CISC/CMPE 327 Software Quality Assurance

Queen's University, 2020-fall

Lecture #2
Prototyping model

The Prototyping Model

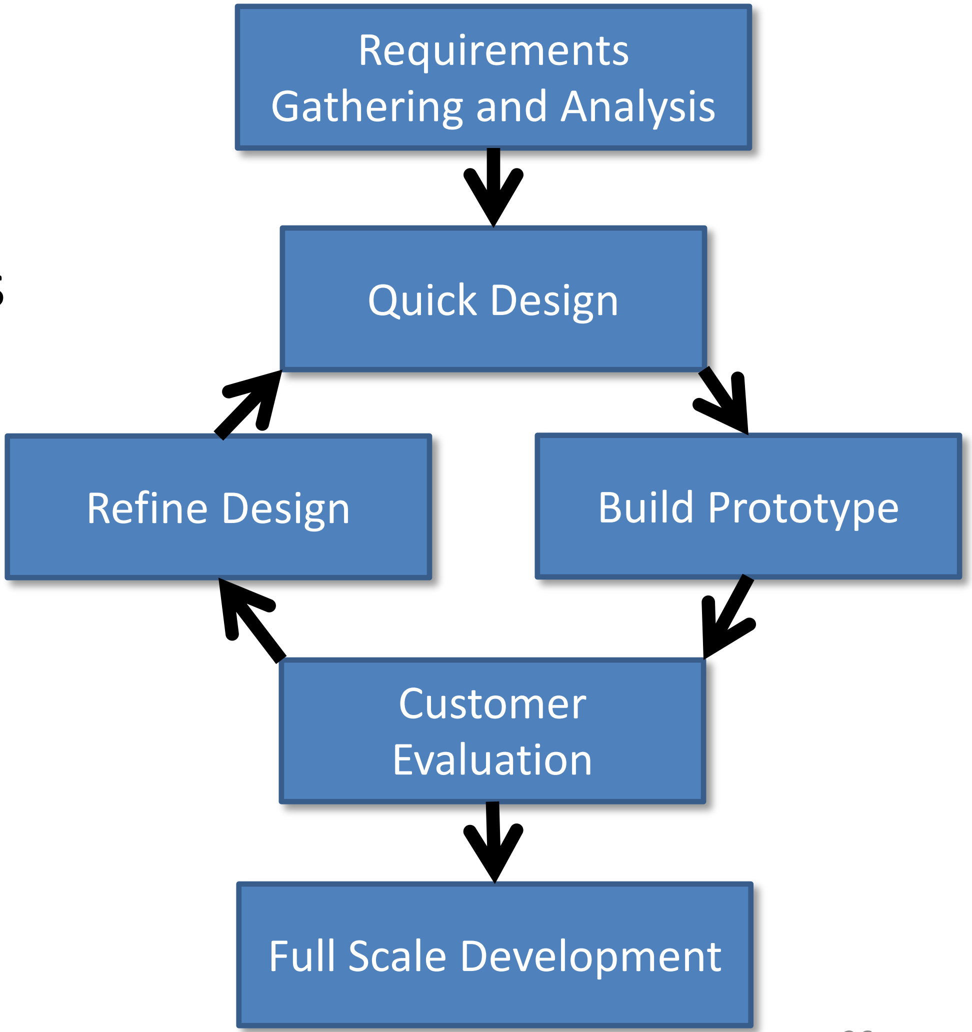
- Problems with Requirements
  - First step in the waterfall is requirements gathering and analysis
  - In practice, this is the most difficult part, and experience with the waterfall indicates that most failures are due to inadequate requirements understanding
  - Users often change requirements as they see what can be done

The Prototyping Model

- Prototyping
  - The prototyping model attempts to address the requirements difficulty by introducing an iterative, by example requirements stage
  - A prototype is a partial implementation of a software system with all external interfaces presented
  - Users use the prototype and provide feedback from which real requirements are gradually refined
  - Final prototype serves as example of intended system

The Prototyping Model

- Prototyping Model
  - Extend requirements phase to include a sequence of prototypes
  - Improve requirements and design as prototypes refined
  - When users and developers are both satisfied, move on to real development



The Prototyping Model

- 1. Requirements Gathering and Analysis
  - Much like waterfall model, but less stringent since prototype will help expose inadequacies
  - Quality control
    - Requirements reviews (inspection)
- 2. Quick Design
  - Make a simple approximate initial design, refine during prototype iteration
  - Quality control
    - Prototype testing

The Prototyping Model

- 3. Build Prototype
  - Quickly hack together an approximate implementation showing salient external features
  - Quality control
    - Essentially none
- 4. Customer Evaluation
  - Users validate prototype, report inadequacies
  - Quality control
    - Acceptance testing and evaluation (inspection)

The Prototyping Model

- 5. Design Refinement
  - Refine design in response to user feedback from prototype
  - Quality control
    - Design reviews (inspection)
- 6. Full Scale Development
  - Remaining stages of traditional waterfall model

Drawbacks of Prototyping Model

- Wasted Work
  - Prototypes are normally built using substandard quality controls ("thrown together") to speed the iteration ("quick turnaround")
  - Thus they must be discarded after the prototyping phase, even if they solve significant problems

