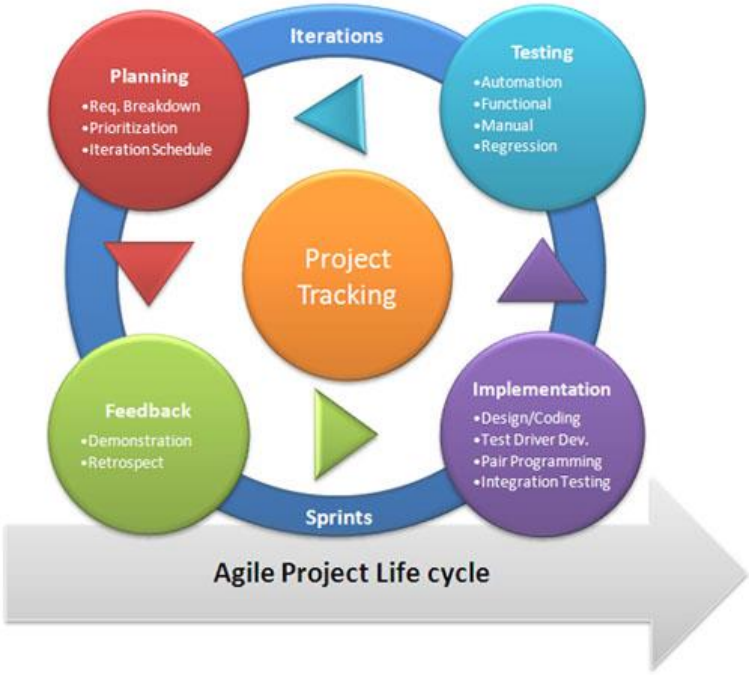

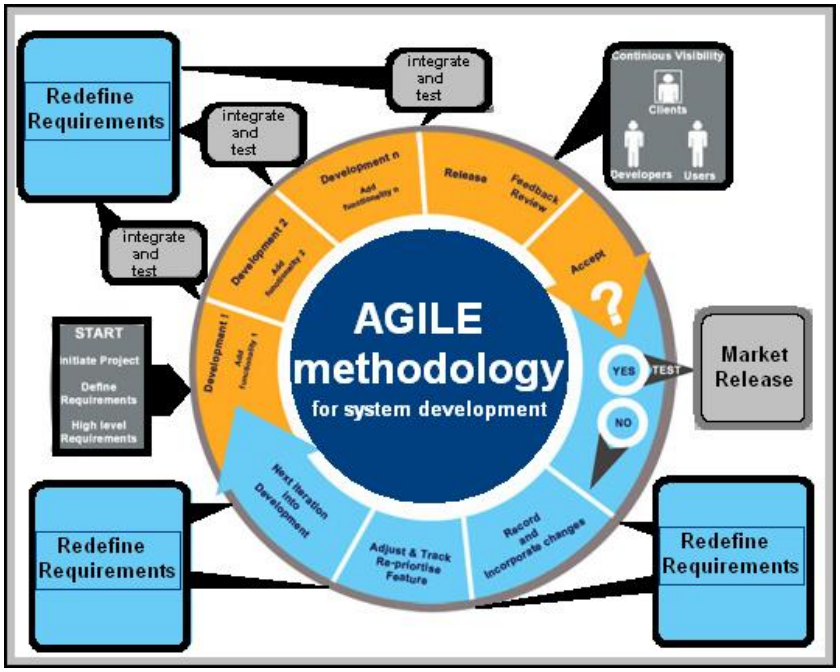

Lecture 2: Agile Software Development (Part 1)

Key Principles and Variants

Overview of Lecture

- Traditional Software Development Methodology
- Agile Software Development
- The Agile Manifesto
- Agile Methods
- Comparison of Agile Methods

Traditional Software Development Methodology

- Classical Waterfall Model
- Iterative Waterfall Model
- Prototyping Model
- Evolutionary Model
- Spiral Model
- V-Model
- Rapid Application Development Model

Weakness of Traditional Software Development Methodology

- Process control or documentation oriented methods like structured analysis and design
- Traditional, hard development tools like entity modelling and data flow diagramming do not take the disorganised world of people into consideration
- The main problems of the traditional development methods are their inability to face challenges set by changing organisational, business and technical environment and their insufficient emphasis on individuals and individual talent and creativity
- Traditional methods are often considered bureaucratic and restrictive

Weakness of Traditional Software Development Methodology

- It is hard to predict in advance:
  - The number of changes and priority changes in software requirements
  - How much design is needed before construction is used to prove it
  - The time to analysis, design, construction and testing
- Solution?
  - Incremental development

Agile Software Development

- Agile software development is a group of software development methods based on iterative and incremental development.
- 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 interactions throughout the development cycle.

Agile Software Development

- The term *agile* can be defined as
  1) marked by ready ability to move with quick easy grace, or
  2) having a quick resourceful and adaptable character (Merriam-Webster 2002)

- "Agility, for a software development organisation, is the ability to adopt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment. An agile process is one that readily embraces and supports this degree of adaptability. So, it is not simply about the size of the process or the speed of delivery; it is mainly about flexibility." (Kruchten 2001, 27)

Agile Software Development

Agile development is a methodology of software development that emphasizes customer satisfaction through continuous delivery of Functional software. Based on a variety of iterative development disciplines Agile methods put developers to work in small teams to tight budgets and short timescales.

