While Shifting to Agile Methodology

Shubham Murudkar
Sasmira Institute of Management Studied and Research, Mumbai University, Mumbai, India

Abstract
In IT companies waterfall model was a respected methodology, but lately, it faced criticism for being an outdated model. The methodology's limitations become more apparent depending on the size, type, and goals of the project it's guiding,so to counter these problems in waterfall model agile methodology was introduced with it's Manifesto and Principles. Agile methodology is been widely used in recent years.This methodology has advanced since business requirements became more demanding. Agile methodologies came into existence after the need for a light way to do software development in order to accommodate changing requirements in environment. The aim of this paper is to help the organizations with selecting and implementing agile practice in different companies as per their culture and structure. In this paper we are theoretically focusing on simple ways to implement agile and challenges during implementation of it.

Keywords
Agile, Implementing agile, Scrum, Kanban, Scrumban

I. Introduction

Agile project management is based on an incremental, iterative approach. Instead of in-depth planning at the beginning of the project, Agile methodologies are open to changing requirements over time and encourages constant feedback from the end users. The goal of each iteration is to produce a working product.
Agile refers to any process that aligns with the concepts of the Agile Manifesto. In 2001, 17 software developers met to discuss lightweight development methods. They published the Manifestoand 12 principles for Agile Software Development, which covered how they found "better ways of developing software by doing it and helping others do it."

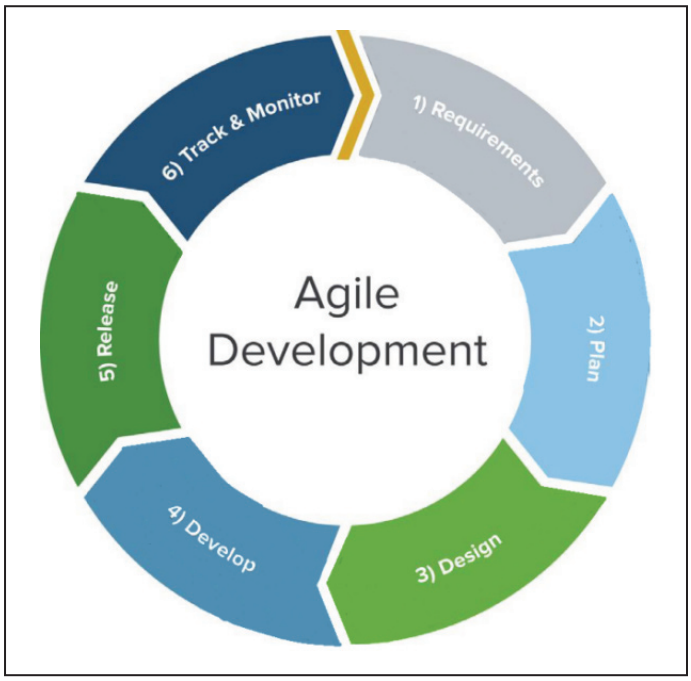

Fig. 1: Agile Development

A. The Agile Manifesto

The Manifesto for Agile Software Development according to the Agile Alliance (Cockburn 2002):

"We are uncovering better ways of developing software by doing it and helping others doit. We value:

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

That is, while we value the items on the right, we value the items on the left more."

B. The Agile Principles

The Agile Software Development principles according to the Agile Alliance (Beck,Martin et al. 2001):

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughoutthe project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective,then tunes and adjusts its behavior accordingly.

II. Literature Review

A. Agile Methodologies

There exist a numerous of different agile software development methods that can be considered as a part of the agile family. But these are the most well-known methods:

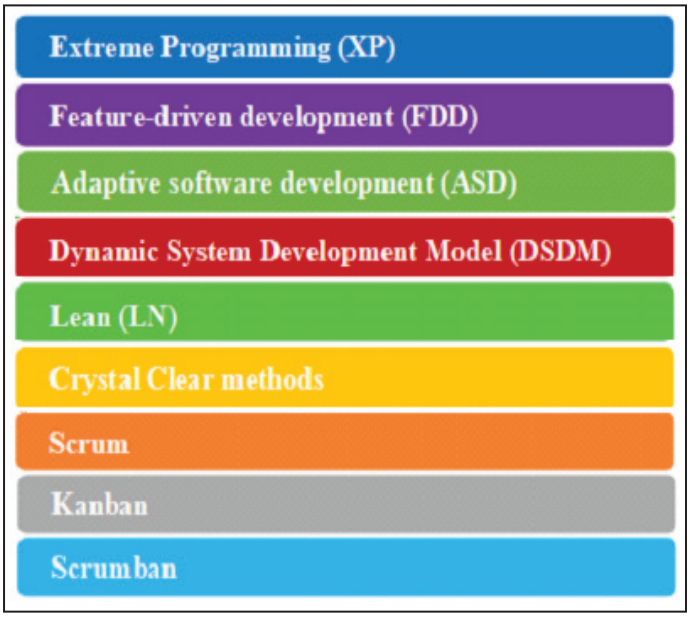

Fig. 2: List of Agile Methods

Some methodologies like Scrum, Kanban and Scrumban are describe below as these methodologies are easy to implement