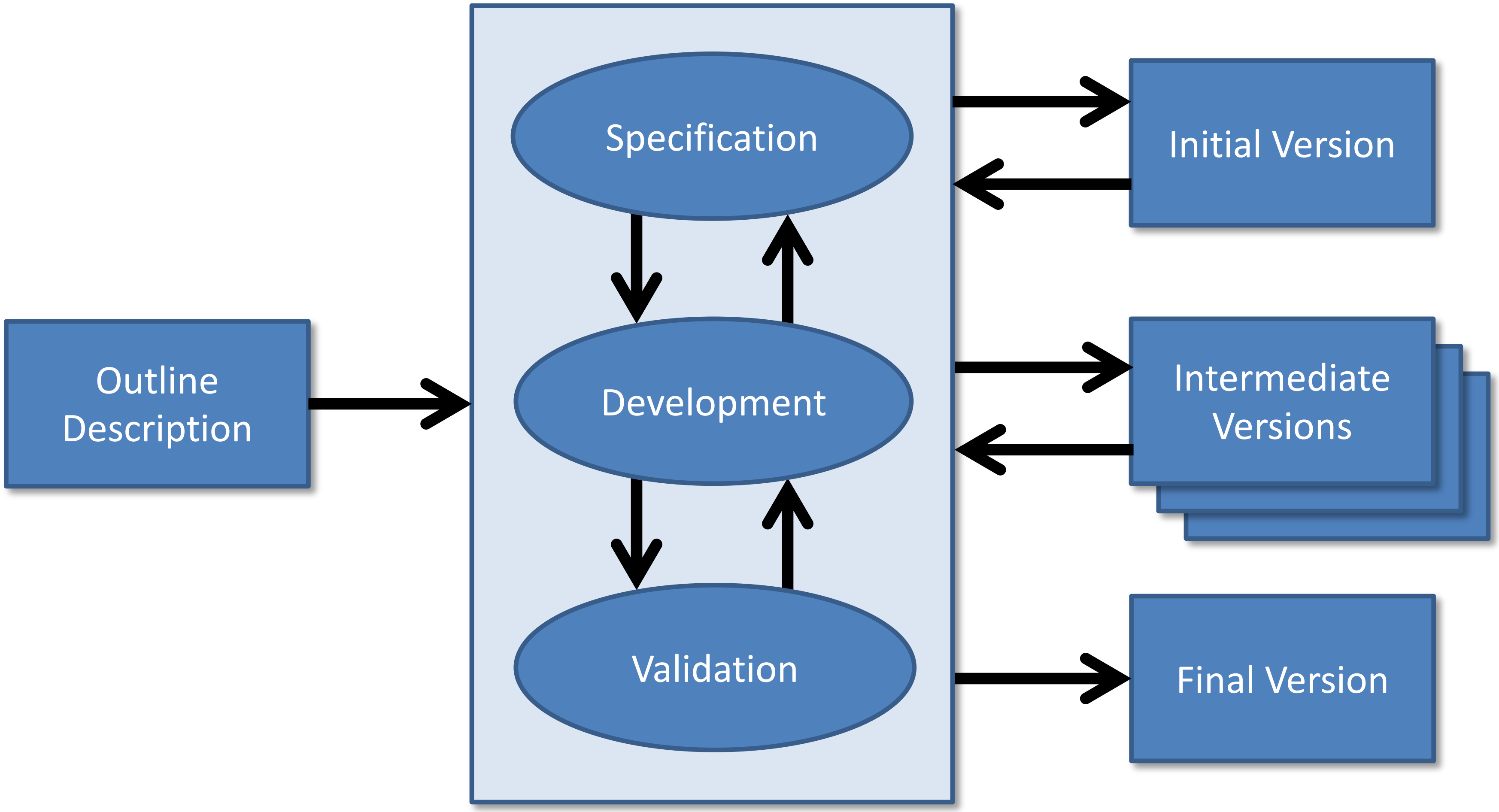
Drawbacks of Prototyping Model

- Inadequate or Incomplete Prototypes
  - Full prototypes of complex systems can be difficult or impossible to create quickly
  - Thus prototypes are often done in parts, which may miss critical requirements at the integration or complete system stage
- When to Stop Iterating
  - Easy to have users convince you to continue refining beyond the point where requirements and design are sufficient ("creeping excellence")

Evolutionary Development

- Prototype Evolution
  - Evolutionary prototyping is a method to avoid wasting work and take advantage of "creeping excellence" by smoothly evolving the initial prototype to the final product
  - In essence, never leave prototype iteration until implementation is complete

Evolutionary Development



Summary

- Software Process, Part I
  - Software development has four tasks
  - Software development processes differ in how these are interlaced
  - Oldest and most common process is the Waterfall Process
  - Some recent and popular processes are based on Prototyping

Summary

- Today's References
  - Kan, Metrics and Models in Software Quality Engineering
    - Ch. 2, Software Development Process Models
  - Sommerville, Software Engineering
    - Ch. 2, Software Processes
- Next time
  - More software process models