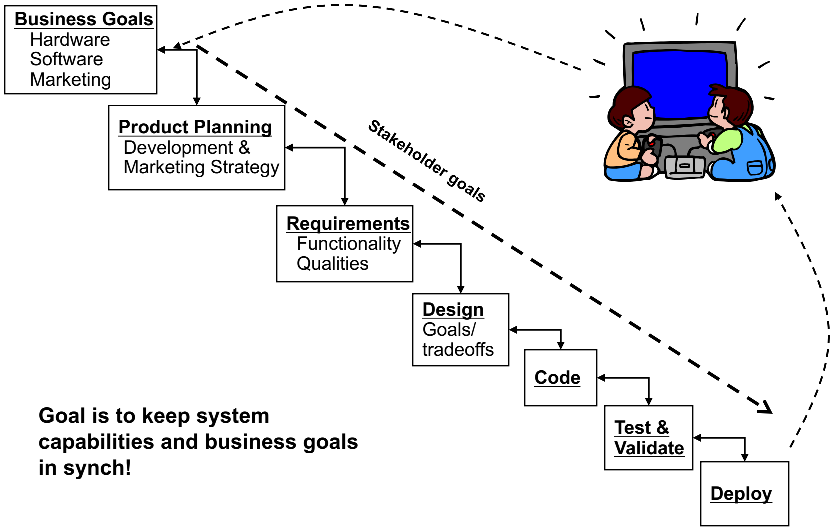
CIS 422/522
Quality Assurance II
Software Reviews

Role of reviews in QA

Types of reviews

Active review method

Product Development Cycle

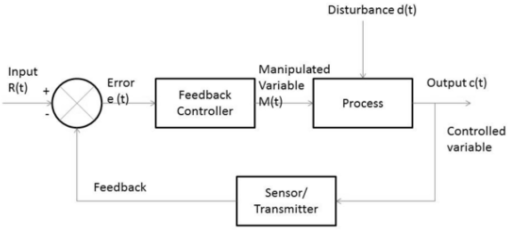


Goal is to keep system capabilities and business goals in synch!

Requires Feedback-Control

- Uncertainty means we cannot get everything under control then run on autopilot
- Rather control requires continuous feedback
  1. Define ideal
  2. Make a step
  3. Measure deviation from ideal
  4. Correct direction or redefine ideal and go back to 2

Role of QA



Quality is Cumulative

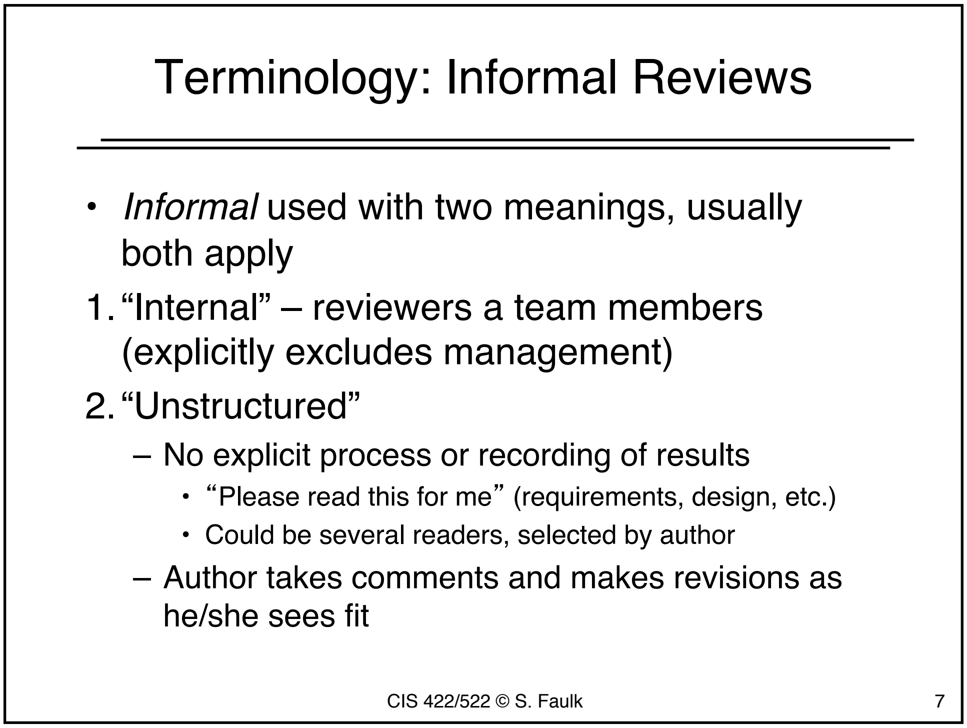
Requirements Analysis	• Are the requirements valid? • Complete? Consistent? Implementable? • Testable?
Architectural Design	• Does the design satisfy requirements? • Are all functional capabilities included? • Are qualities addressed (performance, maintainability, usability, etc.?
Detailed Design	• Do the modules work together to implement all the functionality? • Are likely changes encapsulated? • Is every module well defined
Coding	• Implement the required functionality? • Race conditions? Memory leaks? Buffer overflow?

