

Use and Significance of Agile Software Development

Manisha Singh

M.tech (Computer Science and Engineering)
Department of Computer Science
Amity University (Lucknow)

Anuradha Sharma

Senior Lecturer
Department of Computer Science
Amity University (Lucknow)

ABSTRACT

The field of agile software development is growing remarkably because of the rapidness they brought into the life cycle of a software development. Agile methods have gained the interest of software professionals and researchers globally. To overcome rapidly and fast changing software organizational business requirements, agile software development methods were brought into practice. Though a number of research studies were ended in the sector of agile development methods at different educational and academic point, but when it talks about its usage, very few detailed reporting was founded. It shows that the research is in its initial stages. The paper is making an analysis of the different objectives of current agile methods and practices, how they are different from conventional process methods, considering the difficulties appeared during the implementation of agile methodology in the project, understanding the benefits and various causes of unsuccessful agile projects and concerns and issues of their applicability. The findings indicate that organizations are enduring to scale agile development methods away from single team and single projects. Most of the respondents are using Scrum or different Scrum variants and ninety percent of respondents mentioned that agile has improved their ability to handle changing priorities and manage changing requirements of users.

General Terms

Agile software development, scrum, crystal

1. INTRODUCTION

To overcome rapidly and fast changing software organizational business requirements, agile software development methods were brought into practice. These methods mainly focus on building up the solutions more efficiently and quickly. Software developers have been accepting agile software development methods since the late 1990s, when they were first proposed in the forms of Extreme Programming [1], Scrum [2], Crystal [3], and other methodologies. Barry Boehm described agile methods as “An outgrowth of rapid prototyping and rapid development experience as well as the resurgence of a philosophy that programming is a craft rather than an industrial process.” [4] This paper is focusing on the agile software development. Software industries have been practicing agile methods. Most commonly used agile software development methods will be inspected from the point of their applicability, benefits and their acceptance in industry. In order to investigate and analyze, there is a requirement to compare the issues in the research studies, industry and literature. This will direct us to find the benefits, limitations and difficulties in shifting from conventional methods to agile software development. The rest of the paper is as follows: section 2 contains the previous related work on agile methods, section 3 contains the research questions and section 4 contains the experimental

methodology. Section 5 contains the experimental results and section 6 contains the conclusions. The last section contains the references.

2. BACKGROUND AND RELATED WORK

ASD manifesto was emerged in February, 2001, signed by a group of software expertise and professionals. The most important objectives of the agile approaches are to encourage and accelerate responses to the changing requirements, environments and congregate the project deadlines. It stated as follows (in Agile Agreement, year 2001): “To discover improved ways of developing software by doing it and helping others. We have moved toward to value [5] through this work.” [6]. Following are the main focal points of the agile development which were defined in the agile manifesto [7]:

1. Rather than focusing on processes and tools, it focuses on Individuals and interactions.
2. It focuses on functioning and running software, not the heavy and broad documentation.
3. It emphasizes on Customer association over the contract negotiation.
4. It always responds to change rather than following an appropriate plan.

2.1 Agile Principles

Additionally the twelve agile principles define the earlier described core values of agile methods [8]:

1. The main priority is to satisfy customer needs through early, continuous and iterative delivery of the valuable software.
2. Agile processes welcome the changing environments, even it would happen later in the development, and it also deals with change for customer’s benefits.
3. Agile methods deliver the working and functioning software very frequently, from a number of weeks to a number of months with a preferred shorter amount of time.
4. Project work must be done by the business people and the developers together regularly during the project.
5. Projects are being built around the motivated individuals. They give individuals the appropriate environment and the level of support they required and believe them to get the work done.
6. Face to face conversation is the most effective and efficient way of passing information to and within a development team.
7. The key of progress is working software.
8. Sustainable development is encouraged by agile processes. A constant pace should be maintained by developers, users and sponsors indefinitely.

