

AGILE SOFTWARE DEVELOPMENT AND CHALLENGES

Kiran Jammalamadaka¹, V Rama Krishna²

1 Research Scholar, Computer Science, K L University, Andhra Pradesh, India, JVSKKIRAN@GMAIL.COM

2 Asst. Professor, Computer Science, K L University, Andhra Pradesh, India, VRAMAKRISHNA@KLUNIVERSITY.IN

Abstract

Loyal and steady customer base alone can keep the organizations successful in the current turbulent business environment. In the current era of software engineering, the success of a business process is measured in terms of 'customer satisfaction' rather than any other criteria like meeting deadlines for delivery, optimization of data, architecture etc. Day by day, customers are turning out to be more demanding, as their expectations from the software are growing. In order to achieve customer satisfaction in a meaningful way, software engineers are looking for more effective development models. "Agile" is one such model, that fits the bill and therefore industry is looking at with interest. Is agile better than traditional waterfall model will agile work effectively with distributed teams which is most common in the current software engineering Phenomenon.

This paper highlights a few challenges with Agile->scrum and gives an insight to the user whether the agile is THE SILVER BULLET

Index Terms: waterfall, Agile, Scrum, XP, distributed teams

1. INTRODUCTION

Broadly, software development models can be classified into two major buckets [2]

1. Predictive
2. Adaptive

For considerable time, majority of the industry players are using waterfall model, a major base for predictive models. However, the said model has its own drawbacks in terms of delivering the business process to the customer. As the name suggests, the waterfall model is based upon a few assumptions like clear understanding of the requirements (which is getting nearly impossible in the current situation). The size as well as complexity of business process is drastically increasing in tune with the continuously evolving software industry.

However, here are a few pitfalls of the waterfall model.

A few pitfalls of waterfall model

Requirements need to be frozen upfront [16]:

Most of the projects fail at the 'problem defining stage'. More clarity towards 'what is required'. Requirement is the base and almost a starting point for many projects. Most of the defect origins may reside at this stage, however project manager or the processes are not only responsible for this stage and there could be various other reasons like

- Customers themselves are not clear about what they expect from the software

- Customers may not have been consulted frequently enough for tracking the project development and are not made a party to the interim testing
- Customers may fail to visualize the whole business process until completion.

However customer is the paymaster. Proactively foreseeing probable deviations rather than blaming them would pay towards quality long-term relationship.

Rigid System: [10] As customer realizes the ideal business process after completion. The most observed scenario is the customer may add more or alter the requirements as the time progress as Customer may find new needs, but even a simple change request from the customer may require a fundamental redesign of the system and sometimes may cause severe 'bugs'. This is becoming a major hurdle for customer to cope up with the current trends of concerned market.

Predicting/foreseeing all problems upfront: In waterfall model the project manager has to predict/foresee all the risks and problems upfront that are going to be emerged in real business process environment which is not practical, more over causes firefighting, schedule over runs and stretching which in turn cause serious quality concerns

Heavyweight: The bureaucratic nature of the waterfall model is frequently criticized. It is often observed that there were so many things to be attended, which do not directly affect the project more over slow down the pace of the project.

Though, there may be many reasons that contribute to the failure of the project, undoubtedly the above reasons are major ones.

Here is some such data collected from [11]

- \$80 -145 billion per year is spent on failed and cancelled projects (The Standish Group International, Inc.)
- 25% - 40% of all spending on projects is wasted as a result of re-work (Carnegie Mellon)
- 50% are rolled back out of production (Gartner)
- 40% of problems are found by end users (Gartner)
- Poorly defined applications led to a persistent miscommunication between business and IT. This contributes to a 66% project failure rate for these applications, costing U.S. businesses at least \$30 billion every year (Forrester Research)
- 60% - 80% of project failures can be attributed directly to poor requirements gathering, analysis, and management (Meta Group)
- Nearly two thirds of all IT projects fail or run into trouble.

Addressing such concerns has become a primary goal for any organization. However, if the predictive models are somewhat ineffective then it's a good time to see at other developing models, which are cost effective and promise on time and qualitative in nature.

