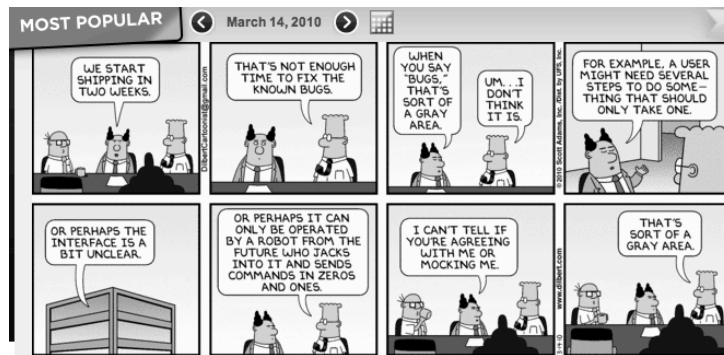


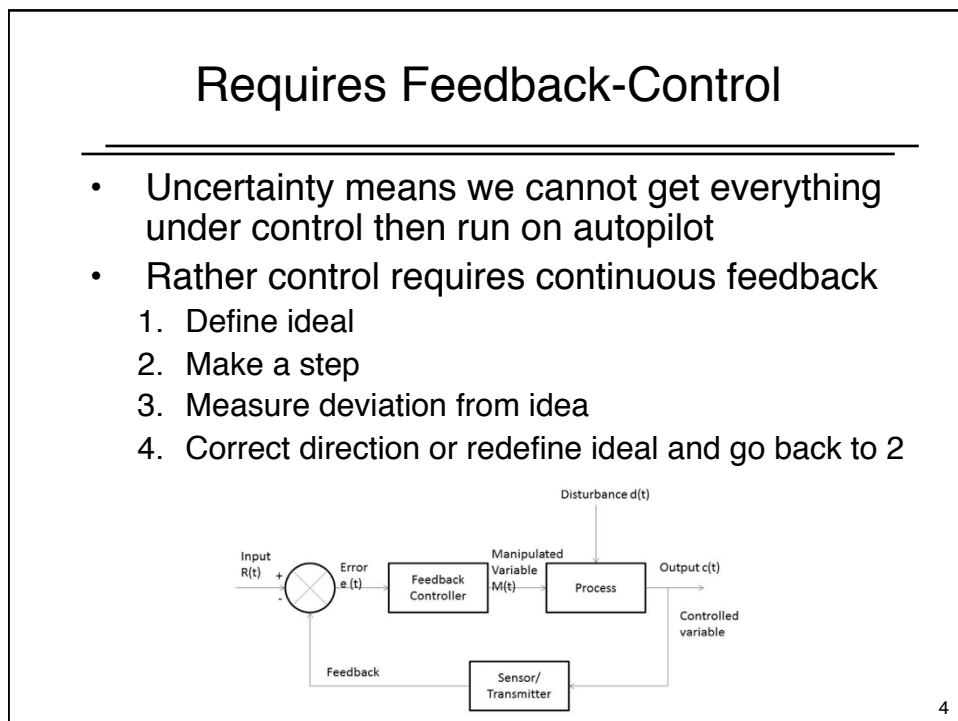
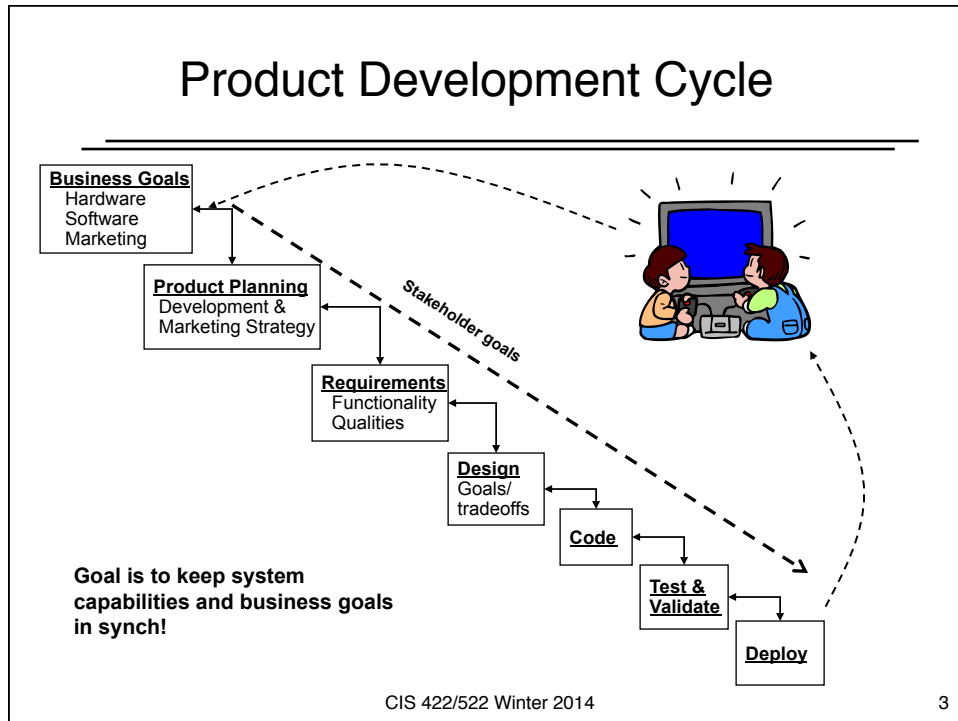
Quality Assurance

