A CRITICAL ANALYSIS AND COMPARISON OF AGILE WITH TRADITIONAL SOFTWARE DEVELOPMENT PROCESS

Priyanka Kataria¹, Shweta Shrivas², A.Hemlata⁴
¹²⁴Department of Computer Science & Engineering, Jabalpur Engineering College
Ishtia Shukla³
³Department of Computer Science & Engineering, Manipal Institute of Technology

Abstract: During the past years, new software development approaches were introduced to suit within the new trend of the software development corporations. Most Software Corporation’s today aim to provide valuable software in short period of time with marginal prices and among unstable, ever-changing environments. Agile methodology focuses on the challenges of unpredictability of the real world by relying on individuals and their creative thinking instead of method. In this paper we tend to explore about the current agile methods, strengths and weaknesses of agile strategies and numerous problems with their relevancy. We have conjointly enclosed comparison between traditional software development process and agile software development process. This paper also includes brief discussion about the benefits and problems associated with these methodologies by performing case study of two corporations.

INTRODUCTION

In today scenario, software development has been expanding. Software has merged into many distinct areas, and is becoming more intricate as compare to old software development methodology. Changing requirements and customer satisfaction is making it even more difficult. Previous software development approaches are not able to satisfy the new requirements of the market in the best way, anymore. As a result, new software development has been approached, like agile methodologies, mainly to solve such kind of problem [3]. The new approaches include changes to software development processes, to make them more productive and flexible. This is not only suitable for software developers but also for Team Leaders, Project Managers, Product Managers, Development Managers, Testers, QA Managers, QA Engineers, Technical Writers, UX Designers, anyone involved in the delivering software. The main theme in agile methods is to promote and speed up response to changing environments, requirements and meeting the deadlines [4]. The agile manifesto [1] states the main focus of the agile development as the following:

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

There are a number of agile software development methods. Methods for agile software development represent a set of practices for software development that have been created by experienced people. The most common methods are extreme programming (XP), dynamic software development method (DSDM), scrum and crystal [1]. This paper is focuses on the agile software development, agile methods centered on current practices in industry. Most commonly used methods will examined from the angle of their applicability, strengths and weaknesses and their adoption in industry. In order to investigate and analyze, there is a need to compare why agile? So, we have compared traditional software development technique and agile software development technique.

RESEARCH OBJECTIVES

The objectives of this paper are:

- Examine, gain insight into the agile methods and practices.
- Find out the issues in where, when and how agile methods are used.
- Strengths and weaknesses of agile methods.
- Understanding transition from traditional methods to agile methods and
- Effects on the organization by case study
DIFFERENT AGILE METHODOLOGIES

Agile is a framework and there are a number of specific methods within the agile movement [7]. The different flavors of agile are:

**Extreme Programming (XP):** It is also known as XP, extreme programming is a type of software development intended to improve quality and responsiveness to evolving customer requirements. The key points of XP include feedback, assuming simplicity and embracing change.

**Scrum:** Scrum is one of the most popular ways to implement agile. It is an iterative software model that follows a set of roles, responsibilities, and meetings that never change. Sprints usually lasting one to two weeks, allow the team to deliver software on a regular basis [2].

**Feature-driven development (FDD):** This iterative and incremental software development process blends industry best practices into one approach. There are five basic activities in FDD: develop overall model, build feature list, plan by feature, design and build by feature.

**Adaptive system development (ASD):** Adaptive system development defines the idea in which projects should always be in a state of continuous adaptive state. ASD has a cycle of three repeating series: speculate, collaborate and learn.

**Dynamic Systems Development Method (DSDM):** This agile project delivery framework is used for developing software and non-IT solutions. It addresses the common failures of IT projects like going over budget, missing deadlines, and lack of user involvement [7]. The eight principles of DSDM are: focus on the business need, deliver on time, collaborate, never compromise quality, build incrementally from firm foundations, develop iteratively, communicate continuously and clearly and demonstrate control.

**Crystal Clear:** Crystal clear is part of the crystal family of methodologies. It can be used with teams of 6-8 developers and it focuses on the people not processes or artifacts. Crystal clear requires the following: frequent delivery of usable code to users, reflective improvement and osmotic communication preferably by being co-located. Crystal is actually comprised of a family of agile methodologies such as crystal clear, crystal yellow, crystal orange and others, whose unique characteristics are driven by several factors such as team size, system criticality and project priorities [3]. This crystal family addresses the realization that each project may require a slightly tailored set of policies, practices and processes in order to meet the project's unique characteristics.

