens states, "We firmly believe that agility is necessary, but that it should be built on top of an appropriately mature process foundation, not instead of it" [22].

However, Agile Methods are not appropriate for all projects. A project manager must consider the characteristics of the project to ensure that an Agile Method is appropriate. The impact on the people, the process, and the project must all be considered. For example, if a team of largely junior members is applied to a project that has very well understood requirements, and mature software process is already in place in the organizations, there are three characteristics that argue against applying Agile Methods as a whole. However, the principle of small team might still be appropriate to reduce risks. Furthermore, specific challenges with using an Agile Method can be offset by adding back some formality. For example, if migration to a separate maintenance group is required, documentation could be written by the development group as part of the transition [5][10].

Agile Methods offer software project managers an alternative development and management methodology that provides good support for projects with ill-defined or rapidly changing requirements. Even on project that are questionable for the application of the entire Agile Method, underlying agile principles may still be effective. Project managers should consider its usage for such projects assuming that they have a team capable of using it and can implement the required processes. Otherwise, more traditional approaches may be more appropriate.

## REFERENCES

- 1 Highsmith J., Cockburn A., "Agile Software Development: The Business of Innovation", *Computer*, September 2001, pp120-122.
- 2 "Manifesto for Agile Software Development", <http://www.agilemanifesto.org>.
- 3 Rasmin R., "The Engineering of an object oriented Software Development methodology", PhD Thesis, york.uk.department of computer science, April 2006.
- 4 Amber S., "More Process patterns: Delivering large scale system using object technology", Cambridge university.
- 5 Paul M.C., "Agile methodologies and process discipline", *Crosstalk* (Oct. 2002).
- 6 Somerville I., "Software Engineering", 9th Edition.
- 7 K. Beck. *Extreme Programming Explained*. Addison Wesley, 2000.
- 8 Schwaber K., "Scrum development Process".
- 9 Pressman R., "Software Engineering: a practitioner's approach" 6th Edition, McGraw Hill.
- 10 Abrahamson P., Salo O., Ronkainen J., Warsta J., "Agile software development methods", VTT Publications 478 (2002).
- 11 From: <http://www.controlchaos.com>
- 12 Bailey P., Ashworth N., Wallace N., "Challenges for stakeholders in adopting XP", 3rd International on extreme Programming and agile processes in software Engineering (XP 2000), Alghero, Italy, 26-29 May
- 13 Boehm, B. and R. Turner, "Using Risk to Balance Agile and Plan-Driven Methods," *Computer*, June 2003, pp. 57-66.
- 14 Cohn, M. and D. Ford, "Introducing an Agile Process to an Organization," *Computer*, June 2003, pp. 74-78.

- 15 Williams, L. and A. Cockburn, "Agile Software Development: It's about Feedback and Change," *Computer*, June 2003, pp. 39-43.
- 16 Derbier, G., "Agile Development in the Old Economy," Proceedings of the Agile Development Conference, June 2003, pp. 125-131.
- 17 Thomas, S. "An Agile Comparison," [http://www.balagan.org.uk/work/agile\\_comparison.htm](http://www.balagan.org.uk/work/agile_comparison.htm).
- 18 Cockburn, A. And J. Highsmith, "Agile Software Development: The People Factor," *Computer*, November 2001, pp. 131-133.
- 19 Coran M., Bohner S., "The Impact of Agile Methods on Software Project Management", Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)
- 20 Boehm, B., "Get Ready for Agile Methods, with Care," *Computer*, January 2002, pp. 64-69.
- 21 DE Marco, T. and B. Boehm, "The Agile Methods Fray," *Computer*, June 2002, pp. 90-92.
- 22 Paulisch F., Volker A., "Agility—build on a mature foundation", in: Proc. Software Engineering Process Group Conference—SEPG 2002, 2002.