

Transition from conventional to Agile process model An Experience Report

Abhinav¹, Vijayalakshmi M. ², Achyut Bhandiwad³, Karthik Mellikeri⁴, Preetham Nagesh⁵

^{1 2 3 4 5} Computer Science Engineering Department, B V Bhoomaraddi College of Engineering and Technology
Hubli - 580031, Karnataka, India

¹abhinavp.klg@gmail.com

²viju11@bvb.edu

³achyut.b6@gmail.com

⁴hmellikeri@gmail.com

⁵preethamnagesh8@gmail.com

Abstract - Software process standards and models provide best set of practices and guidelines to improve the quality of the software process and products resulting from that process. These software models however, do not advise following a particular software development methodology, and consequently, software development teams face problems with respect to compliance between the selected process development strategy and the one actually pursued. In this report, the current software process models used in academia and their transition to newer, more advanced models are studied. The main focus although, is on transition from the hugely popular process model followed in academia, Waterfall model to Agile model.

Keywords - Software process models, traditional methods, agile methods, transition to agile.

1. INTRODUCTION

Over the past decades, many different software development processes have been designed and utilized in the software industry. Each one of them have different characteristics and methodologies and are distinct from others. On a broader sense, they can be divided into two, one usually called the traditional method, usually focus on comprehensive planning, heavy documentation, and big design up-front. The other method is a lightweight method also known as agile methods, focuses on how individuals work and how they interact rather than kind of tools and methods used, agile method concentrates on working software rather than comprehensive documentation, it values customer collaboration more than contract negotiation. The priority is given to responding to change over following a plan.

Because of the structured, straightforward and methodical nature traditional methods are widely used in the software industry; they have proved that they can provide high certainty, stability and predictability. However, they have a number of key shortcomings, including slow adaptation to constantly changing business requirements and a tendency to be over budget and/or behind schedule, delivering fewer features and functions than specified in the requirements. A few shortcomings may be expected using the traditional methods in the form of lower adaptation rates to constantly changing business requirements, has an inclination to not being budget friendly, being behind schedule and delivering

only a subset of all the features as specified in the requirements. The need for a complete set of requirements prior to design is also a major challenge for the traditional methods due to vague user specifications.

New software development models have come up as a remedy for the shortcomings of the traditional approach. These new models also called Agile models are designed to embrace high rates of change rather than rejecting it. Agile software development models include Scrum, eXtreme Programming (XP), Crystal and Adaptive Software Development (ASD). The focus, with these new models shifts towards iterative or incremental development, customer collaboration and frequent product delivery through a light and fast development cycle. As claimed by researchers all over the world, agile methods of software development provide higher levels of customer satisfaction, lower error rates, shorter and more precise development cycles and quick adaptation to changing business requirements.

In spite of these potential benefits of agile methods, many organizations are reluctant to make a transition into agile methods. This reluctance is the result of several issues, including

- Agile methods significantly reduce the documentation work and instead rely on the tactics of the organization.
- These practices haven't been tested rigorously on critical projects.
- A common belief that these methods are not adequate for highly stable projects.
- A concern that the outcome depends upon decisions taken by certain individuals

This report aims at doing the following:

- Describe the current software development processes in practice along with their advantages.
- Explain the advantages of agile practices.
- Issues in transition from traditional to agile methodologies.
- The problems related to agile practices and how to overcome them.

2. SOFTWARE PROCESS MODELS

A software process is a set of related activities that leads to the production of the software. These activities may involve the development of the software from the scratch or modifying an existing system.

Any software process must include the following four activities:

- Software specification: Define the main functionalities of

the software and the constraints around them.

- Software design and implementation: The software is to be designed and programmed.
- Software verification and validation: The software must conform to its specification and meets the customer needs.
- Software evolution: The software is being modified to meet customer and market requirements changes.

Software process is complex, it relies on making decisions. There is no ideal process and most organizations have developed their own software process. Some of the most popular software development process models :

A. Waterfall Model: A sequential approach wherein each fundamental activity is represented as a separate phase to be executed in a linear order and strictly one after the other. Requirements collection, Designing a solution, Implementing the designed solution, testing it and maintenance are some of the most common and widely used phases. These phases are to be executed in a series and once a phase is done, no further modifications are allowed on it and has to be frozen. This model offers no flexibility and any change requested by a client will have to be declined by the development team.

B. Prototyping Model :A prototype is a crude version of the proposed system that is developed quickly considering only the most important features to be visually displayed to a client. Prototype approach is followed when the client is unsure of the requirements, of algorithms, efficiency, how it performs and so on. Client involvement throughout the phases increases the likelihood of client satisfaction.