B. Scrum Methodology

Scrum is a subset of Agile and one of the most popular process frameworks for implementing Agile. It is an iterative development model often used to manage complex software and product development. Fixed-length iterations, called sprints lasting one to two weeks long, allow the team to ship software on a regular cadence. At the end of each sprint, stakeholders and team members meet to plan next steps.

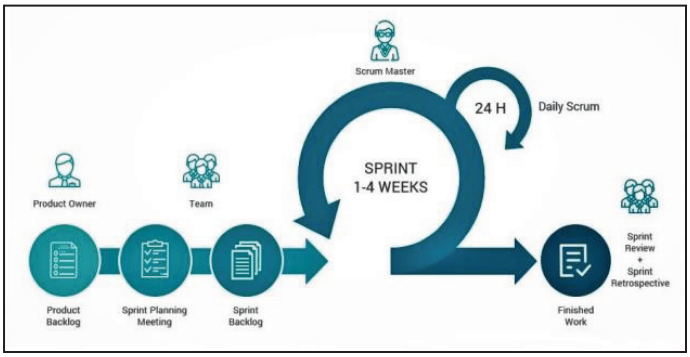

Fig. 3: Scrum Process

C. Roles in Scrum

There are three specific roles in Scrum:

1. Product Owner

The Scrum Product Owner has the vision of what to build and conveys that to the team. He or she focuses on business and market requirements, prioritizing the work that needs to be done, managing the backlog, providing guidance on which features to ship next, and interacting with the team and other stakeholders to make sure everyone understands the items on the product backlog.

2. Scrum Master

Often considered the coach for the team, the Scrum Master helps the team do their best possible work. This means organizing meetings, dealing with roadblocks and challenges, and working with the Product Owner to ensure the product backlog is ready for the next sprint.

3. Scrum Team

The Scrum Team is comprised of five to seven members. Unlike traditional development teams, there are not distinct roles like programmer, designer, or tester. Everyone on the project completes the set of work together.

D. Steps in the Scrum Process

There are a specific, unchanging set of steps in the Scrum flow:

1. Product Backlog

The product backlog is not a list of things to be completed, but rather it is a list of all the desired features for the product. Sprint planning: Before each sprint, the Product Owner presents the top items on the backlog in a sprint planning meeting. The team determines the work they can complete during the sprint and moves the work from the product backlog to the sprint backlog.

2. Backlog Refinement/Grooming

At the end of each sprint, the team and Product Owner meet to make sure the backlog is ready for the next sprint. The team may remove user stories that aren't relevant, create new user stories, reassess the priority of stories, or split user stories into smaller tasks.

3. Daily Scrum Meetings

The Daily Scrum is a 15-minute stand-up meeting that happens at the same time and place every day during the sprint. During the meeting each team member talks about what they worked on the day before, what they'll work on today, and any roadblocks.

4. Sprint Review Meeting

At the end of each sprint, the team presents the work they have completed as a live demo rather than a presentation. Sprint retrospective meeting: Also at the end of each sprint, the team reflects on how well Scrum is working for them and talks about any changes that need to be made in the next sprint.

E. Kanban Methodology

Kanban is Japanese for "visual sign" or "card." It is a visual framework used to implement Agile and shows what to produce, when to produce it, and how much to produce. It encourages small, incremental changes to your current system and does not require a certain set up or procedure (meaning, you could overlay Kanban on top of other existing work flows).

Kanban was inspired by the Toyota Production System and Lean Manufacturing. In the 1940s, Toyota improved its engineering process by modeling it after how supermarkets stock shelves. Engineer Taiichi Ohno noticed that supermarkets stock just enough product to meet demand. Inventory would only be restocked when there was empty space on the shelf (a visual cue).

These same ideas apply to software teams and IT projects today. In this context, development work-in-progress (WIP) takes the place of inventory, and new work can only be added when there is an "empty space" on the team's visual Kanban board. Kanban matches the amount of WIP to the team's capacity, improving flexibility, transparency, and output.

