CSE 333 – Software Engineering

Pranta Sarker
Lecturer
Dept. of CSE, NEUB

Lecture 06

Outline

- Software process models
- Software process activities
- Coping with change
- The Rational Unified process

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., re-analyzing requirements) as well as the costs of implementing new functionality

Reducing the costs of rework

- Change avoidance, where the software process includes activities that can anticipate possible changes before significant rework is required.
  - For example, a prototype system may be developed to show some key features of the system to customers.

- Change tolerance, where the process is designed so that changes can be accommodated at relatively low cost.
  - This normally involves some form of incremental development. Proposed changes may be implemented in increments that have not yet been developed. If this is impossible, then only a single increment (a small part of the system) may have been altered to incorporate the change.

Software prototyping

- A prototype is an initial version of a system used to demonstrate concepts and try out design options.
- Software prototyping is the process in which developers create a model of the actual software.
- The intention behind creating this model is to get the actual requirements more deeply from user.
- 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;
  - In the testing process to run back-to-back tests.

Benefits of prototyping

- Improved system usability.
- A closer match to users' real needs.
- Improved design quality.
- Improved maintainability.
- Reduced development effort.

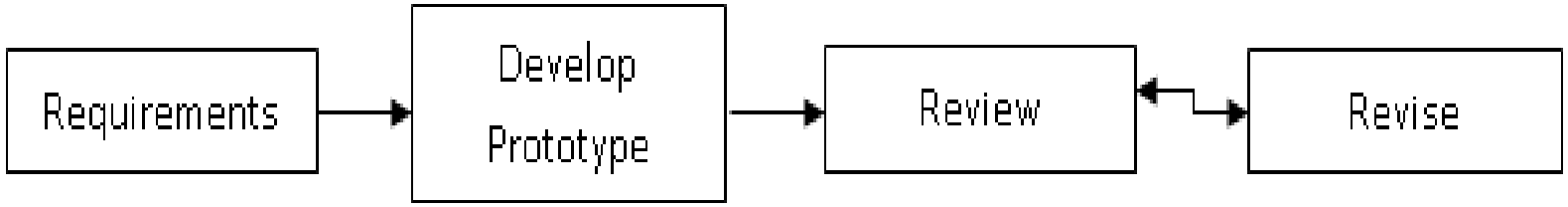
Software prototyping process

- Identify the initial requirements: In this step, the software publisher decides what the software will be able to do. The publisher considers who the user will likely be and what the user will want from the product, then the publisher sends the project and specifications to a software designer or developer.
- Develop initial prototype: The developer will consider the requirements as proposed by the publisher and begin to put together a model of what the finished product might look like. An initial prototype may be as simple as a drawing on a whiteboard, or it may consist of sticky notes on a wall, or it may be a more elaborate working model. Prototype may be a map/design/abstract view/summery of the actual software.

Software prototyping process

- Review: Once the prototype is developed, the publisher has a chance to see what the product might look like; how the developer has envisioned the publisher's specifications. In more advanced prototypes, the end consumer may have an opportunity to try out the product and offer suggestions for improvement. This is what we know of as beta testing.
- Revise: The final step in the process is to make revisions to the prototype based on the feedback of the publisher and/or beta testers.

Software prototyping process



Types of Software prototyping

- Rapid/Throwaway Prototyping: In this type we build a prototype with very little efforts to get the requirements from the customer. After getting the requirements we throwaway the prototype and start building the actual software.
- Evolutionary Prototyping: The prototype on the top of which we can build the whole actual system.

Thank you!!