Need for a plan
QA Basics
Active reviews



View of SE in this Course

- The *purpose of Software Engineering* is to *gain and maintain* intellectual and managerial control over the products and processes of software development.
 - **Intellectual control:** able to make rational development decisions based on an understanding of the downstream effects of those choices.
 - **Managerial control** means we likewise control development *resources* (budget, schedule, personnel).

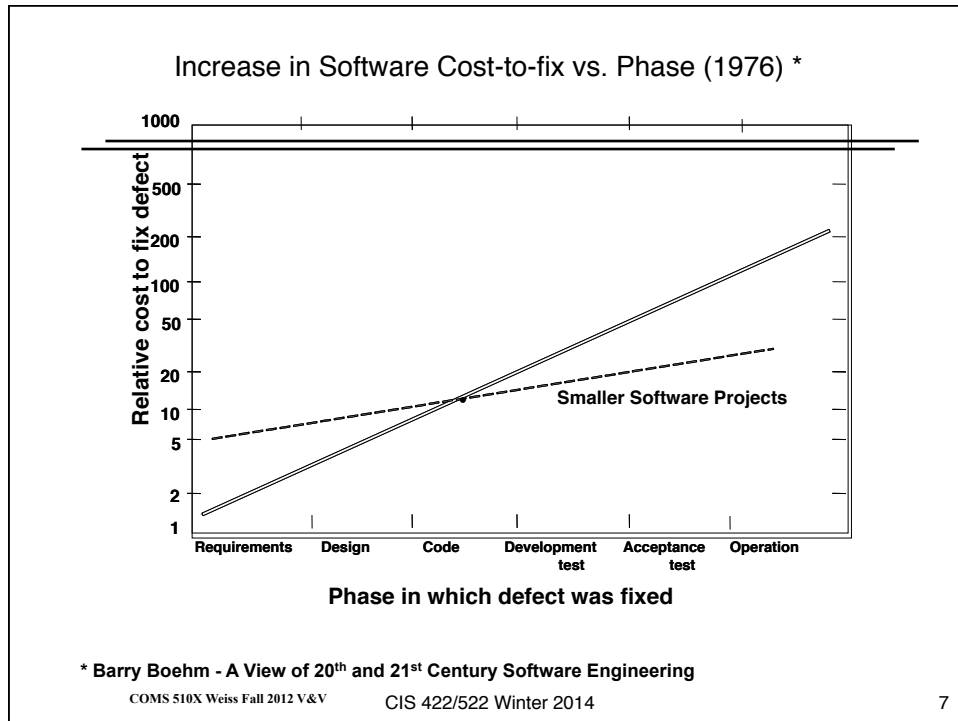


Example: System Requirements

- What happens if we get requirements wrong?
- How do we avoid getting them wrong?
 - What are different ways they can be wrong?
 - How do we check for correctness?
 - How can we maintain correctness over time?
- What is the right time for these activities?
- Who should do the work?