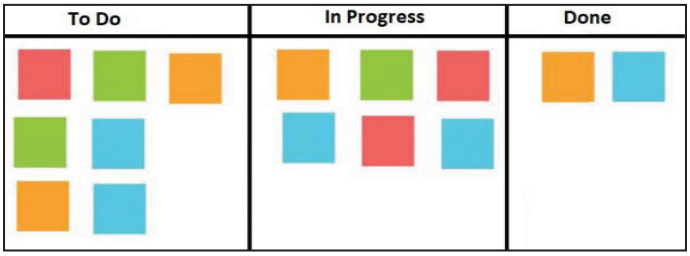

Fig. 4: Simple Kanban Board

F. The Kanban Board

A Kanban board is a tool to implement the Kanban method for projects. Traditionally, this tool has been a physical board, with magnets, plastic chips, or sticky notes on a whiteboard. However, in recent years, more and more project management software tools have created online Kanban boards.

A Kanban board, whether it is physical or online, is made up of different swim lanes or columns. The simplest boards have three columns: to do, in progress, and done. Other projects may consist of backlog, ready, coding, testing, approval, and done columns.

Kanban cards (like sticky notes) represent the work and each card is placed on the board in the lane that represents the status of that work. These cards communicate status at a glance. You could also use different color cards to represent different details. For example, green cards could represent a feature and orange cards could represent a task.

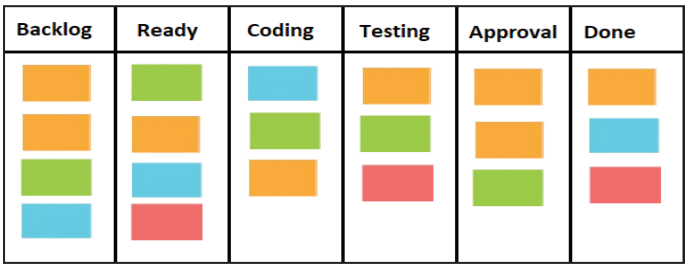

Fig. 5: Kanban Board for Project

G. Core Practices and Principles of Kanban

Every Kanban project should follow these core principles:

1. Visualize the Work Flow

A visual representation of your work allows you to understand the big picture and see how the flow of work progresses. By making all the work visible you can identify issues early on and improve collaboration.

2. Limit Work in Progress (WIP)

Work in progress limits determine the minimum and maximum amount of work for each column on the board or for each work flow. By putting a limit on WIP, you can increase speed and flexibility, and reduce the need for prioritizing tasks.

3. Manage and Enhance the Flow

The flow of work throughout the Kanban board should be monitored for possible improvements. A fast, smooth flow shows the team is creating value quickly.

4. Make Process Policies Explicit

Everyone needs to understand how things work or what qualifies as "done". Modify the board to make these processes more clear.

5. Continuously Improve

The Kanban method encourages small, continuous changes that stick. Once the Kanban system is in place, the team will be able to identify and understand issues and suggest improvements.

H. Scrumban

Scrumban = Scrum + Kanban

We know Scrum and Kanban as method of Agile. Scrum is best-suited for products and development projects. Kanban is best for production support. We use Scrumban – which combines the best features of both for maintenance projects. Scrumban is becoming very popular these days in service industries, where we have both development and maintenance projects.Use the prescriptive nature of Scrum to be Agile.Use the process improvement of Kanban to allow the team to continually improve its process.

With the Kanban's pull system in place, our flow will become smoother as our process capability improves. We can use our inter-process buffers and flow diagrams to show us our process weaknesses and opportunities for kaizen. As we get closer to level production, we will start to become less concerned with burndown and more concerned with cycle time, as one is the effect and the other is the cause. Average lead time and cycle time will become the primary focus of performance. If cycle time is under control and the team capacity is balanced against demand, then lead time will also be under control. If cycle time is under control, then burndowns are predictable and uninteresting.

Since the team now pulls work into a small, ready queue before pulling it into WIP, the team's perspective of the iteration backlog's utility is that it always contains something that is worth doing next. Therefore, we should use the least wasteful mechanism that will satisfy that simple condition.

In Scrumban, we can do iteration planning at regular intervals, synchronized with review and retrospective, but the goal of planning is to fill the slots available – not fill all of the slots, and certainly not determine the number of slots. This greatly reduces the overhead and ceremony of iteration planning. Time spent batch-processing for iteration planning estimation can be replaced with a quality control inspection at the time that work is promoted to the ready queue. If a work item is ill-formed, then it gets bounced and repeat offenders get a root cause analysis.

