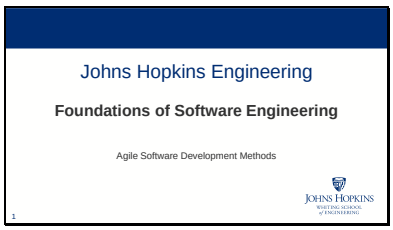
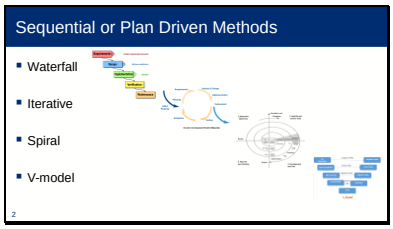
Slide 1



This presentation describes agile software development methods.

Slide 2

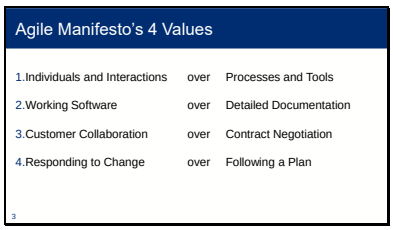


For the past fifty years Project managers have organized, tracked, and paced work to deliver solutions. In software efforts, the general term used is the Software Life Cycle or SLC. Historically the SLC is implemented using a sequential or plan driven method like waterfall, iterative, Spiral, V-model and others. Sequential or Plan driven methods determine up-front the goals and requirements of the project using thorough analysis.

The project transitions through multiple phases; requirements, design, implementation, verification, and maintenance with the intent to deliver exactly to plan. Completion and subsequent delivery to customers can and often does take years, to a rigidly adhered to schedule.

Success is measured by delivering “on time and within budget”.

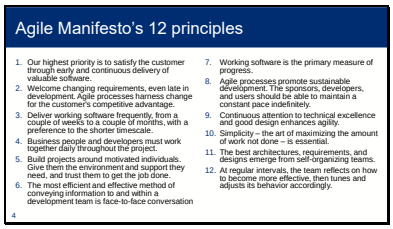
Slide 3



In 2001, a formal structure was applied to a new methodology that allowed continuous requirement reconsideration and evolution. The Agile Manifesto detailed four values and on the next slide, twelve principles that describe the tenants of what today is called the Agile methodology.

Modern development requires the opportunity for frequent “course corrections”. This recognizes that requirements are dynamic things and that two years of development without feedback puts an effort at extreme competitive disadvantage.

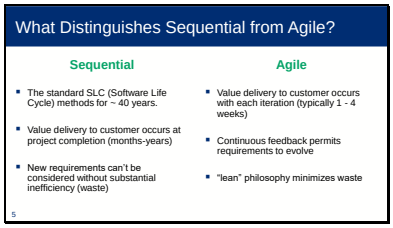
Slide 4



Put simply, the Agile Manifesto promotes processes that focuses on quality and value by continuously creating products and features that meet consumers' expectations. The end result is a process that maintains alignment to business objectives, by responding and pivoting as user needs and market forces change.

It is important to note that Agile methods don’t just apply to software development. They can and often are applied to complex tasks as well.

Slide 5



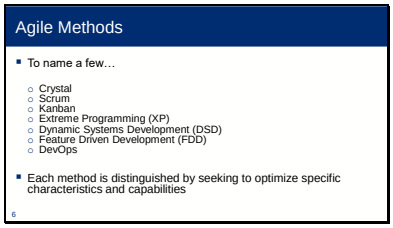
The primary difference between Agile and Sequential methods is the delivery of value to the customer.

- Sequential methods deliver value on completion of testing which is typically months to years from establishment of the concept.
- Since agile provides working code to the customer each iteration, the value delivery is continuous.

Some other differences;

- Agile permits requirements to be reconsidered each cycle, and new requirements to be accepted.
- Agile requires continuous customer involvement
- Agile can use self-organizing teams
- Agile is inefficient when the requirements are static

Slide 6

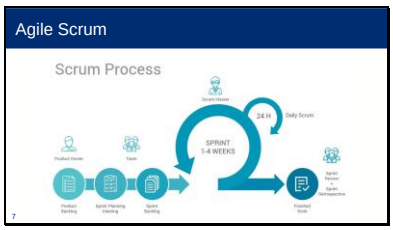


Today there are many agile methods, to name a few: Scrum, Crystal, and Kanban.

Agile methods have been formally available for over fifteen years, and new ones are emerging continuously. Each method has its own characteristics, with unique strengths and weaknesses.

Some feature loose organization while others are more formal and rigid.

Slide 7



Scrum is the most widely used Agile method. It seeks to minimize time to value delivery to the customer as compared to the classic waterfall method.

