Scrum Master

Objective

By the end of this training you will be able to understand:

1. Agile
2. Scrum as a Framework to implement Agile
3. Your Role as a Scrum Master

This training is your one stop solution to all Scrum related questions and problem.

What is Agile?

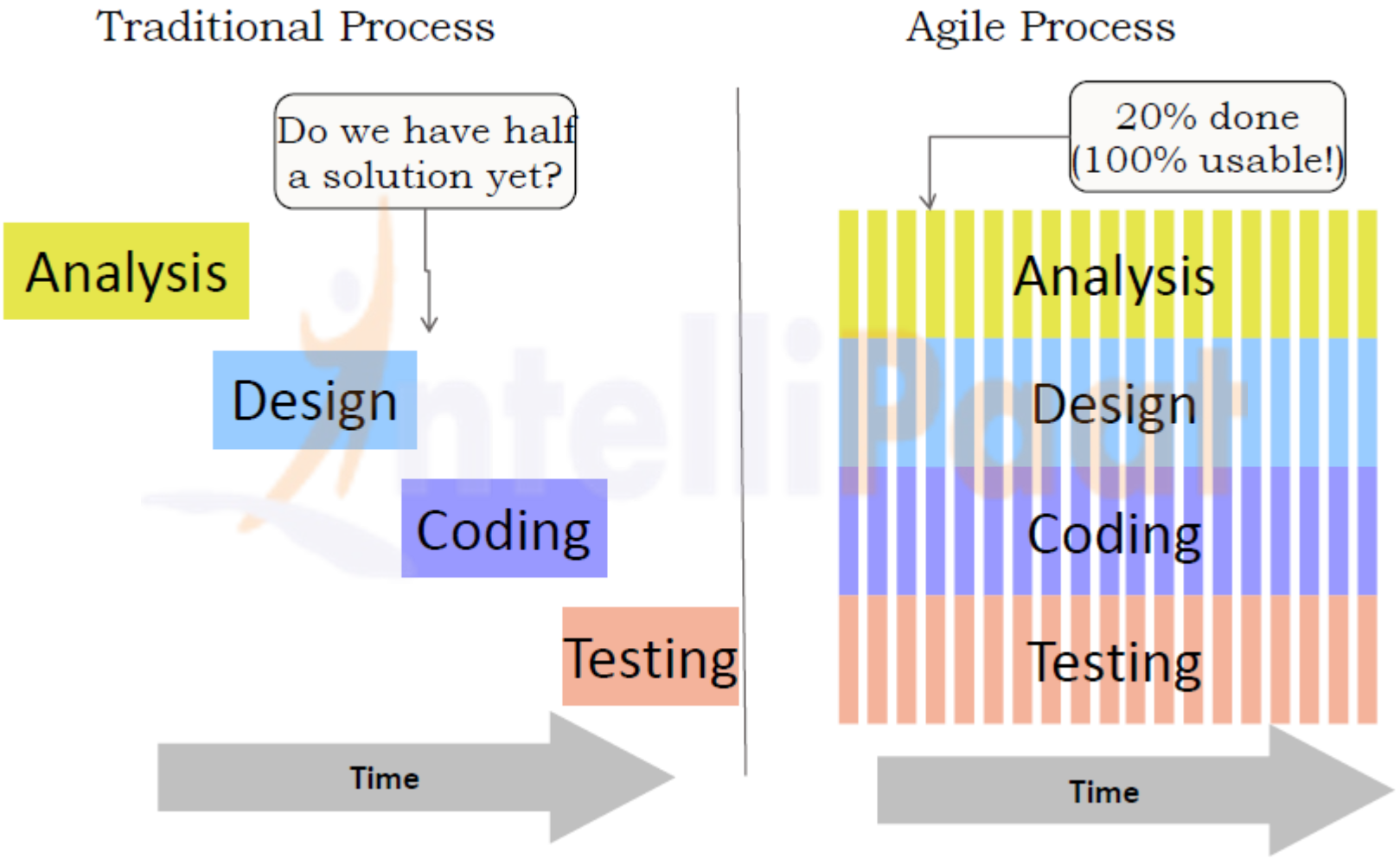
- Agile software development is a conceptual framework for software engineering that promotes development iterations throughout the life-cycle of the project.
- Software developed during one unit of time is referred to as an iteration, which may last from one to four weeks.
- Agile methods also emphasize working software as the primary measure of progress

Issues with Traditional approach

Lets name a few..

- Fixed Scope
- Lack of cross functional developers
- Communication Issues
- Weak Feedback mechanism
- Bad Estimations
- Fail-late process
- Less engagement with client
- Scope creep
- Up-front design resists change

Agile = Early Value



Agile Manifesto

While there are values in the items on the right, we value the items on the left

Kent Beck, James Grenning, Robert C. Martin, Mike Beedle, Jim Highsmith, Steve Mellor, Arie van Bennekum, Andrew Hunt, Ken Schwaber, Alistair Cockburn, Ron Jeffries, Jeff Sutherland, Ward Cunningham, Jon Kern, Dave Thomas, Martin Fowler, Brian Marick

Agile Principles

- Satisfying **customer** is top priority
- Welcome **changing requirements**, even late in development
- Deliver **working software** frequently, from a couple of weeks to a couple of months
- **Development teams** and **business** work together
- Build projects around **motivated** people
- The most efficient and effective method of **conveying information** to and within a development team is **face-to-face conversation**

Agile Principles

- The primary measure of success is **working software**
- Agile processes promote **sustainable development** –The sponsors, developers and users should be able to maintain a **constant pace**.
- Continuous attention to **technical excellence** and **good design**
- **Simplicity**–the art of maximizing the amount of work not done—is essential
- The best **architectures, requirements and designs** emerge from **self-organizing teams**
- The Team regularly **reflects** on work

Myths in Agile

- Myth#1 –Agile Development is undisciplined
- Myth#2 –Agile Teams do not plan
- Myth#3 –No Documentation, No Metrics
- Myth#4 –Compromise in Quality
- Myth#5 –Senior most people will lose their jobs
- Myth#6 –Self Organized Teams can do anything they want
- Myth#7 –Design and Architecture can be ignored

Agile Methodologies

- Scrum
- Scrum /XP Hybrid
- Custom Hybrid
- Don't know
- Kanban
- Scrumban
- Feature-Driven Development
- Extreme Programming XP
- Lean
- Agile Unified Process (AgileUP)
- Agile Modeling
- Dynamic System Development Method

Contribution to Agile

Lean Principles:

- Eliminating Waste
- Amplifying Learning
- Deciding as Late as Possible
- Delivering as Fast as Possible
- Empowering the Team
- Building Integrity In
- Seeing the Whole

Kanban:

- Visualize what you do today (workflow)
- Limit the amount of work in progress (WIP)
- Enhance flow

Actual use of requested feature: