Agile Software Development

Lecturer: Raman Ramsin

Lecture 1

Agile Development: Basics
Software Development Methodology (SDM)

- A framework for applying software engineering practices with the specific aim of providing the necessary means for developing software-intensive systems

- Consisting of two main parts:
  - A set of modeling conventions comprising a Modeling Language (syntax and semantics)
  - A Process, which
    - provides guidance as to the order of the activities,
    - specifies what artifacts should be developed using the Modeling Language,
    - directs the tasks of individual developers and the team as a whole, and
    - offers criteria for monitoring and measuring a project’s products and activities.
Object-Oriented Software Development Methodology (OOSDM)

- Specifically aimed at viewing, modeling and implementing the system as a collection of interacting objects
- First appeared in late 1980s
- Categorized as
  - Seminal (First and Second Generations)
  - Integrated (Third Generation)
  - Agile
- UML was the result of the ‘war’ among seminal methodologies
- Process has now replaced modeling language as the main contentious issue
Agile Development: Brief History


- The once-common perception that agile methodologies are nothing but controlled code-&-fix approaches, with little or no sign of a clear-cut process, is only true of a small – albeit influential – minority.

- Essentially based on practices of program design, coding and testing that are believed to enhance software development flexibility and productivity.

- Most agile methodologies incorporate explicit processes, although striving to keep them as lightweight as possible.
Major Agile Methodologies

- Scrum (1995)
- FDD – Feature-Driven Development (1999)
Agile Methodologies: Evolution Map

[Abrahamsson et al. 2003]
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.
Agile Methodologies: Principles

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

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

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

- Business people and developers must work together daily throughout the project.
Agile Methodologies: Principles (Contd.)

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

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

- Working software is the primary measure of progress.

- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Agile Methodologies: Principles (Contd.)

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

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

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

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