For various reasons traditional development methods are not always feasible for all kinds of projects in this rapidly changing business environment. However "Agile" can be claimed as an alternative, which was proposed by Kent Beck and team

Agile is a set of software development methodologies, which are fundamentally based on some common principles [1], [3], Agile is to have frequent inspection and adaptation in line with customer expectations and needs.

Agile is based on a few principles detailed below.

- KIS [Keep it simple] approach
- Customer satisfaction
- Working software is the measure of progress
- Working software is delivered in frequent intervals
- Late changes in requirements can be accommodated
- Face to face communication with in teams
- Self organizing teams
- Attention to technical excellence and good design
- Adoption to change

- Projects are built around motivated individuals.

*Mentioned only a few

Agile Flavors

Today's most of the agile projects fall in any of the following flavors.

*Mentioned only a few

Extreme programming (XP)

This is one of the most commonly used methods across the world. XP is basically based on these values [8].

1. Communication
2. Simplicity
3. Feedback
4. Courage

Customers will be providing their requirements in the form of user stories, which are basically non-technical write-ups. These user stories will further break up into tasks.

The development will take place in small iterations, in each of the iteration a bunch of tasks will be addressed, developers will be working in pairs and performing unit tests and frequent code integrations. Ideally, Customers will be involved closely for a rapid feedback recommendation. Testing will be done in the form of unit tests and acceptance tests.

Scrum

Scrum is another popular development method The major activities in this method are [12].

1. Sprint Planning
2. Sprint Review
3. Scrum Meeting

'Product backlog' which is a list of features prioritized by the product owner from which the team will be picking up the activities for each iteration, Scrum master facilitates daily Sprint meetings with Product owner and team, where in each individuals will be explaining the status of their current task and what they are going to do next day

A Sprint is usually spans from 2 –4 weeks, at the end of each sprint, team should have working software

Dynamic System Development Method

The DSDM method is an extension of Rapid Application development. The DSDM will emphasize on continues

customer involvement. DSDM are more suitable method for those projects, which are on tight schedules and budgets. The estimations will be based on functionality rather than the Lines of Code [8].

DSDM Project life cycle has 4 stages

1. Study
2. Functional Model Iteration
3. Design and Build
4. Implementation

DSDM provides incremental iterative approach and also provides some core techniques called Time Boxing, Prototyping, Testing, Workshop etc.,

The main objective of the DSDM method is to keep the project in control of budget and timelines

Crystal Methodologies

Crystal family of methodologies was started before agile manifesto and its one of the founder of agile methodologies. It's a family of methodologies. [9].

All Crystal methods have 3 Priorities and 3 Properties

Priorities:

1. Safety
2. Efficiency
3. Habitability

Properties:

1. Frequent Delivery
2. Reflective Improvement
3. Close Communication

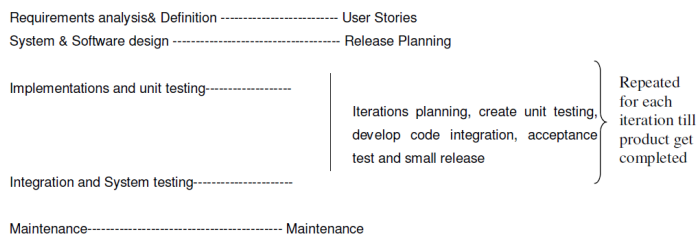
Crystal is a family of Human-Powered, Adaptive, Ultra-light and Stretch-to-fit methodologies

Crystal is more focused on People rather than **process or its architecture**

Agile & Waterfall model

The stages of waterfall model can be mapped to agile development model [15].

