

# **Agile Software Process**

By

Sunil Kumar(Master of Sc.)

Bangalore, India

# Agenda

- What is Agile?
- History of Agile
- Agile Manifesto
- Agile Methodologies Overview
- Roles within in Agile Team
- Summary

# What is Agile

- Software evolve over time.
  - Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing and cross-functional teams.
- Adaptive planning, evolutionary development, early delivery
  - It promotes adaptive planning, evolutionary development, and early delivery for your highly iterative and incremental approaches to software development.

# What is Agile

- The term agile was first used in 2001 when the Agile Manifesto was published as a unifying charter by many of the leading visionaries in the software field at the time, who were fed up with traditional approaches to software development.
- Unlike traditional software development practices, Agile development methodologies such as Scrum, Extreme Programming, Lean, Feature-Driven Development, Crystal, Adaptive Software Development, and others, incorporate close cross-functional collaboration, and frequent planning and feedback as fundamental tenets inherent in the evolution of a software system.

# What is Agile

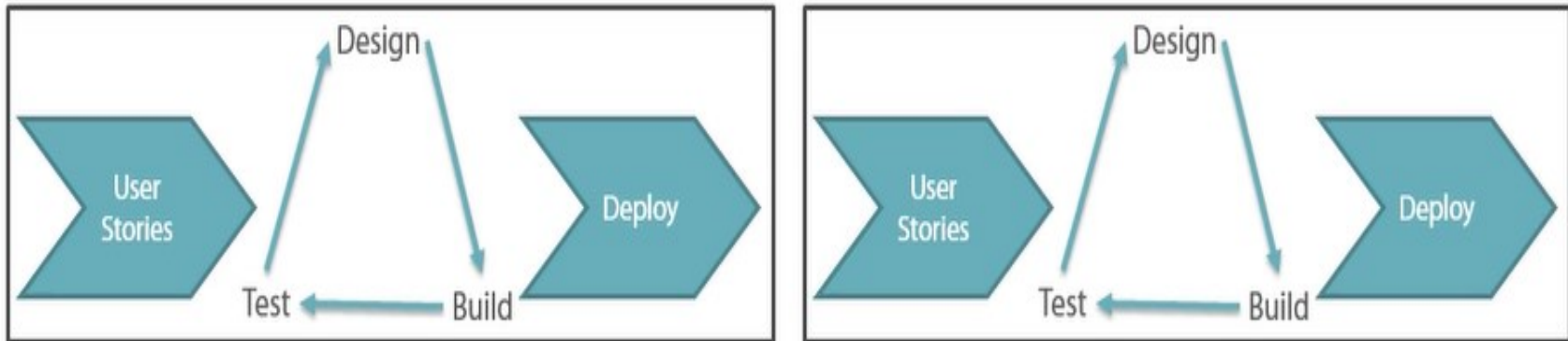
- Deliver value to business sooner
  - One of the key ideas behind Agile is that instead of delivering a "big bang" at the end of the project, you deliver multiple releases of working code to your business stakeholders. This allows you to prioritize on features that will deliver the most value to the business sooner, so that your organization can start to realize an early return on their investment.
  - The number of deliveries that we do depends on how long and complex a project is, but ideally you would deliver working software at the end of each sprint or iteration.
- Flexible response to change

# Agile vs Traditional Model

Traditional Waterfall



Agile Development



Project Timeline



# What is Agile

Incrementally



Instead of all at once



# What is Agile

- As per this diagram, it shows that with Agile we deliver incrementally instead of all at once. You should hold this thought in your mind as we progress through the rest of the material about Agile development.



# History of Agile

- Many attempts to improve software development
- Small group of thought leaders met to discuss software process.
- This term was used as an umbrella reference to a family emerging lightweight software development methods like
  - Scrum
  - Extreme Programming (XP)
  - DSDM (Dynamic System Development Process)
  - FDD (Feature Driven Development)
  - Crystal
  - Adaptive Software Development

# Agile Manifesto

- 17 people met to find the common ground about software development.
- They formed “The Agile Alliance” and signed the agile manifesto.  
<http://agilemanifesto.org>
- 4 core values
- 12 Principles behind the agile manifesto split into 3 groups.
  - Regular delivery of Software
  - Team Communication
  - Excellence in design

# Agile Manifesto 4 Core Values

More Values Items	Valued-but less than left items
Individual and interactions	Over processes and tools
Working software	Over comprehensive documentation
Customer collaboration	Over contract negotiation
Respond to change	Over following a plan

