Introduction to Agile

Lesson 1: Agile Process Framework

Lesson Objectives

- History of Traditional Software Development Model
- Agile Software Development – Definition
- Agile Development Model
- Graphical Illustration of Agile Development Model
- Why use Agile?
- Agile Manifesto and Principles
- 12 Principles of Agile Methods

Lesson Objectives

- Agile Values
- What is NOT an Agile software development?
- Foundation of an Agile software development Method
- Common Characteristics of Agile Methods
- Agile Methods and Practices
- When to use Agile Model?
- Advantages of Agile Model
- Disadvantages of Agile Model
- Difference between Agile and Waterfall Model
- Agile – Myths and Reality
- Agile Market Insight

History of Traditional Software Development Model

- Traditional software development methodologies are often called heavyweight methodologies as they are based on a sequential series of steps that has to be defined and documented in detail
- These methodologies are based on progressive series of steps like requirements definition, design and architectural planning, development and testing
- The traditional software development models depends upon a set of predefined processes
- The success of a project which is build with traditional software development model depends upon how well the requirements are stated before the project begins
- In this approach implementing changes during development life cycle is somewhat critical
- However, this approach also poses benefits like easy estimation of cost of the project, scheduling and allocation of resources

1.2: Agile Process Framework

Agile Software Development – Definition

- Wikipedia defines Agile Software Development as :

> Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen interactions throughout the development cycle.

Agile Software Development – Definition

- The web site, SearchSoftwareQuality.com contains the following definition:

> In software application development, agile software development (ASD) is a methodology for the creative process that anticipates the need for flexibility and applies a level of pragmatism into the delivery of the finished product. Agile software development focuses on keeping code simple, testing often, and delivering functional bits of the application as soon as they're ready. The goal of ASD is to build upon small client-approved parts as the project progresses, as opposed to delivering one large application at the end of the project.