Human vs. Machine

- Two practical choices: testing by machine or review by a person
- A combination of manual and automated techniques is most cost effective
  - People are better at detecting many kinds of errors than machines
  - Machines are better at repetitive checks and minute details (comparing values)
- Testing works best in a supporting role (checking assumptions)
  - Not applicable unless machine readable
  - Poor at detecting thought errors

Peer Review

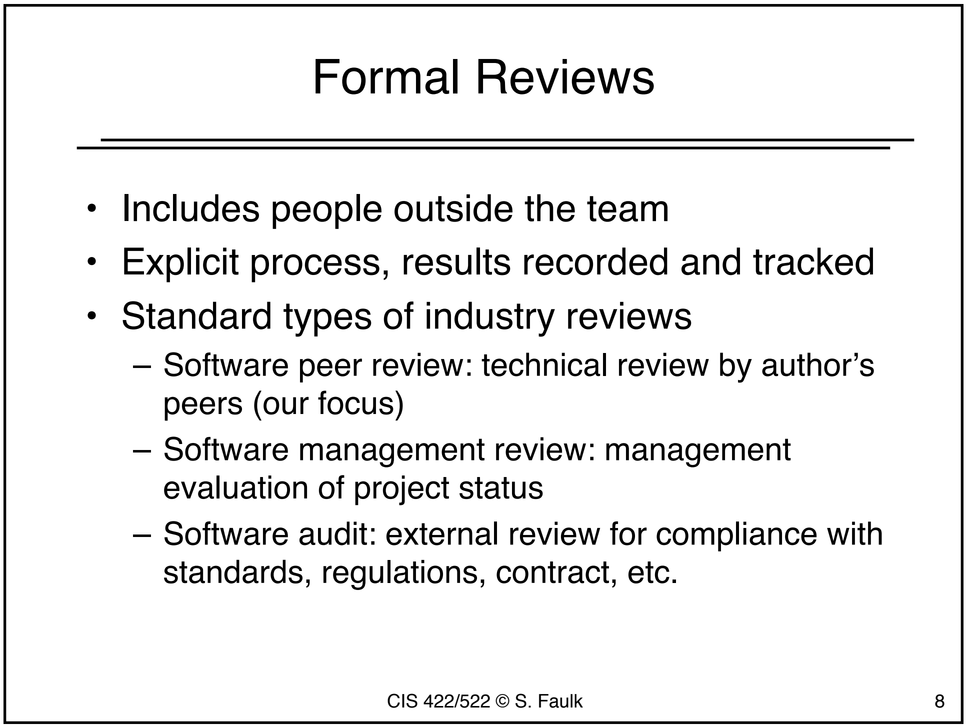
- Primary defect detection method where automation is not possible or practical
  - E.g. review for meaning, intent, goal satisfaction, human factors, etc.
  - Especially review of upstream artifacts (e.g. requirements, design)
- Very effective if done carefully, systematically
  - Analysis of 12,000 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)