Scrum is one of the important and useful agile methodologies discussed in detail below.

**SCRUM**

Scrum is a repetitive and progressive agile software development framework for managing software projects and merchandise or application development. Its focus is on “a versatile, holistic product development strategy where a development team works as a unit to achieve a standard goal” as against a “traditional, sequential approach” [5]. Scrum asks why it takes so long and so abundant effort to do stuff. We don’t know about how long and how abundant effort things can take. Scrum asks uncertainty and creative thinking. It places a structure around the learning method, enabling members of the team to assess both what they’ve created, and even as significantly, how they created it.

**Scrum Roles**

These are the essential roles for scrum success. A scrum team has a slightly different composition than a traditional software development project, with three specific roles: product owner, scrum master and the development team [5]. And because scrum teams are cross-functional, "the development team" includes testers, designers, and engineers in addition to developers.

**PRODUCT OWNER**

- The product owner represents the stakeholders and is the voice of the customer.
- Accountable for ensuring value to the business.
- Writes (or the team) customer-centric items (user stories), prioritizes them, and adds them to the product backlog.
- Scrum teams should have one, may also be a member of the development team.
- Not be combined with that of the scrum master.
DEVELOPMENT TEAM

- Responsible for delivering potentially shippable product increments at the end of each sprint.
- Made up of 3–9 people with cross-functional skills who do the actual work (analyze, design, develop, test, technical communication, document, etc.).
- Self-organizing, even though they may interface with project management organizations (PMOs).

SCRUMMASTER

- Accountable for removing impediments to the ability of the team to deliver the sprint goal/deliverables.
- Is not the team leader but acts as a buffer between the team and any distracting influences.
- Ensures that the scrum process is used as intended.
- Enforcer of rules. Protector of the team and keep it focused on the tasks at hand.
- Also been referred to as a servant-leader to reinforce these dual perspectives.
- Differs from a project manager in that the latter may have people management responsibilities unrelated to the role of scrum master.
- Excludes any such additional people responsibilities.

BACKLOG

PRODUCT BACKLOG

The product backlog is an ordered list of "requirements" that is maintained for a product. It consists of features, bug fixes, non-functional requirements, etc. - whatever needs to be done in order to successfully deliver a working software system [5]. In scrum, it is not required to start a project with a lengthy, upfront effort to document all requirements. This agile product backlog is almost always more than enough for a first sprint. The scrum product backlog is then allowed to grow and change as more is learned about the product and its customers.

SPRINT BACKLOG

The sprint backlog is the list of work the development team must address during the next sprint. The list is derived by selecting stories/features from the top of the product backlog until the development team feels it has enough work to fill the sprint. This is done by the development team asking "Can we also do this?" and adding stories/features to the sprint backlog [5].

Conceptually, the team starts at the top of the prioritized scrum backlog and draws a line after the lowest of the high-priority items they feel they can complete [6]. In practice, it is not unusual to see a team select, for example, the top five items and then two items from lower on the list that are associated with the initial five.
COMPARISON BETWEEN TRADITIONAL SOFTWARE AND AGILE SOFTWARE TECHNIQUE

A software system is built in such a way that it can perform complex tasks and computations on the command of the user. The process of building software requires a rigorous attention to detail and a general guiding algorithm. The traditional and agile processes essentially are platform software development ideologies, more than anything else. They can be further subdivided into more specific processes that form the basic plan of each unique software development life cycle. The agile methodology post-dates the traditional one in the evolution of the software development processes.

The modern software development life cycle methodology can be subdivided into two types – the traditional process and the agile process [8]. The comparison between both the processes is shown in table 1.

Table 1. Comparison between Traditional Software Development and Agile Software Development