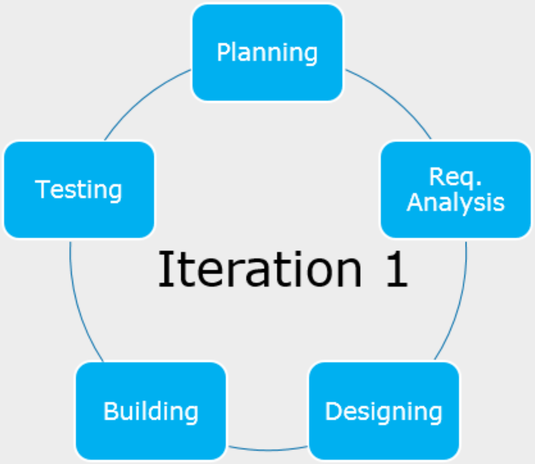
1.2: Agile Process Framework

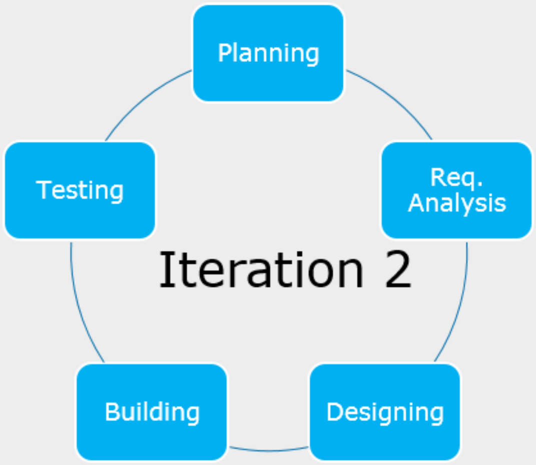
Agile Development Model

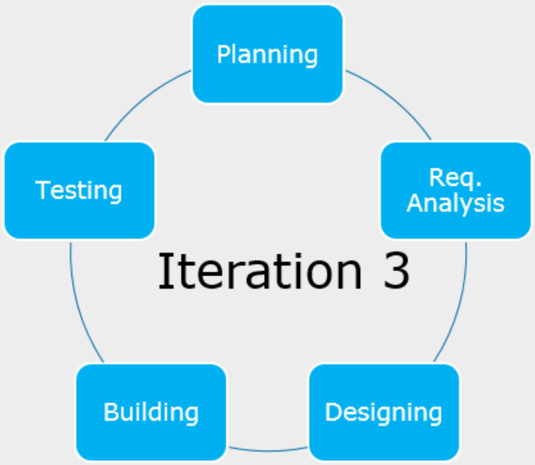
- Agile development model is a an amalgamation of iterative and incremental process models focusing more on process adaptability and customer satisfaction by rapid delivery of functional software product
- Agile development model breaks the software into small incremental builds
- These builds are provided in iterations
- Each iterations lasts from about one to three weeks
- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing
- At the end of the iteration a working product is displayed to the customer and important stakeholders

1.2: Agile Process Framework

Graphical Illustration of Agile Development Model







1.2: Agile Process Framework

Why use Agile?

- Improved return on investment (RIO)
- Early detection and cancellation of failing products
- Higher quality software
- Improved control of a project
- Reduced dependence on individuals and increased flexibility