QA Questions

- Do the requirements capture what the stakeholders want?
 - Are they correct?
 - Are they complete relative to stakeholder needs?
 - Do they define functional and quality requirements?
- Are they internally complete and consistent?
- What if they change?
- Is the code consistent with the requirements?
- How do we check for these properties?



Quality is Cumulative

Requirements Analysis	<ul style="list-style-type: none"> • Are the requirements valid? • Complete? Consistent? Implementable? • Testable?
Architectural Design	<ul style="list-style-type: none"> • Does the design satisfy requirements? • Are all functional capabilities included? • Are qualities addressed (performance, maintainability, usability, etc.?)
Detailed Design	<ul style="list-style-type: none"> • Do the modules work together to implement all the functionality? • Are likely changes encapsulated? • Is every module well defined
Coding	<ul style="list-style-type: none"> • Implement the required functionality? • Race conditions? Memory leaks? Buffer overflow?

CIS 422/522 Winter 2014 8

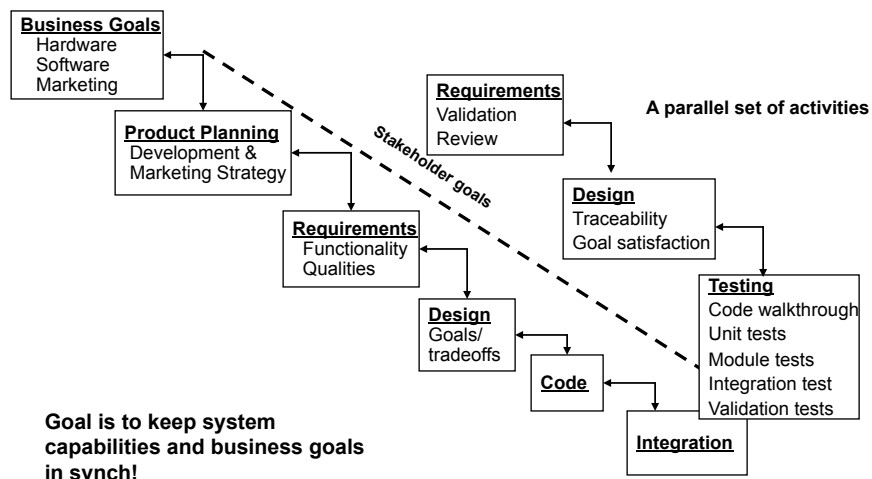
We need a plan!

- QA activities are
 - Critical to control (and project success)
 - Part of every phase of the project
 - Time consuming, labor intensive and expensive
 - Potentially unbounded use of resources
 - Consumes significant project resources
 - Cannot do everything, need to choose
- Suggests need to plan QA activities
 - Detect issues as early as possible
 - Target highest priority/risk issues for project
 - Support cost-effective use of resources

CIS 422/522 Winter 2014

9

Product Development Cycle



CIS 422/522 Winter 2014

10

QA Plan

- Purpose: synchronize QA activities with project deliverables such that
 - Artifacts satisfy quality goals
 - Delivered code is consistent with stakeholder needs
- The plan make clear how the project will meet overall quality goals
 - The overall QA objectives, strategy, and methodologies
 - The kinds of QA activities that should occur
 - Roles that will carry out the activities
 - When the activities should occur

QA Activities

Verification and Validation

Validation and Verification

- *Validation*: activities to answer the question – “Are we building a system the customer wants?”
 - Familiar activity: customer review of prototype
- *Verification*: activities to answer the question – “Are we building the system consistent with its specifications?”
 - Most familiar verification activity is functional testing
- Both are processes, both have many variations

V&V Methods

- Most applied V&V uses one of two methods
- **Review**: use of human skills to find defects
 - Pro: applies human understanding, skills. Good for detecting logical errors, problem misunderstanding
 - Con: poor at detecting inconsistent assumptions, details of consistency, completeness. Labor intensive
- **Testing**: use of machine execution
 - Pro: can be automated, repeated. Good at detecting detail errors, checking assumptions
 - Con: cannot establish correctness or quality
- Tend to reinforce each other

Reviews (1)

