Software Develop Process

- Specification (software / system requirements)
- Design and implementation
- Validation (testing)
- Evolution

Goal

Intend to develop systems more quickly with limited time spent on analysis and design

Agile Manifesto

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

WaterFall Model

Development activities	Teams
Divide the work into stages	A separate team of specialists for each stage
At each stage, the work is passed from one team to another	Some coordination is required for the handoff from team to team - using "documents"
At the end of all of the stages, you have a software product ready to ship	As each team finishes, they are assigned to a new product

Agile Development Model

Documentation

- Still important in an Agile development
- If it is the only kind of communication in your project, it isn't good
- Real working code is more valuable than document

Development plans

- Still important in an Agile project
- the format of an Agile development schedule is a bit different
- Development plan includes "iterations"
- Each iteration gives the team a chance to incorporate what they learn

Agile Principles

1. Our highest priority is to **satisfy the customer** through early and **continuous delivery of valuable software.**	5. Build projects around **motivated individuals.** Give them the **environment** and support they need, and **trust** them to get the job done.	9. **Continuous attention** to technical excellence and good design enhances agility.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.	6. The most efficient and effective method of **conveying information** to and within a development team is **face-to-face conversation.**	10. **Simplicity**--the art of maximizing the amount of work not done--is essential.
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the **shorter timescale.**	7. **Working software** is the primary **measure of progress.**	11. The best architectures, requirements, and designs emerge from **self-organizing teams.**
4. Business people and developers must **work together daily** throughout the project.	8. Agile processes promote **sustainable development.** The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.	12. At regular intervals, the **team reflects** on how to become more **effective**, then **tunes and adjusts its behavior** accordingly.

Waterfall Model Phases

- Requirement's analysis and definition
  - Deliverable: Requirements doc.
- System and software design
  - Deliverable: Design document based on requirements doc.
- Implementation and unit testing
  - Deliverable: Code and test it for system components(using design doc.)
- Integration and system testing
  - Deliverable: Software components are integrated and the resulting system is tested
- Operation and maintenance

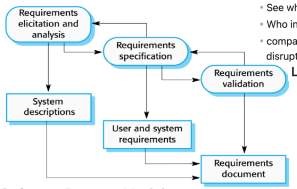
Advantages

- Easy to understand and implement
- Identified deliverables and milestones

Disadvantages

- Intensive documenting and planning
- Discovering issues in later phases should lead to returning to earlier phase

Requirements Engineering Process



Version Control

A method for recording changes to a file or set of files over time so that you can recall specific versions later

Version Control System (VCS)

Software tools to software teams to manage changes to source code over time

- Keep track of every modification to code in a repository
- Revert selected files back to a previous state
- Compare changes over time
- See who last modified something that might be causing a problem
- Who introduced an issue and when
- compare earlier versions of the code to help fix bugs while minimizing disruption to all team members

Local Version Control

Revision Control System (RCS) works by keeping patch sets (i.e., the differences between files)

Centralized Version Control (CVC)

- A single server contains all versioned files and a number of clients check-out files from it
- Single point of failure
  - Developer's work interrupted!
  - Hard disk becomes corrupted, and no proper/up-to-date backups? entire history lost!

Distributed Version Control (DVC)

- Developers fully mirror the repository including the full history
- Several remote repositories
  - Developers can collaborate with different groups of people in different ways simultaneously with the same project
  - Can setup several types of workflows (not possible in CVC)

Software Process Models

- Waterfall Models
- Spiral Models
  - Incremental Development risk-driven
  - Incremental: Add new functionalities
- Agile Model
  - Iterative incremental process for rapid software development
  - Iterative: Refining and Improving the existing system
- The Rational Unified Process (RUP or UP)
  - Bring together elements of different process models
  - Phases of the model in timer, process activities, good practices

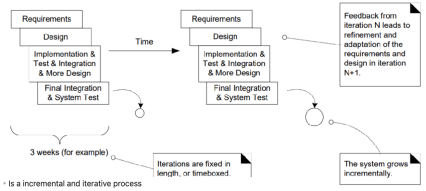
Planning in Software Development

- **Plan-driven** (plan-and-document / heavy-weight)
  - Activities are planned in advance and progress is measured against this plan
  - Plan drives everything and change is expensive
- Agile processes (light-weight)
  - Planning is incremental and continual as the software is developed
  - Easier to change to reflect changing requirements
- Most Software processes include elements of both plan-driven and agile
- Each approach is suitable for different types of software
  - No right or wrong software processes

Planning in Waterfall Model

- Difficulty of accommodating change after the process is underway
- Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements
- Mostly used for larger systems engineering projects where a system is developed at several sites
  - The plan-driven nature of the waterfall model helps coordinate the work

Rational Unified Process (RUP or UP)



- Is a incremental and iterative process
- Software development process utilizing iterative and risk-driven approach to develop OO software systems
- Iterative incremental development
- Iterative evolutionary development