The stages of waterfall model can be mapped to Agile development model,



Challenges in Agile

In The 2013 State of Agile Development survey [13]

, conducted by VersionOne says 83% of the responders are using agile software development, however Agile may not be the silver bullet or the destination for the software development it has his own challenges which may force to re think to adopt agile as a software development methodology In the current phenomenon of software engineering, most of the companies have distributed environment for their Project execution or Product development, a typical scenario would be Designing will be at western countries, Architecture would be at Europe, Test automation and product development at India and testing would be at China, it's a typical scenario we observe the whole project spread across the world

Note: The above-mentioned scenario is just an example. Anything can happen anywhere in the world however. It spreads across the world

In a recent survey conducted by Version one 57% stated that their teams are distributed [13], and the figure is growing as industry considers distributed development as a cost effective model

The major challenges with Distributed environments are

Communication:

[5].In any project/product development communication is the main medium to connect all the stakeholders including customer and keeping everyone on the same page.

Agile insists about co location and Daily scrum of the team members to avoid any communication gap, however in distributed environment its not feasible to co locate all the stake holders at one place because of time zones and Language barriers.

A workable solution would be agree upon a common time slot for having scrum calls with video conferencing, and use instant messengers to communicate with colleagues, however we team need to be very planned to sort out issues soon after scrum having a technical discussion individually, to achieve this one needs a good planning skills

However it's very highly uncomfortable to have technical discussions over the phone and video conferences.

Agile is for highly Motivated and Skilled individuals:

One of the pitfalls we hear about Agile is, it is meant for skilled and motivated individuals. In Agile, work assignment is not done as a separate process, team proactively picks the tasks from the backlog and no control over the estimation. Trust plays a major role in it, however it may not be practical for all. Un like in waterfall no Sr. technical person review the estimation of the tasks, as the back log comes very fresh no one could spend time to understand the depth so to be on safe

side its always been estimated high, understanding the philosophy of agile is also very important, I would like to quote an example,

In one organization, where the team recently adopted the Agile scrum, after some time they are referring to an spread sheet for tracking bugs, I was surprised to see why they dependent on a spread sheet where as a sophisticated bug tracking tool is available, when I enquired about it they answered “we are using agile so we won’t log defects in bug tracking tool”, then I understood that they want to address the defects soon they were identified, as the defects list is piling up team started forgetting the defects and they were worried that the defects may slip to customers, so to avoid that they are tracking the defects in a spread sheet, its sounds funny but it’s the ground reality

Reverse Planning:

When team recently adopts the agile, one observation was that the same team used to spend long hours to finish their tasks, soon after they got trained on the agile, team left on time, no one suggests to stretch or stay long hours, however the point is team is tend to be more relaxed, surprisingly the same team will do night outs to finish the task on time as the higher management committed a date.

Though the above differs from the agile principles, it happens in the reality especially for legacy projects, higher management announces the date to the public or to the customers, so team starts reverse planning to meet the date as the features and time is already committed.

Flexible Scope:

Agile talks about continues working bug free software delivery to the customer with a demo, since the customer starts using and comes up with more ideas to be implemented which increases the scope of the development though it’s a trivial change from the customer perspective it could be a vital from technical perspective, in such scenarios scrum master should stop the change in the scope, however most of the projects are customer driven projects for various reasons and to make customer happy always team had to accept the change without moving the deadline

Agile is known for accepting the requirements at any stage, this cannot be the ground reality though it’s a software change is not a small thing to accommodate

Very difficult to estimate the efforts required which is very critical for budgeting, as no one knows about the whole requirements, more often after starting the tasks- the team will identify more tasks to be in order to finish the task which will affect the estimates

Involving customer in daily meeting helps a lot. However, It ignites more ideas to customer, though it is good it increases the scope of work to be done

Immediate Customer adoption:

Agile’ s most important principle is to be in touch with the customer as close as possible by delivering the working software so that the customer will be engaged, however this may not work well for the products which involves good deployment time and customer cannot invest again and again on the deployment for every increment as it involves many tasks like up gradation, compatibility and data migration last but the not the least customization, unless it’s a web based its really hard to see the continues delivery to the customer, if team is looking this is the only reason to adopt agile then better to re think.

