CS 320
Fundamentals of Software Engineering

Lecture 4: Software Processes (2)
Software Lifecycle

inception  pre-development  development  post-development
In an ideal world...
But...
Coping with Change

- Change is inevitable in all large software projects.
  - **Business changes** lead to new and changed system requirements
  - **New technologies** open up new possibilities for improving implementations
  - **Changing platforms** require application changes

- Change leads to rework so the costs of change include both rework (e.g. reanalyzing requirements) as well as the costs of implementing new functionality
Cost of Change Progressively Higher
Reducing the Cost of Rework

- Change avoidance
  - The software process includes activities that can anticipate possible changes before significant rework is required

- Change tolerance
  - The software process is designed so that changes can be accommodated at relatively low cost
A prototype is an initial version of a system used to demonstrate concepts and try out design options.

A prototype can be used in:

- The requirements engineering process to help with requirements elicitation and validation
- In design processes to explore options and develop a UI design
Benefits of Prototyping

* Improved system usability
* A closer match to users’ real needs
* Improved design quality
* Improved maintainability
* Reduced development effort
Prototyping

- **Evolutionary prototyping**
  - An approach to system development where an initial prototype is produced and refined through a number of stages to the final system

- **Throw-away prototyping**
  - A prototype which is usually a practical implementation of the system is produced to help discover requirements problems and then discarded. The system is then developed using some other development process
Incremental Development and Delivery

- Incremental development
  - Develop the system in increments and evaluate each increment before proceeding to the development of the next increment;
  - Normal approach used in agile methods;
  - Evaluation done by user/customer proxy.

- Incremental delivery
  - Deploy an increment for use by end-users;
  - More realistic evaluation about practical use of software;
  - Difficult to implement for replacement systems as increments have less functionality than the system being replaced.
Incremental Delivery

1. Define outline requirements
2. Assign requirements to increments
3. Design system architecture
4. Develop system increment
5. System incomplete?
6. Validate increment
7. Integrate increment
8. Validate system
9. Deploy increment
10. System complete?
11. Final system
Incremental Delivery Advantages

- Customer value can be delivered with each increment so system functionality is available earlier.
- Early increments act as a prototype to help elicit requirements for later increments.
- Lower risk of overall project failure.
- The highest priority system services tend to receive the most testing.
Incremental Delivery Problems

* Most systems require a set of basic facilities that are used by different parts of the system.

* As requirements are not defined in detail until an increment is to be implemented, it can be hard to identify common facilities that are needed by all increments.

* The essence of iterative processes is that the specification is developed in conjunction with the software.

* However, this conflicts with the procurement model of many organizations, where the complete system specification is part of the system development contract.
Spiral Model

- Process is represented as a spiral rather than as a sequence of activities with backtracking.

- Each loop in the spiral represents a phase in the process.

- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.

- Risks are explicitly assessed and resolved throughout the process.
Spiral Model

- Objective setting
  - Specific objectives for the phase are identified.

- Risk assessment and reduction
  - Risks are assessed and activities put in place to reduce the key risks.

- Development and validation
  - A development model for the system is chosen which can be any of the generic models.

- Planning
  - The project is reviewed and the next phase of the spiral is planned.
Example
Example
Example
Example
Example
Example
Example
Example
Example