

Agenda

- Announcements
- Lecture
 - Software Processes

Announcements

- No lecture on Tuesday
- For Thursday, read “No Silver Bullet”

- DCS Mentorship Program
 - Application deadline November 15
 - <http://www.cs.toronto.edu/dcs/mentorship>

Software Process

- A Software Process Model is a simplified representation of the software process, presented from a specific perspective.
 - General and abstract
- Software Process is a set of activities whose goal is the development or evolution of software
 - Specific and enacted

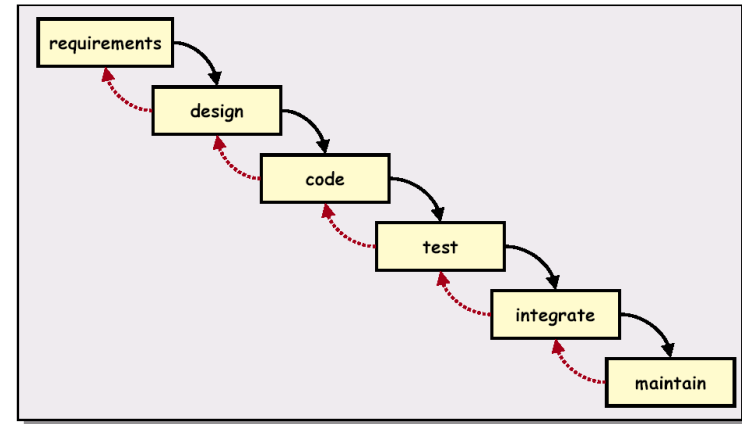
- Like the difference between class and object/instance

Software Process Models

Dimensions of Variation

- Phased or incremental
- Plan-based or iterative
- Continuous testing or late testing
- Feedback
- Risk management

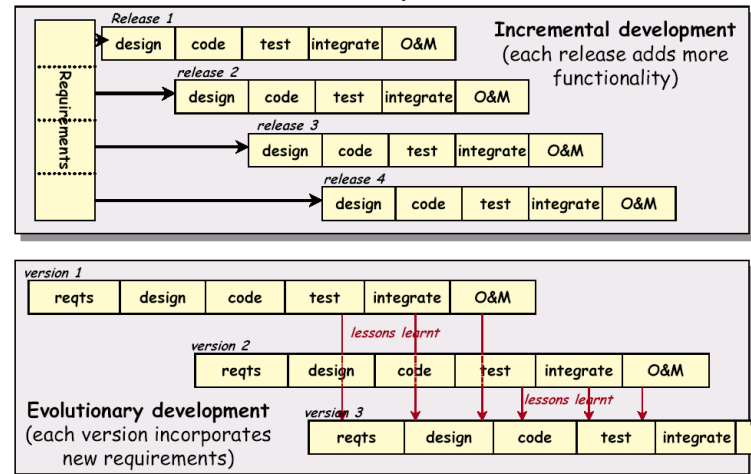
Waterfall Model



Waterfall Model

- **Points in Favor**
 - Some things must occur before others in the process, it makes sense to have code before test, requirements before design etc.
 - It works as a model because it's essentially true
- **Points Against**
 - No project or design has ever proceeded this way
 - Very difficult to lock down details before proceeding to next step

Evolutionary Model



Evolutionary Model

- **Points in Favor**
 - Accommodates throw-away prototyping
 - Allows for lessons from each version to be incorporated into the next
- **Points Against**
 - Hard to plan for versions beyond the first
 - Lessons may be learned too late
 - Process is not visible
 - Systems are often poorly structured
 - Special tools and techniques may be required

Spiral Model

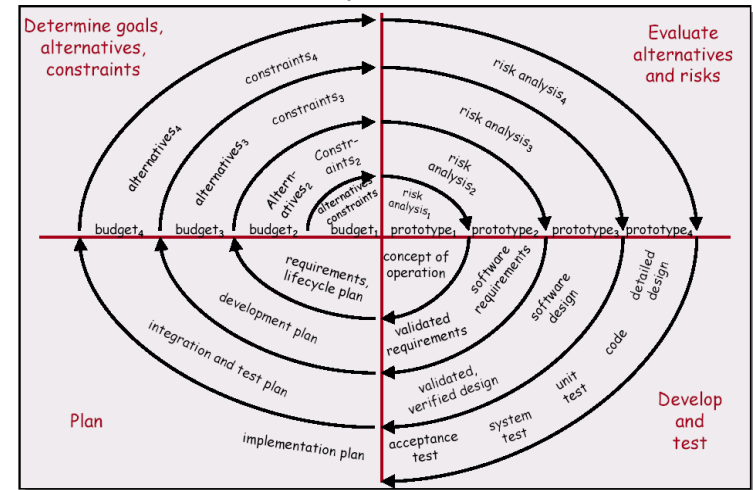


Diagram © Steve Easterbrook, University of Toronto
CSC301, Winter 2013

Spiral Model

- **Points in Favor**
 - Incorporates prototyping and risk analysis
- **Points Against**
 - Cannot cope with unforeseen changes (e.g. new business objectives)
 - Not clear how to analyze risk

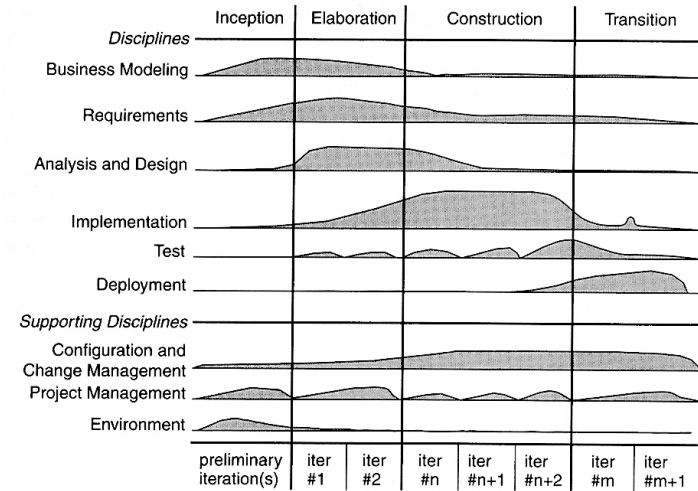
Question

- What is the difference between iterative and incremental?

Rational Unified Process

- RUP is a software (development | design) process that is:
 - Use case-driven
 - Architecture-centric
 - Iterative and incremental

Rational Unified Process



UP Phases

- **Inception**
 - Approximate vision, business case, scope, vague estimates
- **Elaboration**
 - Refined vision, iterative implementation of the core architecture, resolution of high risks, identification of most requirements and scope, more realistic estimates
- **Construction**
 - Iterative implementation of lower risk and easier elements, preparation for deployment
- **Transition**
 - Beta tests, deployment

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.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Agile is not entirely new

- Iterative and incremental process models have existed for a long time
 - Spiral model by Barry Boehm (1985)
- Evolutionary software development
 - Mentioned by Fred Brooks in “No Silver Bullet” (1987)
- Frequent deliveries and feedback
 - EVO by Tom Gilb (1985)

Current Differences

- Popularity
 - Initiated by software developers
 - Now taken up by executives
- New Techniques and Tools
 - Test-driven development
 - Refactoring
 - User stories
- Growing Community with Shared Terminology
- Infrastructure
 - Coaches, training, certification, courses, conferences

Current Reasons for Popularity

- Effectiveness
- Results
- Compatibility with web applications
- Cool factor