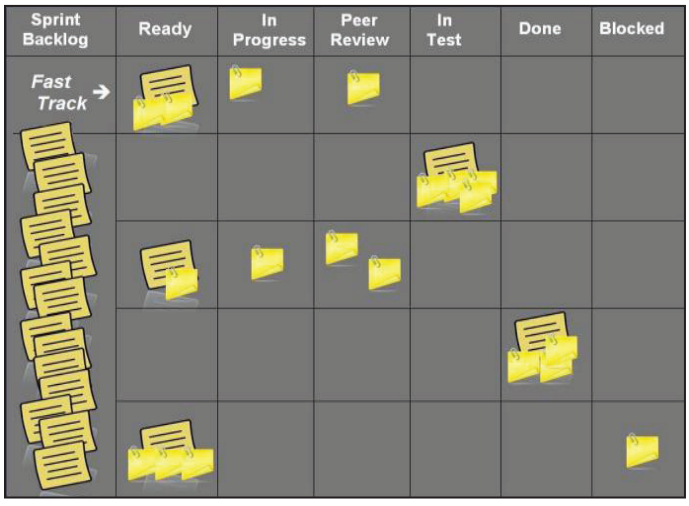

Fig. 6: Scrumban Board

III. Methedology

A. Easy ways to Implement Agile

Whichever route you choose, remember that Agile is flexible in its vary nature. There is no wrong or right way to get started with Agile.

B. Stand up Meetings

A simple way to get started with Agile is to incorporate daily stand-up meetings into your project. Daily stand-up meetings are easy to incorporate into any project methodology you already use (even Waterfall) and don't require any training or knowledge transfer.

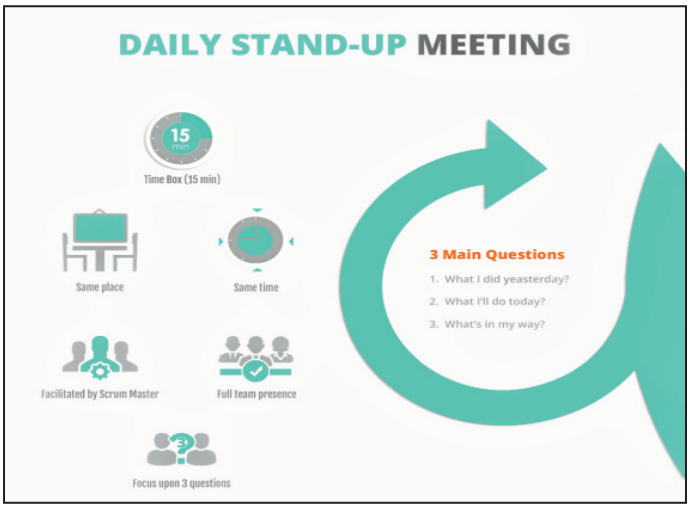

Fig. 7: Stand Up Meet

C. Kanban Board

Another way to incorporate Agile practices is to create and use a Kanban board. The Kanban board is simple tool to help your team visualize the flow of work as it's getting done. Use the board during stand-up meetings to discuss current work in progress or display it where your team can easily access it to make updates to task status.

Fig. 8: Kanban Board

D. Changing Team Roles

Some methods of Agile may result in the need to change team roles. For example, working with Scrum, the team may need to take more responsibility and boost speed of delivery. A good place to start with Scrum is to talk about the roles and responsibilities. Every project must have a Scrum Master, Product Owner, and Scrum Team. Clarifying these roles will help teams understand their responsibilities and remain accountable.

Fig. 9: Team Roles

E. The Agile software development lifecycle

The overall goal of each Agile method is to accept the change and deliver working software as soon as possible. However, each methodology has slight variations in the way it defines the phases of software development. moreover, even though the goal is the same, each team's process flow may differ depending upon the specific project or situation. The full Agile software development lifecycle includes the concept, inception, construction, release, production, and retirement phases.

Concept	Projects are envisioned and prioritized
Inception	Team members are identified, funding is put in place, and initial environments and requirements are discussed
Construction	The development team works to deliver working softwarebasedoniterationrequirements and feedback
Release	QA (Quality Assurance) testing, internal and external training, documentation development and final release of the iteration into production
Production	Ongoing support of the software
Retirement	End-of-life activities, including customer notification and migration

This presents the full Agile lifecycle model within the enterprise. In any enterprise there may be projects operating simultaneously, multiple iterations will take place during the Agile software development lifecycle and each follows its own work flow. During an iteration, it is important that the customers and business stakeholders provide feedback to ensure that the features meet their needs.