C. Incremental development : It is a software development model where an initial implementation is developed. The initial implementation contains the most important features which are of higher priority. The user feedback is taken for the implemented functionalities and specified changes are done. This process is repeated by incrementing the functionalities. This helps the user have a brief idea about the functionalities and he can verify if it's a viable idea.

D. Iterative Model: Iterative development model aims to develop a system through building small portions of all the features, across all components. Building a product which meets the initial scope and releasing it quickly for customer feedback. An early version with limited features important to establish market and get customer feedback. In each increment, a slice of system features is delivered, passing through the requirements till the deployment. The phases of iterative development are:

- Inception*: The goal is to establish a business case for the system, identify all the external entities that will interact with the system, and define these interactions.
- Elaboration*: Good understanding of the problem domain and the architecture framework is developed. The project plan is developed and associated risks are identified.
- Construction*: The architecture is step by step completed by constructing production ready code produced from analysis, design, implementation and testing of the requirements. Some of the architectural components may be independent of each other and can

be implemented in parallel and then be integrated together.

d. *Transition*: The system is delivered into production environment.

E. Agile Model: Agile is a flexible, dynamic model that can be adapted to specific circumstances. The agile methods refers to a group of software development models based on the incremental and iterative approach, in which the increments are small and typically, new releases of the system are created and made available to customers every few weeks. Customer involvement is necessary so as to dynamically change any requirement during the development phase. They minimize documentation by using informal communications rather than formal meetings with written documents. They are best suited for application where the requirements change rapidly during the development process. There are a number of different agile methods available such as: Scrum, Crystal, Agile Modeling (AM), Extreme Programming (XP), etc. The below diagram depicts the key principles that guide agile practices.

3. LITERATURE SURVEY

Software development process works with deadlines for completion of tasks at hand. Various strategies for achieving these tasks are devised. We have discussed popular software development models being 'Waterfall' & 'Agile'.

Waterfall model in software engineering was introduced formally as a notion in a paper published by Winston Royce in 1970. However, it had its shortcomings as identified by himself. Nevertheless, it was preferred as a development model by many companies. The philosophy of Waterfall model was derived from the hardware manufacturing strategies and construction strategies that were in practice during the 1970s. Hence it was noted to be a structured approach for software development.

Meanwhile, the agile model of software development emerged in the 1990s, when developers opted to drift away from the segmented, structured, bureaucratic ways to software development and inclined towards more compatible development styles. The Agile methods as they were formally defined by Edmonds in his research in 1974. More prominent and popular agile models for development, that evolved subsequently, are 'Scrum' in 1995, 'Adaptive Software Development', 'Dynamic Systems Development Method' in 1995, 'Crystal Clear', 'Extreme Programming' in 1996, and 'Feature Driven Development'. In 2001, the 'Agile manifesto' was declared by a group of pioneers in agile software development, which is a set of canonical rules for agile software development methods.

4. PREVIOUS EXPERIENCE WITH ACADEMIC PROJECTS

The Projects done by the students during previous semesters adopted the waterfall methodology which required them to firstly find the problems and requirements for the project. The students are asked to present these requirements during the initial phase of the project. Students need to have a clear understanding of the project requirements since it cannot be changed during the further phases of the project. In this way the frozen requirements are adopted in the next phases of the project. The next phase being Design phase has to be done carefully based on the requirements. The implementation

phase completely depends on the previous phases. If during any phase any requirement is to be changed it can lead to chaos since it may change the design of the project.

Whereas the students who undertook industrial projects during their final semester adopted the industry popular Agile model in which it is not compulsory to freeze the requirements in the initial phase of the project. Students have liberty to make necessary changes in the design phase. There are daily meetings and brainstorming sessions which can help students get better view of the project. Teams from different sections can have meetings together to help understand and build upon their knowledge. Each individual member of the project maintains a scrum sheet containing the details regarding the work they have done each day, the problem that might have occurred while solving the problem and the task they intend to complete the next day.

Thus we can say that the agile model approach adopted by the students in their industry project benefits more compared to waterfall model approach adopted by most of the students doing their student project.

5. TRANSITION FROM CONVENTIONAL TO AGILE METHODOLOGY

Companies use majority of their resources for development and therefore their success or failure influences the company's capability to do well in the industry. Projects generally endure from issues such as task priorities, narrow deadlines, short of appropriate coordination amongst the scattered teams along with quickly varying market situation that result in changing requirements. The three essential stages in a development process are requirements analysis, development and testing which in a conventional process goes from a analyst to a development team followed by a testing team with every team accountable only for their part. A lot many coordination issues and confusions come up with the raise in complexity and magnitude of projects. Here is where the Agile methodology for development shines over conventional development model. Agile development suggests a tightly bound cross functional teams where the analyst, development and testing team with product owner are accountable for the complete product development cycle.

Agile teams are more productive than conventional teams because teams function simultaneously on different modules and they bring product functionality incrementally so the organization need not wait till the end for a delivery.

Team meets regularly throughout agile process such as planning, daily status, review and retrospective having an appropriate objective for every meeting.

Agile incremental development implements methods such as constant integration, Test driven development, Pair programming which leads to early detection of flaws and therefore enhance quality at the end of every iteration.

a) *Transitioning from waterfall to agile:*

Waterfall model is a rigid, top down development process which works in series of phases that start with gathering requirements and ends with testing. While it works well for straightforward projects, it doesn't work well for complex projects which meets the industry requirements. Transitioning from waterfall model to a agile model requires a few steps that can be followed:

- Agile is a methodology with many benefits so, motivate

your stakeholders to welcome the change.

- Agile transformation needs training so that the participants understand the agile principles and practices.
- Communication is important in Agile, sprint planning, daily stand-ups, retrospective meetings need communication skills. This includes verbal, written and tool based methods of presentation.
- Creating an environment where teams can collaborate with each other.

Using the above steps can prove to be helpful in a smooth transition from other conventional models of development to Agile development.

b) *Academic-Industry collaboration*

One of the reasons for the Academics - industry Collaboration is to produce students with work-based skills and to provide students exposure to industry standards. Academics - industry collaboration can be achieved in several ways - getting involved in project development is one of the most effective ways. Collaboration includes mentorship from industry professionals and hands on experience on some latest technologies. Students are involved in learning, developing and functioning as a onboard team member with such collaborations. On the other hand, this works for the industry since they get interested students with fresh ideas to work with them. This collaboration has been helpful to both the partners at various levels. Students are aware of the need of additional skill and attributes required for career success and view work experience as a means of achieving this.

6. PROPOSED METHOD

Our study mainly focuses of implementation of agile methodology using SCRUM in the academia. The curriculum has been designed such that it helps students learn the software process methodology followed by projects in collaboration with industry to get hands on experience with industry practices which is required for the students, in order to train them for the real world.

The aim is to analyze the results of Agile practices in academia and understanding how it deals with students and the development process. This research takes a qualitative approach including surveys with personal experiences from students working in collaboration with companies on their projects. This paper holds comparative results of students experiences with various software development methodology used in their prior projects during their academic curriculum.

Most of the companies in the industry today are moving towards Agility. Adopting agile methods improves the management of development process and increases the efficiency. The inclusion of Agile practices in academic software development process in a project is achieved by analyzing Scrum technique and know how the process works. In this research we have made an attempt to justify how scrum fits into the academia. A survey was conducted among the students working on projects with the industry to verify whether the practices increased the efficiency of project development and their learning experience. Scrum is a development process that can be used for development of any kind of project. Its values are based on Agile development methods. All tasks under a project occurs within a process

pattern called Sprints. The work under each sprint. The work under each sprint depends upon the priority and deadline of the objective.

In our research we are including a case study of a project incorporating the scrum methodology under development with a company in collaboration with academics. This shows how the students work on their projects efficiently and in turn get trained and equipped with developmental tools and practices in a company, increasing the rate of project success in a software development environment.

a) Case Study

The collaboration of industry based project in the academia has provided students with opportunity to understand and learn industry standards of development. The project at hand is an ongoing project which has had multiple prior iterations by various teams who worked on it, further developmental iterations are based on the feedback collected from our the End-Users, customers, teams and other stakeholders useful for preparing the scope of work for the currently working teams. Scope of work contains the product backlog from the previous iteration and the latest feedback received. This Scope of work is divided into modules which is provided to teams. With the formation of teams the roles of Product owner, Scrum master and team members are defined. Teams decide and allot meeting schedules for the discussions of scope of work and any clarification required with the Product owner. These are the Sprint Planning Meetings where the teams decide about the commitments to do by the Sprint's end. 4-week sprints are defined with deadlines. A Scrum Sheet [fig(2)] for the whole project is maintained along with the Gantt chart [Fig(3)] and burndown chart. The Scrum sheet is updated on a regular basis with accomplished tasks, task backlogs and issues. The Project is divided into a planned number of sprints with delivery deadlines and status of the ongoing sprint. Deliverables may change priority with requirement at times hence some flexibility on deadlines is maintained. Since a single project has multiple interdependent modules, teams are cross-functional. Daily Stand-up meeting are done to discuss ongoing agenda and progress.

master All deliverables after completion and testing are integrated to the main module and shipped to the product owner for the real world testing and deployment.