Agile scrum utilizes a self-organizing team of 5-9 people. Their roles are team member, scrum master, product owner, and stakeholder.

The Scrum master facilitates the team and removes obstacles.

The product owner:

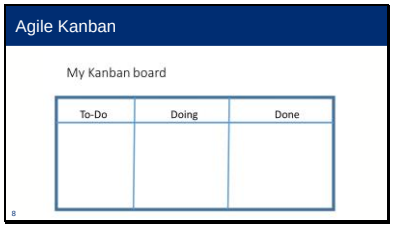
- Represents all stakeholders (customers)
- Identifies tasks to be done, writing them as user stories
- Places tasks to be done into the product backlog with associated priority
- Receives deliverables and assesses their value

Some ‘guidelines’ for Scrum ‘rituals’

- Sprint (fixed duration set by scrum master and product owner typically 1-4 weeks)
- Daily scrum – 15 minutes
- Sprint planning – 2 hours per week of sprint
- Sprint review – 1 hour per week of sprint
- Sprint retrospective – 45 minutes per week of sprint

The team will iterate through sprints attempting to implement items selected from a backlog of desired features. At the end of each sprint, new working code is delivered to the product owner.

Slide 8



Agile Kanban seeks to achieve Just-in Time delivery while not overloading the team members. It is particularly suited for tasks of known duration and features transparent status of all proposed, initiated and completed work. This is visualized on a “Kanban board”.

Unlike Scrum, Kanban is not time-boxed and there are no sprints. Instead tasks begin in a to-do state, and are attempted by team members who move them to ‘doing’ and finally to ‘done’

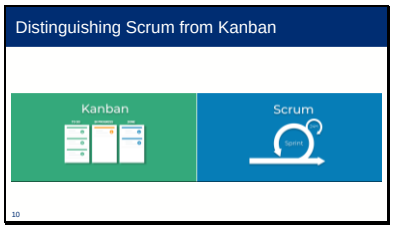
The amount of Work in each column is limited via Work in Process (or WIP) limit, and each column may have its own WIP limit.

Additional columns can be added to facilitate better granularity.

Slide 9

There are many Kanban board implementations, with typically simple characteristics.
Vertical columns illustrate the status of each task.

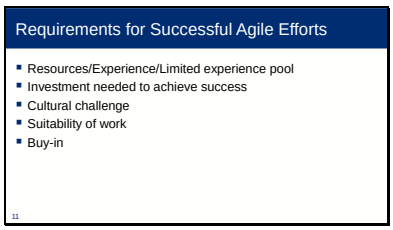
Slide 10



There are distinguishing characteristics for each agile method.

- Scrum for example has utility for cross functional teams (or teams with multiple capabilities), while Kanban doesn't need them.
- Kanban uses a Kanban board, while scrum can utilize a graphical board it merely shows the tasks being worked in a given sprint.
- Scrum tasks are time boxed (because sprints have a specific duration) while Kanban limits only number of simultaneous tasks being worked.
- Since Scrum time-boxes tasks, it can use metrics such as the # of tasks attempted/implemented per sprint (known as velocity). In Kanban that metric is not used since there are no sprints.

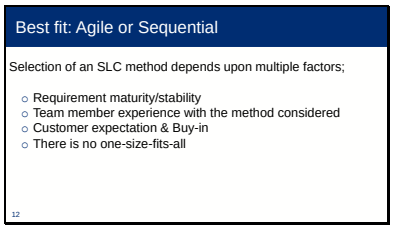
Slide 11



Success in agile can be influenced by multiple factors.

- The availability of team members fluent in a particular agile method can often constrain the desire to attempt it.
- Developing agile experience can take significant time and investment, and this often presents a cultural challenge.

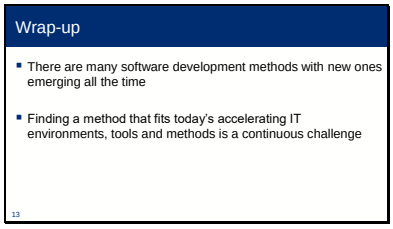
Slide 12



The selection of an appropriate SLC method, be it sequential or agile is critical to success in software development.

This decision will determine and manage the expectations of the stakeholders and greatly influence the outcome of the project.

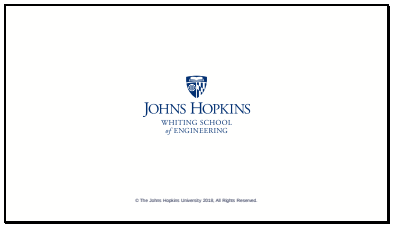
Slide 13



There are many software development methods, and new ones appear all the time.

The challenge is to select the appropriate method, considering the accelerating IT environments

Slide 14



This has been a brief introduction into agile software development methods.