1.2: Agile Process Framework

Agile Manifesto and Principles

Individuals and interactions **over processes and tools**

Working software **over comprehensive documentation**

Customer collaboration **over contract negotiation**

Responding to change **over following a plan**

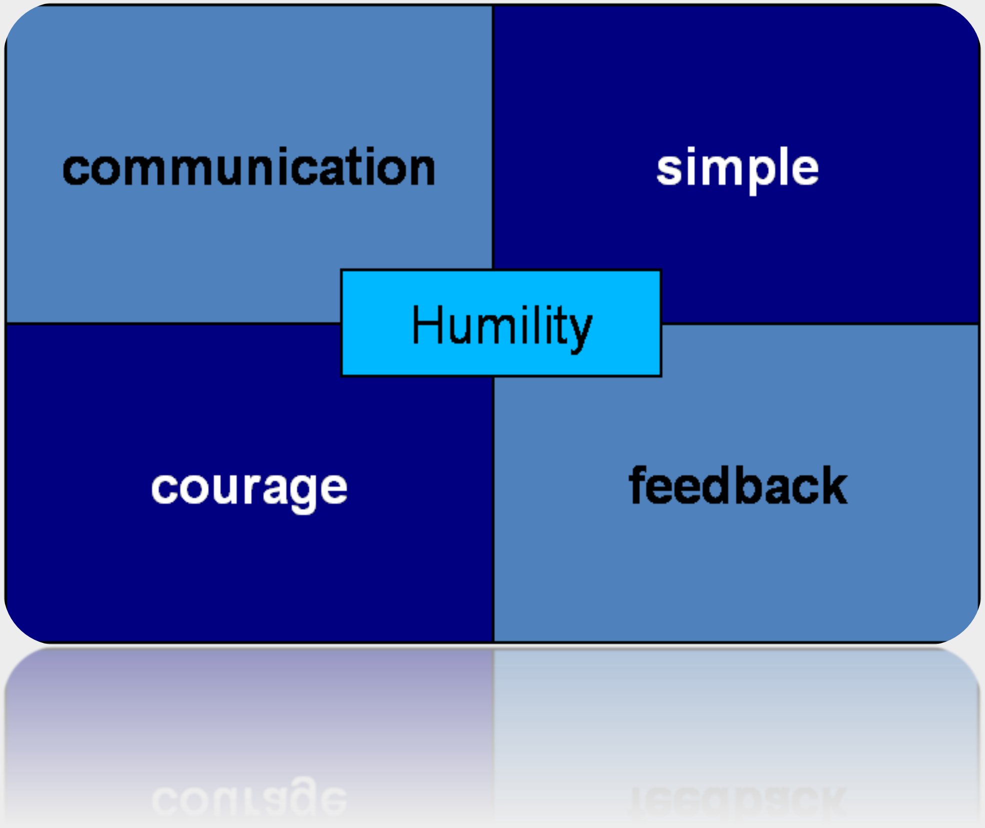
1.2: Agile Process Framework

12 Principles of Agile Methods

- According to Kent Beck, the Agile Manifesto is based on twelve principles:
  - Customer satisfaction by rapid delivery of useful software
  - Welcome changing requirements, even late in development
  - Working software is delivered frequently (weeks rather than months)
  - Working software is the principal measure of progress
  - Sustainable development, able to maintain a constant pace
  - Close, daily cooperation between business people and developers
  - Face-to-face conversation is the best form of communication (co-location)
  - Projects are built around motivated individuals, who should be trusted
  - Continuous attention to technical excellence and good design
  - Simplicity—the art of maximizing the amount of work not done - is essential
  - Self-organizing teams
  - Regular adaptation to changing circumstances

1.2: Agile Process Framework

Agile Values



communication

simple

Humility

courage

feedback

1.2: Agile Process Framework

What is NOT an Agile software development?

- Compressing the project schedule
- Eliminating all existing software development models
- Eliminating all documentation
- Writing code up to the last minute
- An excuse for doing nothing

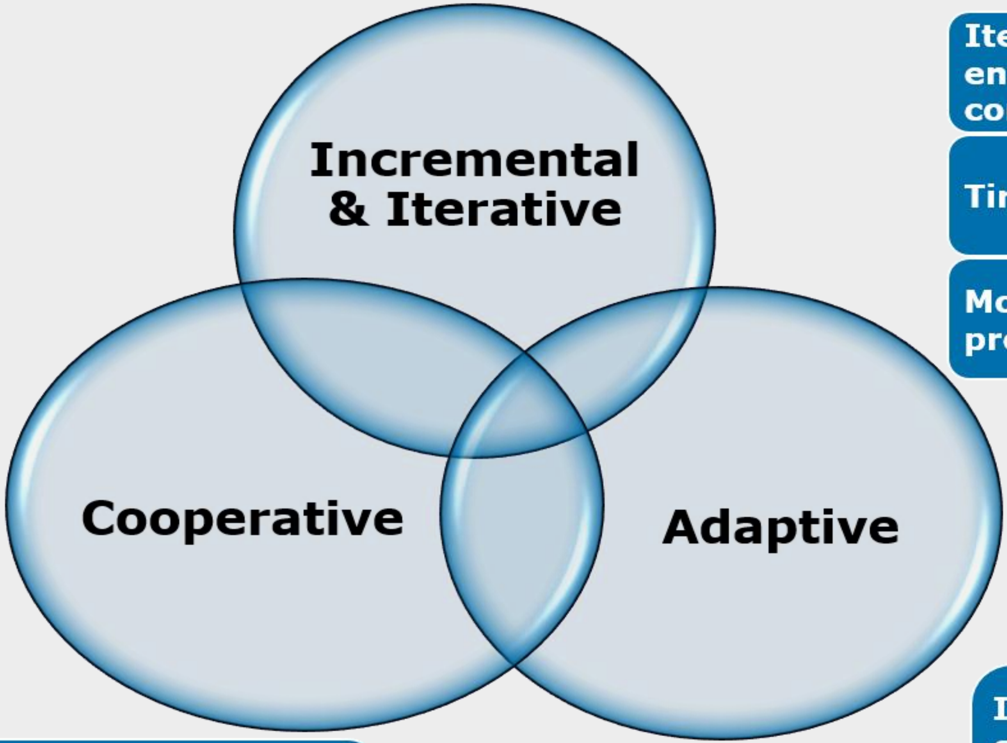
1.2: Agile Process Framework

Foundation of an Agile software development Method

- Agility
- Change
- Planning
- Communication
- Learning
- Team

1.2: Agile Process Framework

Common Characteristics of Agile Methods



Iterative with short cycles enabling fast verification & corrections

Time bound iteration cycles

Modularity at development process level

Incremental and convergent approach that minimizes risks and facilitates functional additions