9. Ability is enhanced by continuous attention to technical excellence and fine design.
10. Simplicity, the skill of raising the amount of work not done – is necessary.
11. Considering all the teams, the self organizing teams emerge the most excellent architectures, requirements and designs.
12. The team thinks carefully how to be converted into more efficient and effective team at regular time intervals, and then it adjusts its behavior as a result.

2.2 Main Agile Software Development Methods

2.2.1 Scrum:

Scrum basically focuses on project management where plan is difficult to move ahead. Feedback mechanisms are also important [9].

2.2.2 Crystal methodologies:

In this methodology, communication takes place in small teams that are developing non life critical software [10].

2.2.3 Extreme Programming (XP):

Extreme Programming (XP) emphasis on best and finest practice for the development with twelve practices [11].

2.2.4 Feature Driven Development:

Model-driven approach and agile development are combined in feature driven development with emphasis on iterative designs [12, 13].

2.2.5 Dynamic Software Development Method (DSDM):

In DSDM, projects are divided into three phases: pre-project, project life-cycle and post- project [14].

2.2.6 Lean Software Development:

It defines seven principles: remove waste, increase learning, take decisions as late as possible, deliver fast, build integrity make the team powerful and see the whole [15].

3. RESEARCH QUESTIONS

The paper has following objectives:

- Respondent Demographics and experience with agile development
- Observing agile methods and practices
- Agile acceptance in projects
- Different reasons for accepting agile methods
- Different benefits found from implementing agile methods
- Various causes of not succeed agile projects
- Use and preferences of agile tools

4. EXPERIMENTAL METHODOLOGY

The research was based on a survey which was conducted over a period of twelve weeks; the survey was polled. In research analysis design, individuals from different software development communities and different software industries answer the previously mentioned research questions. Two basic approaches Quantitative and qualitative approaches are used to complete the analysis. Information was gathered through interviews and questionnaires as a part of qualitative method and a survey was conducted to collect the quantitative data.

5. EXPERIMENTAL RESULTS

It is clearly indicated that software companies are accepting and implementing a variety of agile methodologies on the basis of various studies. Following are the reporting on the findings from the survey:

5.1 Respondent Demographics

In the demographics, it is clearly indicated that approximately 81% of respondents accepted they are atleast familiar with agile software development techniques. 25% of respondents were very knowledgeable whereas 19% were not or very least familiar with agile techniques. Table 1 provides the above information.

Table 1. Individuals experience with agile practices

1.	Very knowledgeable	5+ years	25%
2.	Fairly knowledgeable	3-4 years	30%
3.	Knowledgeable	1-2 years	26%
4.	Very modest/ No Knowledgeable	<1 year	19%

5.2 Agile Methods and Practices

Several Agile Software Development methodologies are available today. Scrum or the other scrum variants are still the most accepted and popular agile methodologies being used. Figure 1 shows the level of adoption of different agile software development methodologies.

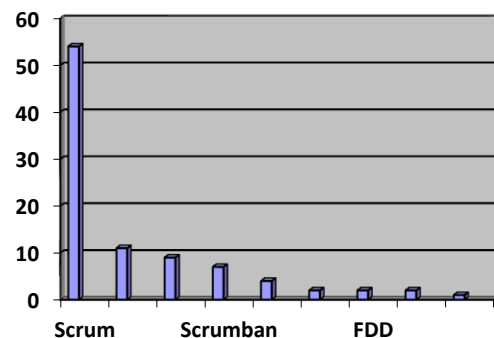


Figure 1. Different ASD Methodologies

5.3 Agile Acceptance in Projects

From an adoption point of view, it is indicated that there is an increment in the number of teams practicing agile methods at each organization surveyed. 38% of the respondents have adopted agile methods across 5 or more teams who are working at companies, up from 33% in the year 2011.