The use of various tools under the Agile-Scrum methodology for software development have proved to be effective in achieving deliverables in a saleable product even while the project is in full swing. This approach shortens the completion time and the backlogs completed are added onto the product as new or modified features. Reviewing every sprint before starting on with the next one ensures testing is conducted throughout the process which allows changing the scope or direction of project at any point by the development team. The project requirements are not frozen but the deadlines are. In fact, stakeholders anticipate changes along the way. The product owner's involvement in the project management process facilitates these changes.

	At Risk	Task Name	Status	Feature Type	St... P...	Start	Finish	Pr...	Durati...	Spr...
1	Phase 1	In Progress	Enhance/Dev	45	06/04/17	07/10/17			27d	
2	Develo	Completed	Study	45	06/04/17	07/10/17			27d	
3	Stud	Completed	Study	5	06/04/17	06/08/17			5d	
4	Stud	Completed	Study	5	06/09/17	06/14/17			4d	
5	Stud	Completed	Study	5	06/15/17	06/20/17			4d	
6	Stud	Completed	Study	5	06/21/17	06/27/17			5d	
7	Imple	Completed	Development	20	06/28/17	07/05/17			6d	
8	Integ	Completed	Testing	5	07/06/17	07/10/17			3d	
9	Phase 2	In Progress	Development	70	07/11/17	08/23/17			32d	
10	Design	In Progress	Development	80	07/11/17	08/23/17			32d	
11	Discu	Completed	Study	8	07/11/17	07/14/17			4d	
12	Maki	Completed	Enhancemen	12	07/15/17	07/20/17			5d	
13	Imple	Completed	Development	9	07/15/17	07/20/17			5d	
14	Imple	Completed	Development	17	07/29/17	08/10/17			10d	
15	Displ	Completed	Development	8	07/21/17	07/24/17			2d	
16	Color	Completed	Enhancemen	5	07/25/17	07/28/17			4d	
17	Deve	In Progress	Development	12	08/12/17	08/23/17			9d	
18	Integ	In Progress	Testing	9	08/15/17	08/23/17			7d	
19	Phase 3	Not Started	Enhance/Re-	100	09/01/17	10/13/17			31d	
20	Redesi	Not yet Star	Re Design	80	09/01/17	10/13/17			31d	
21	Enha	Not yet Star	Enhance	30	09/01/17	09/12/17			8d	
22	Optin	Not yet Star	Enhance	40	09/13/17	10/13/17			23d	
23	Optin	Not yet Star	Enhance	30	10/01/17	10/13/17			11d	
24	Phase 4	Not Started	Development	50	10/15/17	11/30/17			35d	
25	Develo	Not yet Star	Development	50	10/15/17	11/30/17			35d	
26	Desic	Not yet Star	Development	15	10/15/17	10/25/17			9d	
27	Admi	Not yet Star	Development	8	10/25/17	10/31/17			5d	
28	Enha	Not yet Star	Enhance	12	11/01/17	11/15/17			11d	
29	Enha	Not yet Star	Enhance	15	11/15/17	11/30/17			12d	
30	Phase 5	Not yet Star	Development	60	12/01/17	12/29/17			21d	

Fig. 2 Scrum Sheet



Fig. 1 Scrum Board

To keep a track of backlog a physical scrum board [Fig(1)] is maintained with sticky notes. A review meeting is conducted by the scrum master after each sprint to review completed objectives and discuss backlogs if any. Scrum

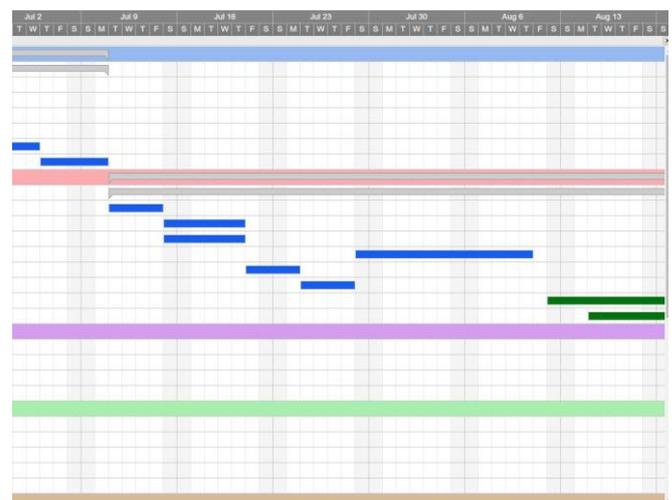


Fig. 3 Gantt Chart

7. DISCUSSIONS

Members of the teams participated in discussions on

software development process models. This discussion highlighted major points of Agile process model with respect to these teams and their prior experience with other process models.