Everything is TEAM:

Agile is more about people than process, which is conflicting with CMM. As CMM demands more documentation and more focus on process rather than people, however agile doesn’t say not to documentation and says its team’s decision. The common word we hear in agile teams is its TEAM decision and TEAM is responsible the TEAM concept is very good in a way to avoid blame games and pressures, however TEAM won’t work for responsibility though its TEAMS responsibility one persons should own and perform the task who owns is a question, it’s a common question every one faces it and most will address the same with ease by creating the stories and assigning the owner, however that is not in the scope of this paper.

Most of people think agile methodologies built around highly skilled and experienced individuals in order to respond quickly to the change and not suitable for less experienced people

Most of the People will not consider Agile for critical applications, Outsourcing and Legacy Software development

Another criticism about Agile is time management, in meetings as well in completing the tasks. Agile demands good time management skills from each and every stakeholders of the project. Maintaining time management skill for long is a challenge

Various authors addressed the above-mentioned challenges with some best practices, and some explained them as Myths [14]. However the scope of this paper is not to go in details of them

However, every organization has to evaluate agile and waterfall models in order to determine the applicability based on the above discussion, some teams may conclude that Agile or waterfall is not their cup of tea. It is perfectly fine provided the decision is made up on the facts rather than assumptions.

CONCLUSION

The scope of this paper is not to ascertain or recommend the best model. Of course, each model has got its own merits and demerits. Whether or not to use a certain software

development methodology/tool is very critical and depends on a host of factors including but not limited to organizational culture, flexibility offered by the customer, customer awareness and knowledge about what he wants

However, the paper is intended to deal and come out with a fair comparison of the models and make the users aware of the characteristics of each, in order to enable them to match the same with their experience.

Understanding the differences among various methods improves the decision-making regarding selection of the most suitable methodology in a suitable way.

If your current model is working fine for you stick to it Agile is not a fashion.

REFERENCES:

- [1]. Thesis, Jonna Kalermo and Jenni rissanen, "Agile software development in theory and practice"
- [2].<http://www.martinfowler.com/articles/newMethodology.html>
- [3].<http://www.agilemanifesto.org>
- [4].<http://www.agilesoftwaredevelopment.com>
- [5]. <http://www.drdoobs.com/architecture-and-design/the-distributed-agile-team/212201434?pgno=3>
- [6].http://en.wikipedia.org/wiki/Agile_software_development
- [7].http://en.wikipedia.org/wiki/Dynamic_Systems_Development_Method
- [8]. http://en.wikipedia.org/wiki/Extreme_programming
- [9].<http://alistair.cockburn.us/Crystal+methodologies>
- [10].<http://www.ccpace.com/Resources/documents/AgileProjectManagement.pdf>
- [11].Kathleen B. Hass, "TIPS & TECHNIQUES The Blending of Traditional and Agile Project Management", PM World Today, May 2007 (Vol. IX, Issue V),
http://cs.anu.edu.au/courses/COMP3120/local_docs/readings/the-blending-of-traditional-and-agile-project-management.pdf
- [12]. Prof. James Noble, Dr. Stuart Marshall
http://nzcsrsc08.canterbury.ac.nz/site/proceedings/Individual_Papers/pg218_Agile_Project_Management.pdf
- [13].3rd Annual Survey: 2008 "The State of Agile Development".

[14].

http://www.versionone.com/pdf/AgileMyths_BetterSoftware.pdf

[15]. Huo, M. et. al. Software Quality and Agile Methods. In Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04). IEEE 520-525, Hong Kong. September 2004. May be found at: <http://csdl.computer.org/comp/proceedings/compsac/2004/2209/01/220910520abs.htm>

[16]. Presentation by Yogi Berra on Software Development Life Cycle (SDLC)

BIOGRAPHIES:



Kiran Jammalamadaka received his MCA from Acharya Nagarjuna University, India. He is currently associated with GE India as a Software technical leader. His interests include Software engineering, Agile->Scrum and automated debugging.



V. Ramakrishna received his PhD from Acharya Nagarjuna University, India in 2011. He is currently associated with K L University, India as an Associate Professor. His general research interests are Software Engineering and Neural Networks.