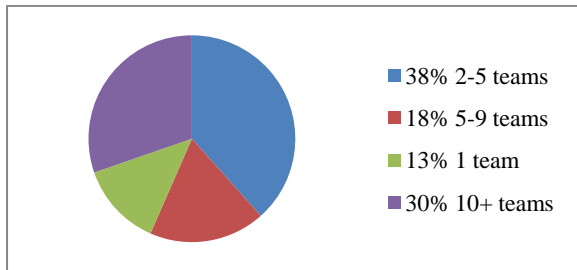


Figure 2. Teams Adopted Agile

It is also observed that the majority of respondents have approximately 5 agile projects (59%), compared to 50% in the previous year. About 30% of respondents (one third) said that their organizations have 10 or more agile practices and approximately 11% respondents said that their organizations have 6-9 agile projects. Table 2 shows the observations:

Table 2. Number of agile projects in organizations

0-5 agile projects	59%
10+ agile practices	30%
6-9 agile projects	11%

5.4 Reasons for Adopting Agile

There are a number of reasons for adopting agile in software communities. The following table illustrates the various reasons motivating adoption of agile.

Table 3. Reasons Motivating Adoption of Agile

Reasons motivating to adopt agile methodology	Agile methodology key aspect
It is able to adapt rapidly to change	Agile methodologies recognized that customer requirements will change.
It has short time frames and releases.	Agile methodologies are based on small iterations of 1 week or more between releases of working software
It has an ability to get immediate feedback from the customer.	Short time frames allow developers to gather quick feedback from the customers and users.
It provides high quality bug free software (XP) and Organizational processes demand it.	Continuous testing and integration regime of agile methodologies

It is found that the vast majority of respondents near about 70% experienced that agile projects are completed in faster time. Figure 3 indicates the overall consensus regarding adoption of agile.

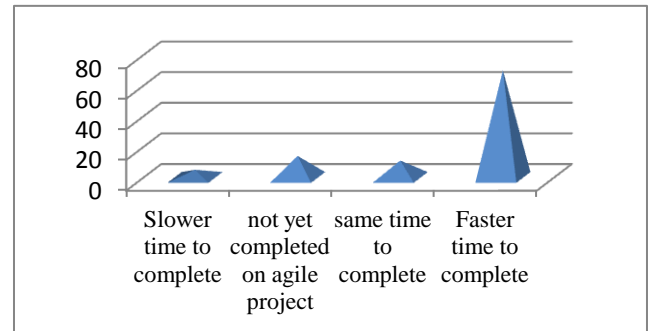


Figure 3. Overall Consensus regarding adoption of agile

It is also observed that the most important 3 reasons respondents mentioned for accepting agile were more easily handle changing priorities, to accelerate time to market, and to better align IT and business objectives. Figure 4 shows the different reasons for adopting agile in the enterprise.