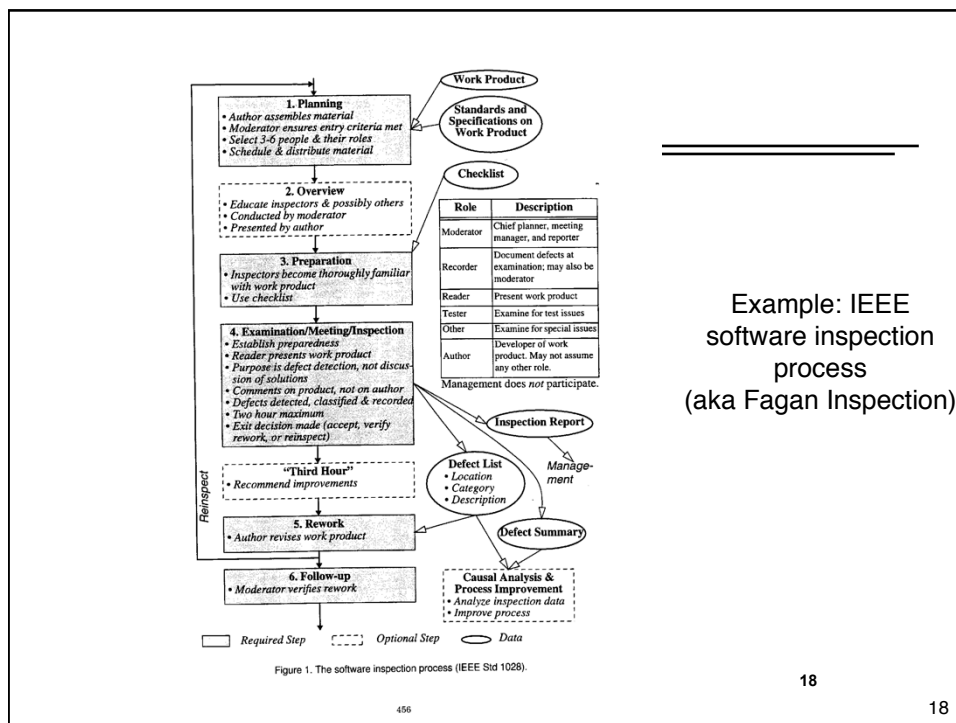
- Informal
 - No explicit process or recording of results
 - “Please read this for me”
 - “This”: requirements, architecture or design document, code, test plan, etc.
 - Could be several readers, selected by author
 - Author takes comments and makes revisions as he/she sees fit.

Reviews (2)

- Formal
 - Includes people outside the team
 - Explicit process, results recorded and tracked
 - Examples:
 - Desk check with questions to be answered by a specific date
 - Issues raised by answers recorded
 - Author revises artifact after receiving all answers
 - Revised artifact recirculated among reviewers for consensus
 - Formal Meeting held at a pre-defined time and place (maybe online or by conference call)
 - Reviewers read artifact in advance
 - Comments may or may not be recorded in advance of meeting
 - Facilitator leads discussion of artifact, often on line-by-line basis
 - Issues raised by discussion recorded
 - Author revises artifact after the meeting in response to issues
 - Revised artifact recirculated among reviewers for consensus

Peer Review Process

- Peer Review: a process by which a *software product is examined by peers of the product's authors with the goal of finding defects*
- Why do we do peer reviews?
 - Review is often the only available verification method before code exists
 - Formal peer reviews (inspections) instill some discipline in the review process
- Particularly important for distributed teams
 - Supports communication and visibility
 - Provides feedback on both *quality and understanding*
 - i.e., makes the communication effectiveness and level of understanding visible
 - A good review shows communication is working!



Active Reviews

Effectiveness of Peer Reviews

- Generally considered *most effective manual technique for detecting defects*
 - Analysis of 12,000 development projects showed defect detection rate of 60-65% for formal inspection 30% for testing
 - Bell-Northern found 1 hour code inspecting saves 2 to 4 hours code testing
 - Effect is magnified in earlier inspections (e.g., 30 times for requirements in one study)
- Means that you should be doing peer reviews, but...
 - Doesn't mean that manual inspections cannot be improved
 - Doesn't mean that manual inspections are the best way to check for every properties (e.g., completeness)
 - Should be one component of the overall V&V process