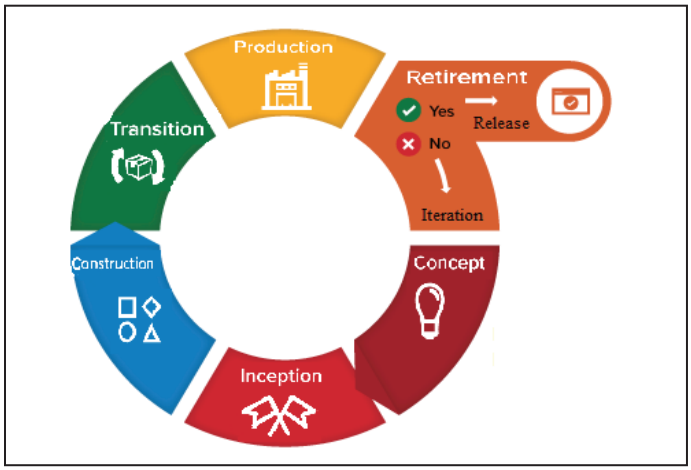

Fig. 10: Agile Software Development Lifecycle

F. Challenges During Implementing Agile Methodology

1. Mindset Change

When you catch yourself in the trap of saying "this is just like what we're already doing except…" you've missed the key point. Agile Software development requires a complete mindset change. While some of the events (i.e. Daily Scrum) may be superficially similar the mindset is different.

2. Self Organization

Self organization is messy and takes time to achieve yet its at the heart of a high performing Scrum team. So its sad to see Scrum Master, an Architect or other team member with a leadership position taking control. This problem can be obvious or subtle but either way they're not giving the team room to grow.

3. Meetings

You can hold all the right meetings (Planning, Daily Standup, Review and Retrospective) if the joy and spirit is missing then they will be wooden events that don't but help the team collaborate and grow.

4. No Engineering Practices

Agile processes (Scrum, OpenAgile, Kanban etc.) tend to lead to an immediate speed up in the development process. But if there is no focus on working on Engineering practices (TDD, ATDD and Pair Programming), then the team will soon end.

5. Retrospectives Without Improvement

A retrospective who's action items are not acted on quickly becomes a meeting people will have no respect for. Take the SMART actions from your retrospective, post them on your task board and check in on them in stand-up everyday.

6. Command and Control Agile

A Scrum Master, XP Coach or even Open Agile Process Facilitator isn't a Project Manager. They don't assign tasks or tell team members what to do. Instead they're there to help the team achieve its goal and to help them grow into a high performing team.

7. Pre Assigned Tasks

During Sprint Planning team members often pre-assign stories/ tasks to particular team members. The problem here is that we can't foretell the future if the person to whom the task is assigned we have a bottleneck. The task is blocked until they become available.

Instead of pre-assigning the best person to task, its better to wait to see who is available to tackle it. This often leads to team members learning new skills.

8. Attempting to Scale at the Start

Organizations that are adopting Agile often want to work at scale (more than 3 teams) right off the bat. Its difficult to get a few teams off to a good start, without complicating matters with additional communication problems. Launching at scale can be done, but it tends to require alot of coaching support. If you're starting on your own start small.

G. Agile project management software

Companies using agile are likely to leverage software geared to agile development in order to get the full benefits of this methodology. Here are just some of the agile solutions available:

1. Atlassian Jira + Agile

This is an agile project management tool that supports Scrum, Kanban, and mixed methodologies. This project management software comes with a comprehensive set of tools that help Scrum teams perform events with ease.

2. Agilean

Agilean automates work flow management for small and midsize IT companies fitting different verticals. It is customizable and has 50 built-in templates.

3. Sprint Ground

This is a project management tool created for developers to organize work and help them track progress.

4. Version One

This project management solution is built to support the Scaled Agile Framework at all levels.

IV. Conclusion

As agile methodologies are suitable in changing environments, Scrum, Kanban and Scrumban are easy to implement because of new practices and principles that enable a team to develop a product in short duration which will help in organizational growth.

Reference

[1] Abdallah Salameh: On Process Tailoring - An Agile Example Master of Science Thesis in Software Engineering.

[2] Diana Ramos: Agile Project Management 101 - A Beginner's Guide for Non-Project Managers.

[3] Savita Pahuja: [Online] Available: https://www.agilealliance.org/what-is-scrumban/

[4] Zain : [Online] Available: https://www.smartsheet.com/understanding-agile-software-development-lifecycle-and-process-workflow

[5] MikeCottmeyer: [Online] Available: https://kellycrew.wordpress.com/2011/02/28/top-10-issues implementing-agile-okay-14/

[6] Moira Alexander: [Online] Available: https://www.cio.com/article/3156998/agile-development/agile-project-management-a-beginners-guide.html