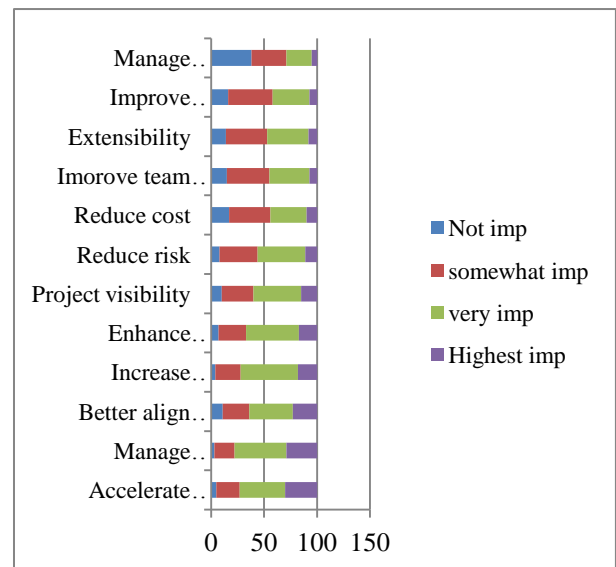


Figure 4. Reasons for adopting Agile

5.5 Benefits obtained from implementing Agile

From the findings, It is very much cleared that implementation of agile software development methods in software projects lead the projects towards a great success. It has many benefits as compared to traditional methods such as waterfall model. It is found that experienced agile users and developers of the software industries accept and said that the ability to manage changing priorities; its productivity and the project visibility were the top three benefits of agile software development methods. Overall the greatest increase is in the project visibility category from 77% in the year 2011 to 84% nowadays. There are various benefits of implementing agile methods. Figure 5 shows various benefits achieved from implementing agile software development methods.

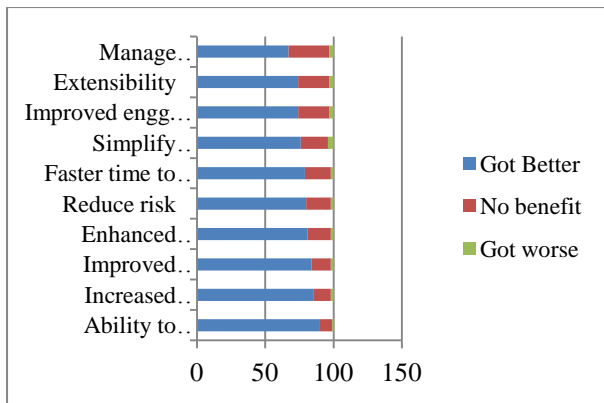


Figure 5. Various Benefits from implementing agile

5.6 Causes of Failed Agile Projects

In the findings, majority of respondents said none of their agile projects would be considered unsuccessful (18%). The respondents with failed agile projects said it was because either a company’s values or culture does not match with core agile principles (12%). They had an external pressure to trail the conventional waterfall processes (11%), organizational problems (11%) and there was a lack of experience with agile methods (9%), lack of enthusiasm in the team to follow agile methods (6%) and management support was missing (6%). Insufficient training (4%) is also a cause of it. Table 4 shows the above information.

Table 4. Causes for failed agile projects

company’s values does not match with core agile principles	12%
external pressure to trail the conventional waterfall processes	11%
organizational problems	11%
lack of experience with agile methods	9%
lack of enthusiasm in the team to follow agile methods	6%
Lack of management support	6%
Insufficient training	4%

Respondents most frequently mentioned a failure to combine the right people (34%) or to train a team based values (28%), when they were asked about the organizational problems causing agile project failure.

Table 5. Specific Organizational Issues

Failure to integrate people	34%
Failure to teach team-based culture	28%
Communication between Dev/Product Owner	21%
Communication between Dev/QA	9%
Scrum Master problem	8%

5.7 Use and Preferences of Agile Tools

On an average, respondents are using standard office productivity tools such as Excel, which is followed by specialized and specific tools like Microsoft Project, Atlassian/JIRA and VersionOne.

Respondents were also asked whether they would recommend the tools they are using based on their present and past experience and use. VersionOne was highly recommended tool (93% of respondents recommend it), followed by Atlassian/JIRA (92%) and then the LeanKit (92%) tool.

Generally Excel tool was one of the tool which was used most commonly (69%), but near about 40% of the users would not suggest Excel as the most effective agile tool. When users using agile tools were asked how they will decide the appropriate time for using a tool, most commonly mentioned reasons were:

- (1) More collaboration are needed,
- (2) Roll-up metrics across projects/groups,
- (3) Co-located teams are not capable to notice physical boards

Table 6. Specific Agile Tools Used

Tools	Usage
Excel	69%
Microsoft Project	48%
VersionOne	36%
JIRA/Greenhopper	33%
HP Quality Center	24%
Bugzilla	22%

Respondents are currently using a wide range of agile tools, the most commonly used tools were Bug Trackers (83%), Wikis (72%) and Task boards (77%) followed by Unit test tools (66%), automated build tools (68%) and Spreadsheets (69%). Table 7 shows the most commonly used tools in agile methods.