# Agile Manifesto Values

Values	Explanation
Individual and interactions over processes and tools	<ul style="list-style-type: none"><li>• Strong collaborating team &gt; process tools</li><li>• Coherent team &gt; Individual gurus</li><li>• Put more effort in building coherent teams</li></ul>
Working software over comprehensive documentation	<ul style="list-style-type: none"><li>• Documentation is important</li><li>• Too much documentation is costly and documents become outdated</li><li>• Generate just enough high-level structure/dynamics documents.</li></ul>
Customer collaboration over contract negotiation	<ul style="list-style-type: none"><li>• Continuous customer feedback is essential</li><li>• Customer work closely with team</li><li>• Feedback derive future iterations</li></ul>
Respond to change over following a plan	<ul style="list-style-type: none"><li>• Plans must be flexible to change</li><li>• Change can and will happen</li><li>• Multi-level planning.<ul style="list-style-type: none"><li>• Weekly detailed plans</li><li>• Iteration plan</li><li>• Overall plan</li></ul></li></ul>

# Agile Methodology Overview

- Scrum
  - Scrum is a lightweight management framework with broad appeal for managing iterative and incremental projects of all types.
  - Ken Schwaber, Mike Beedle, Jeff Sutherland, and others, contributed significantly to the evolution of Scrum over the last decade and a half.
  - Over the last few years in particular, Scrum has garnered increasing popularity in the software community due to its simplicity, proven success, and improved productivity, and its ability to act as a wrapper for various engineering practices promoted by other Agile methodologies.

# Agile Methodology Overview

- Extreme Programming (XP)
  - XP or Extreme Programming was originally devised by Kent Beck, and has emerged as one of the more popular and controversial Agile methods.
  - XP is a disciplined approach at delivering high quality software quickly and continuously.
  - It promotes high customer involvement, rapid feedback loops, continuous testing, continuous planning, and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks.
  - The original XP recipe is based on four simple values: Simplicity, communication, feedback, and courage.
  - There are also 12 supporting practices. These are the planning game, small releases, customer acceptance tests, simple design, pair programming, test-driven development, refactoring, continuous integration, collective code ownership, coding standards, metaphors, and sustainable pace.

# Agile Methodology Overview

- Crystal Methodology
  - The Crystal methodology is one of the most lightweight, adaptable approaches to software development.
  - Crystal is actually comprised of a family of methodologies like Crystal Clear, Crystal Yellow, and Crystal Orange, whose unique characteristics are driven by several factors, such as team size, system criticality, and project priorities.
  - 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.

# Agile Methodology Overview

- DSDM
  - Dynamic Systems Development Method, or DSDM as it is most commonly known.
  - DSDM dates back to 1994 and grew out of the need to provide an industry standard project delivery framework for what was referred to as rapid application development or RAD for short.
  - While RAD was extremely popular in the early 1990's, the RAD approach to software delivery evolved in a fairly unstructured manner.



# Agile Methodology Overview

- As a result, the DSDM consortium was created and convened in 1994 with the goal of devising and promoting a common industry framework for rapid software delivery.
- Since 1994, the DSDM methodology has evolved and matured to provide a comprehensive foundation for planning, managing, executing, and scaling agile and iterative software development projects.
- DSDM is based on nine key principles that primarily revolve around business needs and values, active user involvement, empower teams, frequent delivery, integration testing, and stakeholder collaboration. DSDM specifically calls out fitness for business purpose as the primary criteria for delivering acceptance of a system, focusing on the useful 80% of the system that can be deployed in 20% of the time.

# Agile Methodology Overview

- FDD
  - Feature-driven design was originally developed by Jeff De Luca. The first incarnations of FDD occurred as a result of collaboration between De Luca and a thought leader, Peter Coad.
  - FDD is a model-driven, short-iteration process. It begins with establishing an overall model shape. Then it continues with a series of 2-week, design-by-feature, built-by-feature iterations. The features are small and useful in the eyes of the client.
  - FDD designs the rest of the development process around feature delivery using the following eight practices: Domain object modeling, developing by feature, components and class ownership, feature teams, inspections, configuration management, regular builds, visibility of progress, and results. Next we have Lean software development.

# Agile Methodology Overview

- Lean Software Development
  - Lean software development is an iterative methodology originally developed by Mary and Tom Poppendieck.
  - Lean software development owes much of its principles and practices to the Lean enterprise movement and the practices of companies like Toyota.
  - Lean software development focuses the team on delivering value to the customer and on the efficiency of the value stream and the mechanisms that deliver that value.
  - The main principles of Lean include eliminating waste, amplifying learning, deciding as late as possible, delivering as fast as possible, empowering the team, building in integrity, and seeing the whole. Lean eliminates waste by selecting only the truly valuable features for a system, prioritizing and delivering them in small batches. It emphasizes the speed and efficiency of development workflow and relies on rapid and reliable feedback between programmers and customers.