People oriented

Collaborative and communicative working style

Agile Methods and Practices

- Scrum - Ken Schwaber, Jeff Sutherland, Mark Beedle
- Extreme Programming (XP) - Kent Beck, Eric Gamma, and others
- Dynamic System Development Method (DSDM) - Dane Faulkner And others
- Agile Unified Process (or Agile RUP) - Scott Ambler
- Feature Driven Development - Peter Coad and Jeff Deluca
- Lean Software Development - Mary and Tom Poppendieck
- Kanban - David Anderson

1.2: Agile Process Framework

When to use Agile Model?

- This model can be followed when:
  - New changes must be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
  - To implement a new feature, the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
  - Unlike the Waterfall Model, in the agile model, limited planning is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This gives the customer the finished system they want or need.
  - Both system developers and stakeholders alike, find that they also get more freedom of time and options than if the software was developed in a more rigid, sequential manner. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

Advantages of Agile Model

- Is a very realistic approach to software development
- Promotes teamwork and cross training
- Functionality can be developed rapidly and demonstrated
- Resource requirements are minimum
- Suitable for fixed or changing requirements
- Delivers early partial working solutions
- Good model for environments that change steadily
- Minimal rules, documentation easily employed
- Enables concurrent development and delivery within an overall planned context
- Little or no planning required
- Easy to manage
- Gives flexibility to developers

1.2: Agile Process Framework

Disadvantages of Agile Model

- Not suitable for handling complex dependencies
- More risk of sustainability, maintainability and extensibility
- An overall plan, an agile leader and agile PM practice is a must without which it will not work
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction
- There is very high individual dependency, since there is minimum documentation generated
- Transfer of technology to new team members may be quite challenging due to lack of documentation

Difference between Agile and Waterfall Model

Agile

- Software development lifecycle is carried out in the form of Sprints
- Agile method proposes incremental and iterative approach to software design
- It follows an incremental approach towards solution development
- Agile methodology is known for its flexibility
- Agile can be considered as a collection of many different projects

Waterfall

- Software development process is divided into distinct phases
- Development of the software flows sequentially from start point to end point
- It follows linear, sequential design approach towards software development
- Being a traditional software development model, Waterfall exhibits characteristic of a structured model so most of the times it can be very rigid
- Software development will be completed as one single project

1.2: Agile Process Framework

Agile – Myths and Reality

Myth	Reality
No Documentation	Agile Documentation
Undisciplined	Requires great discipline
No Planning	Just-in-time (JIT) planning
Not Predictable	Far more predictable
Is a Fad	It’s quickly becoming the norm
Silver Bullet	It requires skilled people
RUP isn’t agile	RUP is as agile as you make it
Not Fixed Price	Agile provides stakeholders control over the budget, schedule, and scope

1.2: Agile Market Insight

The 12th Annual State of Agile Report 2018 Source : COLLABNET VERSIONONE

Reasons for Adopting Agile

- **75%** Accelerate software delivery
- **64%** Manage changing priorities
- **55%** Increase productivity
- **49%** Better Business/IT Alignment
- **46%** Increased Software Quality

Benefits of Adopting Agile

- **71%** Manage Changing Priorities
- **66%** Project Visibility
- **65%** Business/IT Alignment
- **62%** Delivery speed/Time to market
- **61%** Team Productivity

Summary

- In this lesson, you have learnt
  - Various conventional software development models like Waterfall Model, SDLC & V-Model
  - The traditional software development models like Waterfall Model, V-Model are classified into the heavyweight methodologies
  - These methodologies are based on progressive series of steps like requirements definition, design and architectural planning, development and testing
  - Every traditional software development model has its own advantages and disadvantages
  - We need to select the software development model which best suits our project requirement
  - Introduction to Agile
  - Agile Manifesto and Principles
  - Agile Values
  - Agile – Myths and Reality
  - Agile Market Insight