<table>
<thead>
<tr>
<th>S. No.</th>
<th>PROPERTIES</th>
<th>TRADITIONAL SOFTWARE DEVELOPMENT</th>
<th>AGILE SOFTWARE DEVELOPMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Development Process</td>
<td>It is a structured process, where you can’t start on a new phase until the previous one has been completed.</td>
<td>Agile is a flexible process, allowing you to move through the project as you like.</td>
</tr>
<tr>
<td>2</td>
<td>Linearity</td>
<td>It is sequential.</td>
<td>Agile does not enforce a linear process.</td>
</tr>
<tr>
<td>3</td>
<td>Requirements Specification</td>
<td>These projects usually include defined requirement in advance.</td>
<td>Requirements are expected to change and evolve in Agile projects.</td>
</tr>
<tr>
<td>4</td>
<td>Consistency</td>
<td>you can’t change things that were done in previous stages,</td>
<td>Agile is very accommodating to changes.</td>
</tr>
<tr>
<td>5</td>
<td>Time Boxing Period</td>
<td>Time limit could be increased.</td>
<td>Rapid Deployment</td>
</tr>
<tr>
<td>6</td>
<td>Continually Evolving</td>
<td>Developer met once or twice per week with other team members in roughly half-hour or hour-long slots. These problems were compounded by a lack of cohesion among our team members.</td>
<td>Methodology has several elements but the central one is a 15-minute, daily, all-hands meeting (called the Scrum) in which developers update everyone in the group.</td>
</tr>
</tbody>
</table>

CASE STUDY

The benefits and problems of successful adoption of agile methodologies with scrum methodology are shown below:

DISTRIBUTED SCRUM PROJECT FOR DUTCH RAILWAYS

The Dutch railways are among the most heavily used in the world, providing transport for 1.2 million passengers daily. Dutch railways built a new information system to provide travelers with more accurate travel information, requiring less manual intervention. As part of this program, they built the PUB (publish) system that centrally controls information displays and audio broadcast systems in all stations.

The first attempt to build the PUB system was executed using a traditional software development approach. Detailed requirement specifications were handed over to the IT vendor, expecting a fully built system to materialize without much further customer involvement. After 3 years, the project was cancelled because the vendor failed to deliver a working system. The customer then engaged their company to build the PUB system from scratch. They introduced an agile approach using scrum, focusing on close cooperation with the customer, open communication and working in small increments. The outcomes of agile implementation project in Dutch railways were higher rate of customer satisfaction.
AGILE CASE STUDY – H&R BLOCK

CHALLENGE

The Information Systems department at H&R block had been following the traditional “waterfall” methodology for over 10 years. With an average of 20+ active projects and increasing pressure from both internal and external stakeholders, their workload was always growing. The CIO and his team began to look for alternate, more efficient ways to get software out the door.

SOLUTION

In 2011, H&R block decided that the lightweight processes of the agile/scrum methodology met their needs for improved efficiency and effectiveness. However, they needed help with everything from getting started to full roll-out. Then they partnered with Braintrust Consulting Group (BCG). Their team of expert trainers and coaches built a custom implementation plan to meet the specific needs of H&R block.

CUSTOM IMPLEMENTATION PLAN

For H&R block, their implementation plan began with an evaluation. They assessed the types of projects that they had in their portfolio, they evaluated staff capabilities, and built the overall implementation strategy for agile/scrum. The plan began with training. The certified scrum trainer’s at Braintrust held a series of training classes over a period of weeks to ensure that every team member was comfortable with the basic principles of Scrum. For H&R block, BCG executed a form of coaching called embedded coaching. BCG expert coach was on-site with the H&R block team to work with them daily; participating in sprint planning, sitting in on daily meetings, and guiding the team while mentoring the scrum master (SM) and the product owner (PO). Once the teams felt comfortable with the process, BCG moved to the final phase of the plan: coaching support. Under this phase, BCG provided on-going support and specified training for 12 months.

OUTCOMES

The outcomes of our agile implementation project with H&R block were both quantitative and qualitative are listed below:

- Better employee engagement
- Higher employee moral
- Higher development success ratios
- Greater ROI on products
- Increased customer satisfaction and time-to-market delivery
- Improved communication
- Greater flexibility

CONCLUSION

Agile methodologies offer some practices that facilitate communication between the developer and also the client, and undergo develop-deliver-feedback cycles, to possess additional specific view of the necessities, and be prepared for any modification at any time. The main aim of agile methodologies is to deliver what is required when it is required. Through our analysis, we discover out that with most projects, the desired functionality, time and budget are shifted because the project progressed, creating "on time, on budget", a obscure measure of accomplishment.

The main agile methodologies that are being employed is scrum methodology. The developers involved in the study mainly stressed the subsequent four reasons for adopting agile methods: ability to alter, short time frames of releases, continuous feedback from customers, high-quality and bug free software.

In scrum, once the sprints are known and assigned to the team members, they need to be stable because they're frozen. No modifications are allowed till the completion of the development of that sprint. Adding of recent sprints within the middle of the development is not possible. In scrum, team size is restricted to seven. Scrum will support the distributed development. This paper performs the critical analysis of software development process and compares the agile methodology with traditional methodology.
REFERENCES