Table 7. Most commonly used tools

Bug trackers	83%
wikis	72%
Task boards	77%
Unit test tool	66%
Automated build tools	68%
spreadsheets	69%

5.8 Conclusions and future work

Agile software development approaches are intended to increase efficiency, flexibility, agility and they are supposed to be more familiar to the atmosphere where software development projects are working together. To recapitulate main findings 90% of the respondents said implementing agile enhanced their ability to manage changing requirements. Still the scrum methodology is the most popular among all the agile methods, evidently 72% of respondents are using it in their teams. Apparently an interesting fact has been noticed

that software companies are in a benefit by developing smaller projects using agile methods. We have also observed that a number of software companies are not using a variety of agile methods because of lack of sound proficiency in this field. It is also found that not very significant amount of work has been published yet. It is therefore essential that empirical research and investigation should be carried out in this field. The paper analyses the current state of research with regard to agile methodologies.

The results of the study contribute to our understanding of how ASD methodologies are being implemented in the workplace. The follow up work in the above analysis is outlined below:

1. A study on the metrics used to compute quality, productivity, predictability and value delivery of agile teams is also needed.
2. Exploring agile tools: configuration and resource management, testing, project routing, user participation, agility design, virtual teams or others.
3. Scaling Agile: How agility can be adapted to work for large teams, it would be investigated.

6. ACKNOWLEDGEMENTS

We would like to express our greatest gratitude to the people who have helped and supported throughout work. Our thanks to the faculties of computer science department and experts who have contributed towards development of the template.

7. REFERENCES

- [1] Beck, K. *Extreme Programming Explained: Embrace Change*, Second ed. Reading, Mass.: Addison-Wesley, 2005.
- [2] Schwaber, K. and Beedle, M. *Agile Software Development with SCRUM*: Prentice-Hall, 2002.
- [3] Cockburn, A. *Crystal Clear: A Human-Powered Methodology for Small Teams*: Addison Wesley, 2004.
- [4] Beck, K. 2000 *Extreme Programming Explained: Embrace Change*.
- [5] Highsmith "Agile software development: the business of innovation", *IEEE Software*, 2001.

- [6] Beck, K. et al. (2001). "Manifesto for Agile Software Development". Agile Alliance. Retrieved 14 June 2010.
- [7] Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jefferies, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., and D. Thomas, D. 2001. Manifesto for agile software development. Available: agilemanifesto.org
- [8] Beck, K. "Principles behind the Agile Manifesto", Agile Alliance. Archived from original on 14 June 2010. Retrieved on 6 June 2010
- [9] Raising, L. and Janoff, N.S. 2000. The Scrum software development process for small teams. *IEEE software*, 17(4), 26-32.
- [10] Cockburn, A. *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley, 2004, ISBN 0-201-69947-8.
- [11] Beck, K. *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2000, ISBN 0-201-61641-6.
- [12] Kajko-Mattsson, M., Lewis, G.A., Siracusa, D., Nelson, T., Chapin, N., Heydt, M., Nocks, J., and Snee, H. "Long-term Life Cycle Impact of Agile Methodologies," *Software Maintenance*, 2006. *ICSM*, 06. 22nd IEEE International Conference on, vol., no., pp.422-425, Sept. 2006.
- [13] Miller, G. "Want a better software development process? Complement it," *IT Professional*, vol.5, no.5, pp. 49-51, Sept.-Oct. 2003.
- [14] Raising, L. and Janoff, N.S. 2000. The Scrum software development process for small teams. *IEEE software*, 17(4), 26-32.
- [15] Poppendieck, M. and Poppendieck, T. *Lean Software Development – An Agile Toolkit for Software Development Managers*, Addison-Wesley, Boston, 2003, ISBN 0-321-15078-3.