# Agile Methodology Overview

- Kanban
  - Kanban is applied to software development as a based planning and execution method. Rather than planning work items up front and pushing them into the work queue of a team, the team signals when they are ready for more work and pulls it into their queue.
  - Kanban historically uses cards to signal the need for an item. For software development teams, these cards are kept on a Kanban board, which is organized into columns and rows.
  - The columns represent the different states of a work item, from initial planning through customer acceptance. The specific columns a team uses should meet the needs of the team and be tailored to their context.
  - The rows on a Kanban board represent work items. Work items are sometimes grouped within areas such as feature sets and category types.

# Agile Methodology Overview

- Kanban focuses on maximizing the throughput of a team. One of the ways it achieves this goal is through the application of work-in-process limits, or WIP limits, in each of the states of a work item.
- Under a Kanban, or Lean approach, of work in any state are seen as waste. The work-in-process limits enable the team to focus on the optimal flow of work items through the system, minimizing any associated waste. Kanban allows teams to achieve process optimizations while respecting and maintaining at a sustainable pace

# Roles in Agile Team

- Agile teams are software development team first and members of a department second.
  - Everyone is a team member committed to a common vision and goal. In order to achieve these goals, team members do what is needed at the time it is needed.
  - This can be a bit unnerving for a number of folks used to strict roles and titles, but it is critical for the common good of the team.
  - Statements like I can't write tests, I'm a developer, have no place in an Agile team. Naturally everyone comes into a team with their own area of expertise and experiences.
  - Outside of the common project, the team still needs to fit into the larger organization, so you do still need to keep some of these titles, but if the entire team can make the shift in thinking, the results can be amazing.

# Roles in Agile Team

- A team should have a product or domain expert.
  - There are team members with product and domain knowledge.
  - These people interact closely with users and/or customers, and in a commercial software scenario attempt to understand and research what benefits the market.
  - They serve the team by creating and prioritizing work, along with helping understand when work is complete. Commonly called a product owner in Scrum, this group can also consist of product managers, business analysts, and customers, and anyone with deep domain knowledge.
  - A critical success factor for this person or group is to act and speak as a single deciding voice of software product decisions.

# Roles in Agile Team

- A team has members with cross functional skills
  - There are also team members who possess all the cross-functional skills necessary to build, validate, and deploy the product, and to determine how, if necessary, it interoperates with the larger system.
  - This group of core team members can consist of developers, testers, user interface designers, architects, business analysts, team leads, and technical writers.
  - This group works as a single team to design, code, test, and implement the software system.
  - The skills required to do so are extensive, and as a result, it is critical that everyone on the team can work together and act as a single unit. This does not mean that the team does not consist of Agile developers, Agile testers, or Agile usability experts, etc., just that they all work and collaborate together on the delivery of a software system. As a result, team members often perform multiple roles in addition to their primary responsibility.



# Roles in Agile Team

- A team should have some form of leadership role.
  - In Scrum, this role is typically referred to as the Scrum master.
  - In some project, this may take on the project manager, program manager, team lead, or other title. On Agile teams, the primary goal of this person or group is simply to enable and ensure the success of the team.
  - This servant leadership role is in sharp contrast to the traditional task master and resource management focus of a Waterfall project.
  - Additional responsibilities may revolve around managing stakeholder expectations, company communication, progress reporting, facilitation, motivating team members, budget management, and staff management.
  - Just because a team is operating in an agile fashion, does not mean that all organizational requirements disappear, but every effort is taken to ensure that they have minimal impact on the ultimate success of the team.

# Roles in Agile Team

- A team member can benefit from an agile coach or mentor.
  - Often referred to as an Agile coach, this person ideally does not have specific project deliverables, and instead maintains a focus on continuous improvement and engineering discipline. One goal of an Agile team should be to become better every single day, and the coach's job is to help guide the team to this direction.
  - The larger the organization, the more complex team structures can get. Cross-project teams, shared services, operations, configuration management, and database administration can all come into play, but the goal remains the same, define a software project and cross-functional team capable of delivering on that project and empower the team to do so.

# Summary

- What is agile and how it differs to the more traditional Waterfall software development process. The main visible difference with Agile is that we deliver a business value sooner to the business by following an iterative development process, as opposed to the more "big bang" approach of Waterfall. We deliver incrementally instead of all at once.
- History of Agile with the formation of The Agile Alliance, who defined the Agile Manifesto.
- The Agile Manifesto represents a series of 4 core values and 12 principles that should guide us on how we develop software.
- Series of different Agile software development methodologies that are in use today in different software development teams.
- The most common are Extreme Programming and Scrum, but Lean development and Kanban are also increasing in popularity.
- Different roles we would find in an Agile team, from the development team itself to the product owners and domain experts, team leadership, and Agile coaches.