The highlights to the discussion are as follows:

1. Agile Project lifecycle is segregated as Sprints and helps managing time as a resource.
2. Waterfall model often turns out to be a rigid structure whereas Agile model is quite flexible.
3. In case of Waterfall model, software development is to be completed as one single project, which is divided into different phases, with each phase appearing only once during the SDLC. However, the Agile methodology can be considered as a collection of many different projects, which are nothing but the iterations of the different phases focusing on improving the overall software quality with feedbacks from users or the testing team
4. In the Waterfall model, the “Testing” phase comes after the “Build” phase, but, in the Agile methodology, testing is typically performed concurrently with programming or at least in the same iteration as programming.
5. Waterfall methodology does not require the participation of customers, the Agile software development approach focuses on customer satisfaction and thus, involves the participation of customers throughout the development phase
6. The Agile methodology is a highly collaborative software development process, thereby leading to better team input and faster problem solving.

8. SURVEY RESULT ANALYSIS

A survey was conducted among the students who have been working on the industrial project with the companies to gather their experiences on Agile practices. These students have worked on various projects with other software development models hence, they we have a comparative response in the survey.

A few survey results are provided as follows:

Learned and practiced Agile development Process for:

8 responses

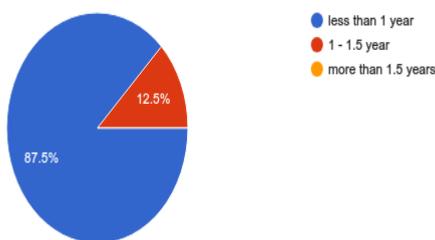


Fig. 4 Learned and Practiced Agile development Process

The inclusion of Software engineering and industrial project in the curriculum has made students experienced with these practices. This [Fig(4)] data clearly shows that most of the students have quite recently started with Agile development process.

What type of development practices have you used in projects other than industry project

8 responses

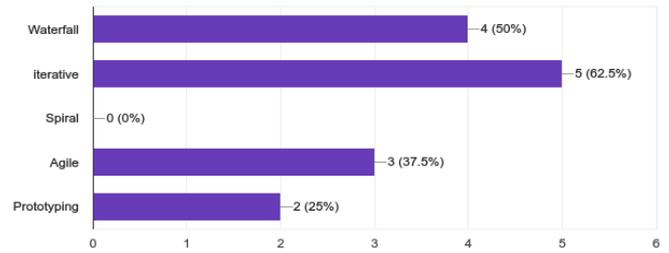


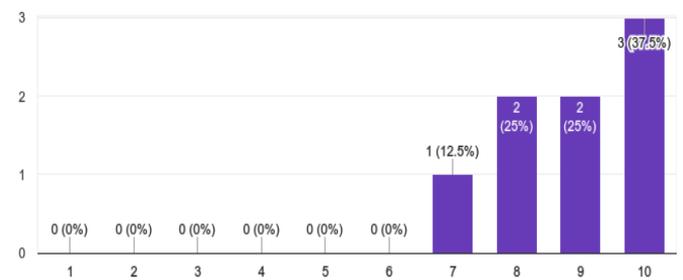
Fig.5 Survey results: Process models previously used

The students have previously worked on project other than the industry project with the above mentioned process models. Fig(5) clearly shows iterative model to be highly productive for these students followed by waterfall model. Agile offers as a composition of various features from both iterative and waterfall model.

The learning experience of these students have been remarkable and the level of agreement to the inclusion of industry project in the curriculum is shown in Fig(6).

Level of agreement on collaborating with industry for projects in your curriculum

8 responses



Fig(6)Survey results: Agreement on Industry collaboration

Use of Agile process model have proved to be effective in project completion and managing various resources of the team.

9. LEARNING EXPERIENCE

The students working on these projects have shared their learning about the demands of the IT industry in the field of development. Acknowledging the Academic-Industry collaboration project provided by our Institution. A few learnings are mentioned:

- A. Learned how the industry works, manages teams, their tasks and meets customer requirements.
 - B. Implementing challenging tasks and managing their deadlines.
 - C. Work environment of a fast paced company.
 - D. Learning from other teams and collaboration towards project completion.
- Learning new technologies with market needs.

REFERENCES

1. A Case Study on Software Development Projects in Academic Knowledge Centers using SCRUM