Terminology: Informal Reviews

- *Informal* used with two meanings, usually both apply

1. "Internal" – reviewers a team members (explicitly excludes management)
2. "Unstructured"
   - No explicit process or recording of results
     - "Please read this for me" (requirements, design, etc.)
     - Could be several readers, selected by author
   - Author takes comments and makes revisions as he/she sees fit

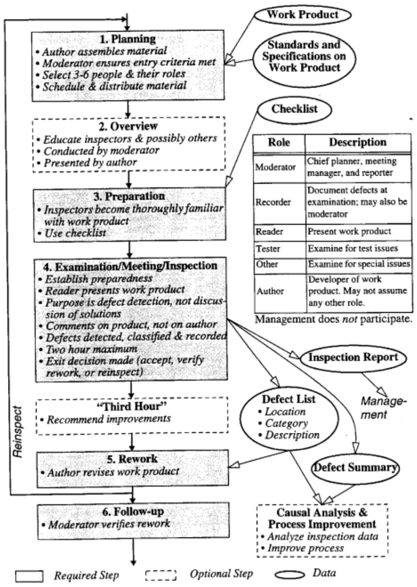


Formal Reviews

- Includes people outside the team
- Explicit process, results recorded and tracked
- Standard types of industry reviews
  - Software peer review: technical review by author's peers (our focus)
  - Software management review: management evaluation of project status
  - Software audit: external review for compliance with standards, regulations, contract, etc.

Example: Software Peer Review

- Definition: a form of *technical review in* which a software product is examined by peers of the product's authors with the goal of finding defects
  - Also called "software inspection"
  - Most common type of technical review in industry
  - Often standardized part of milestone planning
- Formal Meeting held at a pre-defined time and place
  - Reviewers read artifact in advance
  - 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