Peer Review Problems

- Tendency for reviews to be incomplete and shallow
- Reviewers typically swamped with information, much of it irrelevant to the review purpose
- Reviewers lack clear individual responsibility
- Effectiveness depends on reviewers to initiate actions
 - Review process requires reviewers to speak out
 - Keeping quiet gives lowest personal risk
 - Rewards of finding errors are unclear at best

Peer Review Problems (2)

- Large meeting size hampers effectiveness, increases cost
 - Makes detailed discussion difficult
 - Few present reviewers have interest/expertise on any one issue
 - Wastes everyone else's time and energy
- No way to cross-check unstated assumptions

Qualities of Effective Review

- Ensures adequate coverage of artifact in breadth and depth
- Reviewers review only issues on which they have expertise
- Review process is active: i.e., performing the review produces visible output (risk in in doing nothing)
- Individual responsibilities are clear and fulfilling them is evidence of a job well done.

Qualities of Effective Review (2)

- Review process focuses on finding specific kinds of errors.
- Limit meetings to focused groups and purposes requiring common understanding or synergy
 - Permit detailed discussion of issues
 - Expose where assumptions differ

Active Reviews

Goal: Make the reviewer(s) think hard about what they are reviewing

- 1) Identify several types of review each targeting a different type of error (e.g., UI behavior, consistency between safety assertions and functions).
- 2) Identify appropriate classes of reviewers for each type of review (specialists, potential users, methodology experts)
- 3) Assign reviews to achieve coverage: each applicable type of review is applied to each part of the specification

Active Reviews (2)

- 4) Design review questionnaires (key difference)
 - Define questions that the review must answer by using the specification
 - Target questions to bring out key issues
 - Phrase questions to require “active” answers (not just “yes”)
- 5) Review consists of filling out questionnaires defining
 - Section to be reviewed
 - Properties the review should check
 - Questions the reviewer must answer
- 6) Review process: overview, review, meet
 - One-on-one or small, similar group
 - Focus on discussion of issues identified in review
 - Purpose of discussion is understanding of the issue (not necessarily agreement)

Examples

- In practice: an active review asks a qualified reviewer to check a specific part of a work product for specific kinds of defects by answering specific questions, e.g.,
 - Ask a designer to check the functional completeness by showing the calls sequences sufficient to implement a set of use cases
 - Ask a systems analyst to check the ability to create required subsets by showing which modules would use which
 - As a developer to check the data validity of a module's specification by showing what the output would be for in-range and out-of-range values
 - Ask a technical writer to check the SRS for grammatical errors
- Can be applied to any kind of artifact from requirements to code

Conventional vs. Active Questions

- **Goal: Make the reviewer(s) think hard about what they are reviewing***
 - Define questions that the review must answer by using the specification
 - Target questions to bring out key issues
 - Phrase questions to require “active” answers (not just “yes”)

Conventional Design Review Questions	Active Better Design Review Questions*
Are exceptions defined for every program?	For each access program in the module, what exceptions that can occur?
Are the right exceptions defined for every program?	What is the the range or set of legal values?
Are the data types defined?	For each data type, what are • an expression for a literal value of that data type; • a declaration statement to declare a variable for that type; • the greatest and least values in the range of that data type?
Are the programs sufficient?	Write a short pseudo-code program that uses the design to accomplish {some defined task}.

Role of Use Cases

- Use cases or scenarios can be effectively used in active review
- Apply requirements scenarios to verify design against requirements
 - “Show the sequence of program calls that would implement use case C”
 - “Which modules would have to change to add feature F (a likely change)?”
- Conversely, can check properties ask the reviewer to construct scenarios
 - “What sequence of calls would result in an exception E?”

Why Active Reviews Work

- Focuses reviewer’s skills and energies where they have skills and where those skills are needed
 - Questionnaire allows reviewers to concentrate on one concern at a time
 - No one wastes time on parts of the document where there is little possibility of return.
- Largest part of review process (filling out questionnaires) is conducted independently and in parallel
- Reviewers must participate actively but need not risk speaking out in large meetings
- Downside: much more work for V&V (but can be productively pursued in parallel with document creation)

Summary

- Need to do reviews to find defects
- Active reviews are more efficient and effective but may take more effort