- Agile methods are considered
  - Lightweight
  - People-based rather than Process-based
  - Adaptive rather than predictive
  - Less document oriented than rigorous methods

Agile Software Development

- Principles
  - Communicate better
  - Competence ; use of extreme tools
  - Working Functionalities over Documentation
  - Change Acceptance

- Simply Put
  - Short Releases
  - Small Releases
  - Simple Design
  - Continuous Testing

Agile Software Development

- Characteristics for fast, light and agile processes are for instance:
  - short software development (3-6 months)
  - light development methods and informal communication
  - heavy information systems not used
  - adaptive, suits different environments
  - non-bureaucratic working environment
  - high quality requirements
  - close customer relationships through the development process
- Characteristics
  - Modularity
  - Iterative
  - Time-bound
  - Incremental
  - People-oriented
  - Collaborative

Manifesto for Agile Software Development

- The Agile Alliance is a non-profit organisation dedicated to promoting the concepts of agile software development, and helping organisations adopt those concepts (Agile Alliance 2002)
- Agile Alliance was formed by seventeen professional software developers practicing lightweight approaches to software development
  - Representatives of different agile methods, such as Extreme Programming (XP), Scrum and Crystal Family
- Their aim was to discuss alternatives to rigorous, documentation driven software development
- The discovered concepts are outlined in Agile Manifesto

Manifesto for Agile Software Development

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

Through this work we have come to 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 there is value in the items on the right, we value the items on the left more.

Manifesto for Agile Software Development

- Individuals and interactions over processes and tools
  - Agile methods reject the assumption that people who are involved in software development are replaceable parts
  - Although process descriptions and organisation charts are needed to get the project started, Agile Alliance wants to emphasise individual people over roles and encourage interaction between individuals
  - Interaction and communication between people are frequently addressed issues in agile software development

Manifesto for Agile Software Development

- Working software over comprehensive documentation
  - Documents containing requirements, analysis or design can be very useful to guide developer's work and help to predict the future
  - However, working code that has been tested and debugged reveals information about the development team, the development process and the nature of problems to be solved
  - Running program is the only reliable measure of the speed and shortcomings of the team and gives a glimpse into what the team should really be building

Manifesto for Agile Software Development

- Customer collaboration over contract negotiation
  - Emphasis on close relationships between the software development team and the customer
  - Agile Alliance suggests that fruitful and close collaboration can make contracts unnecessary and if a contract situation is in jeopardy, good collaboration can save the situation
  - The basic assumption behind this value statement is customer satisfaction in general, which is a main driver in agile software development

Manifesto for Agile Software Development

- Responding to change over following a plan
  - Plans are useful and planning is included in agile methods, which also have mechanisms for dealing with changing requirements
  - However, instead of following a plan rigorously, development teams should constantly reflect the plan to the current situation and change it accordingly

Principles behind the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

Principles behind the Agile Manifesto

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Principles behind the Agile Manifesto

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Human Factors

- An agile team recognizes that software is developed by individuals working in teams and that the skills of these people, their ability to collaborate is at the core for success of the project.
- Human Factors
  - Competence
  - Common focus
  - Collaboration
  - Decision-making ability
  - Fuzzy problem-solving ability
  - Mutual trust and respect
  - Self-organization

Why Agile Methods?

- "Agile" – who doesn't want to be "Agile"
  - Implies low cost
  - Implies Speed
  - Implies Ease
- It's Appealing (fast; nimble; almost fun)
- Perception is Reality
  - Not Bureaucratic
  - Not Strictly Engineered
  - Adaptive to Change
  - Learning Teams
  - Small (effective) Units

Key benefits of Agile

- Increases Flexibility - Minimizes up-front investment and maximizes return on investment (ROI) by creating an efficient IT development process
- Delivers the right solution - Aligns users and stakeholders with the right people to deliver the solution that the business actually needs
- Accelerates Delivery - Iterations get to the right solution faster
- Reduces risk and increases Quality - Greater stakeholder visibility and control

Disadvantages of Agile

- It's hard!
- Makes all dysfunction visible
  - Its doesn't fix anything: the team has to do it
  - Feels like things are worse at the beginning
- Bad products will be delivered sooner, and doomed projects will fail faster
- Partial adoption may be worse than none at all

Agile Methodologies

- Several methods that are often cited to be agile, e.g.,
  - Extreme Programming
  - Crystal Family
  - Open Source
  - Adaptive Software Development (ASD)
  - SCRUM
  - Feature Driven Development (FDD)
  - Dynamic System Development Method (DSDM)
- In addition, e.g., Rational Unified Process (RUP) and Capability Maturity Model (CMM) can be evaluated from Agile Manifesto point of view
- Further, organisations often develop their own methods, or modify existing methods to better suit their objectives
  - These are called local method development or in-house methods

SCRUM

- Techniques
  - Team creation
  - Backlog creation
  - Project segmentation
  - Scrum meetings
  - Burndown charts
- Phases
  - Review release plans
  - Distribution, review and adjustment of product standards
  - Sprint
  - Sprint review
  - Closure

Links

- Abrahamsson P, Salo O and Ronkainen J. Agile software development methods (Review and analysis).
- http://agilemethodology.org/
- http://agilemanifesto.org/
- http://www.agilealliance.org/
- http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm
- http://agilemanifesto.org/