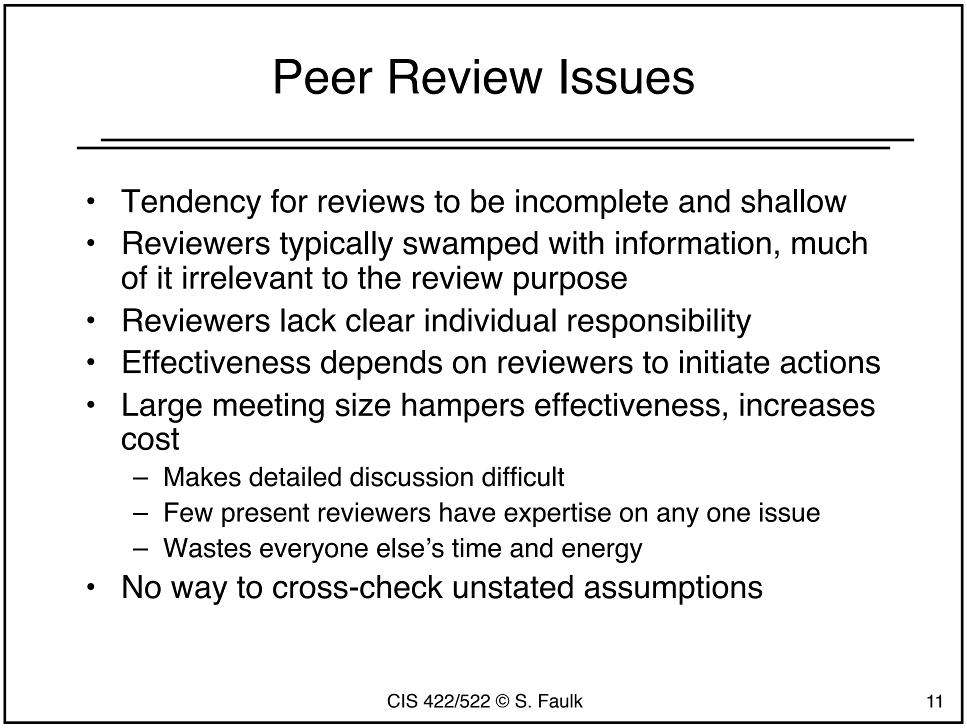

Work Product
Standards and Specifications on Work Product
1. Planning
• Author assembles material
• Moderator ensures entry criteria met
• Select 3-6 people & their roles
• Schedule & distribute material
Checklist
2. Overview
• Educate inspectors & possibly others
• Conducted by moderator
• Presented by author
3. Preparation
• Inspectors become thoroughly familiar with work product
• Use checklist
4. Examination/Meeting/Inspection
• Establish preparedness
• Reader presents work product
• Purpose is defect detection, not discussion of solutions
• Comments on product, not on author
• Defects detected, classified & recorded
• Two hour maximum
• Exit decision made (accept, verify rework, or reinspect)
Inspection Report
Management
Defect List
• Location
• Category
• Description
"Third Hour"
• Recommend improvements
Reinspect
5. Rework
• Author revises work product
Defect Summary
6. Follow-up
• Moderator verifies rework
Causal Analysis & Process Improvement
• Analyze inspection data
• Improve process
Required Step
Optional Step
Data

Role	Description
Moderator	Chief planner, meeting manager, and reporter
Recorder	Document defects at examination; may also be moderator
Reader	Present work product
Tester	Examine for test issues
Other	Examine for special issues
Author	Developer of work product. May not assume any other role.

Management does *not* participate.

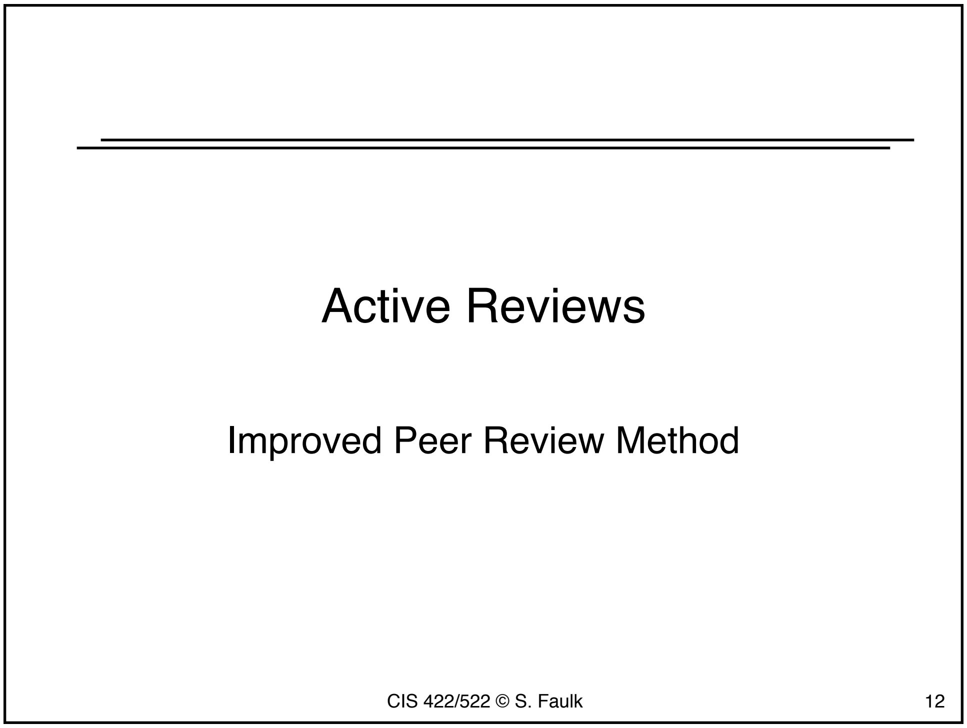
Figure 1. The software inspection process (IEEE Std 1028).

