CS 350 – Software Engineering

October 4, 2017

Week 7, Class #19

Software Process

Class Transition

- Mark Seaman
  - Office Hours – MWF – 10:00-11:30 - Kepner 0095F
- Goals
  - Each week will have a specific topic theme
  - Project will convert to Exercises for Applying Knowledge
  - Each class will have an Application that you will do as homework

This Week

- Monday, 10-2
  - Lecture – Software Engineering
  - Learn about class project
- Wednesday, 10-4
  - Lecture – Software Process
  - Class communications
- Friday, 10-6
  - Lecture – Software Project Management

Software Engineering

- Science – Discovery, Engineering – Invention

- Engineering Knowledge (theory)
  - Engineering
    - Requirements, Design, Code, Test
  - Technology
    - Tools, Tricks

- Software Process (practice)
  - How we apply engineering knowledge

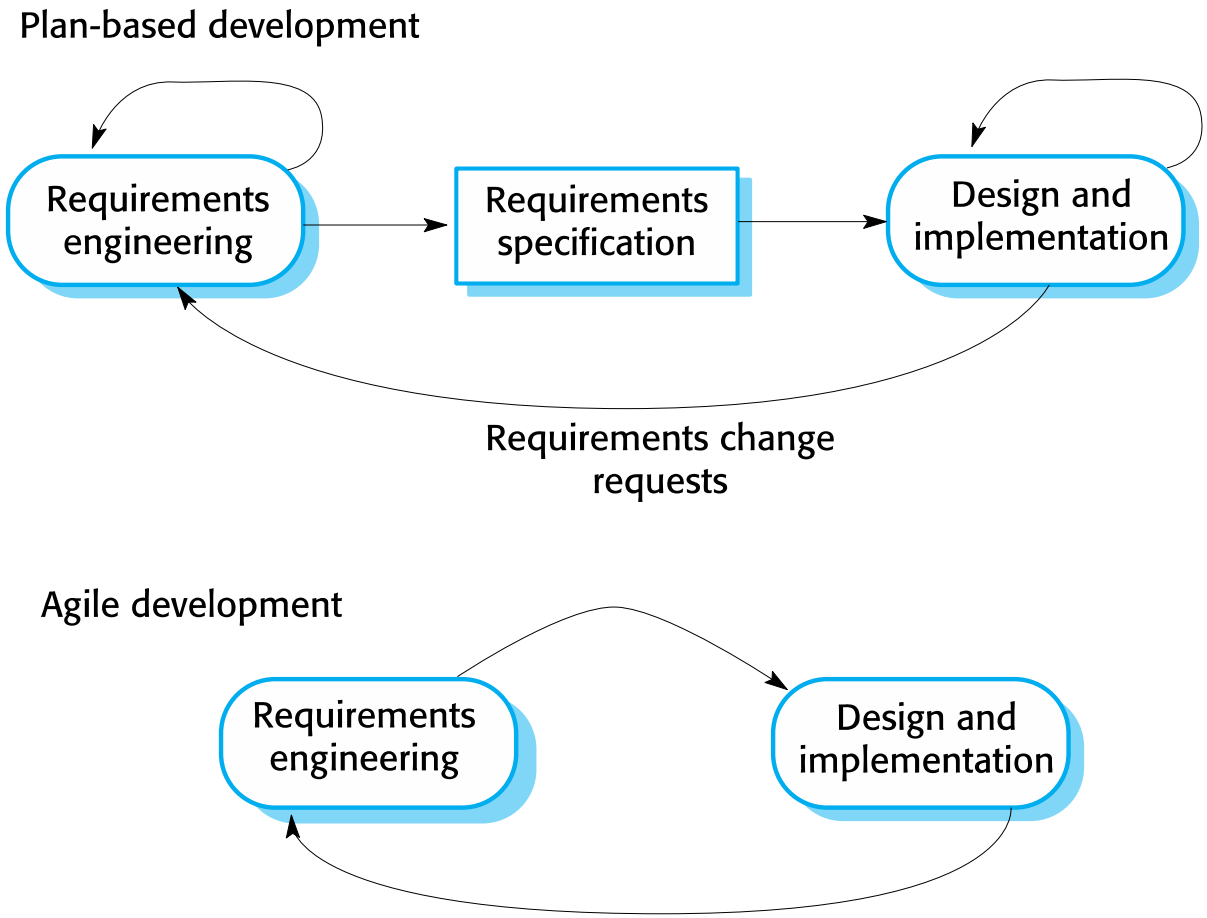
Software Lifecycle

- Software Process – how we do things
  - Project Plan - Requirements
  - Design
  - Code
  - Test

Case Study

- ✧ Late 90s – Dotcom boom
- ✧ Digital Camera & Scanner Software
- ✧ 5 engineers – create imaging apps
  - Unload the camera photos
  - Scan slides, negatives, print
  - Page layout for photos
  - Color enhancement
  - Upload photos to web service
  - Email photos
  - Application install and config

Plan-driven and agile development



Agile development

- Program specification, design and implementation are inter-leaved
- The system is developed as a series of versions or increments with stakeholders involved in version specification and evaluation
- Frequent delivery of new versions for evaluation
- Extensive tool support (e.g. automated testing tools) used to support development.
- Minimal documentation – focus on working code

Agile methods

- Dissatisfaction with the overheads involved in software design methods of the 1980s and 1990s led to the creation of agile methods. These methods:
  - Focus on the code rather than the design
  - Are based on an iterative approach to software development
  - Are intended to deliver working software quickly and evolve this quickly to meet changing requirements.
- The aim of agile methods is to reduce overheads in the software process (e.g. by limiting documentation) and to be able to respond quickly to changing requirements without excessive rework.

Agile manifesto

- *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.*

The principles of agile methods

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Customer involvement

- The role of the customer in the testing process is to help develop acceptance tests for the stories that are to be implemented in the next release of the system.
- The customer who is part of the team writes tests as development proceeds. All new code is therefore validated to ensure that it is what the customer needs.
- However, people adopting the customer role have limited time available and so cannot work full-time with the development team. They may feel that providing the requirements was enough of a contribution and so may be reluctant to get involved in the testing process.

x

- X
- Y