Example: IEEE software inspection process (aka Fagan Inspection)



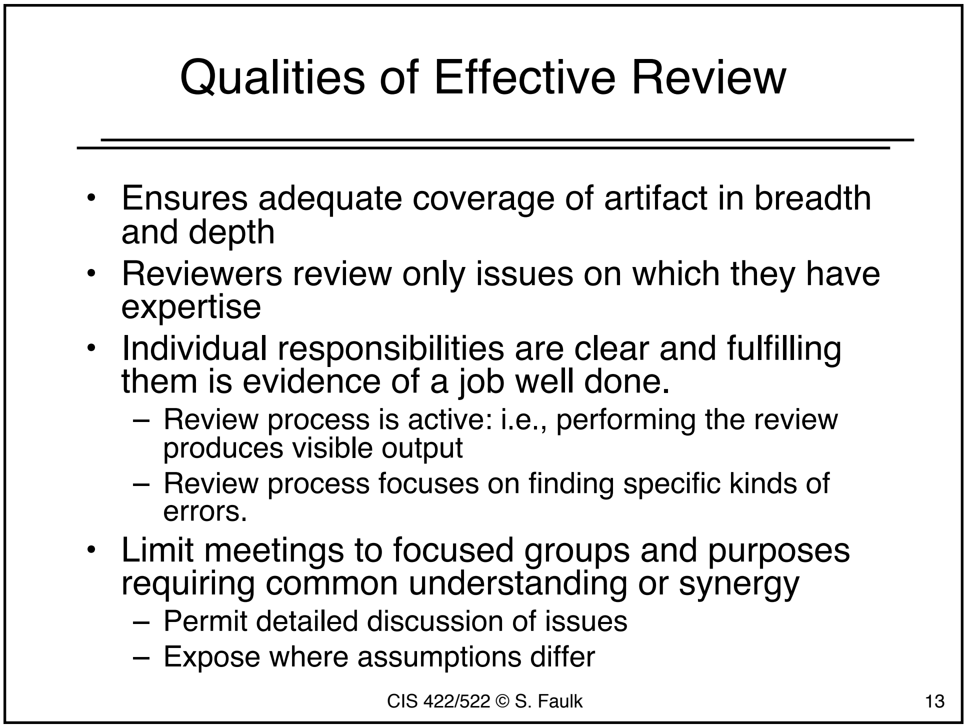
Peer Review Issues

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



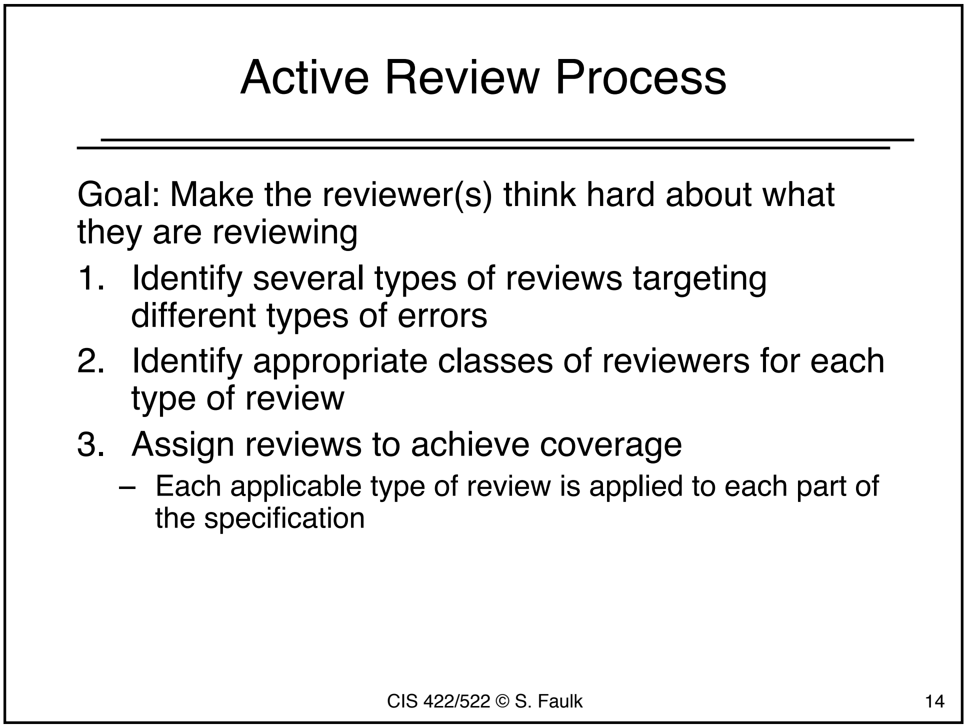
Active Reviews

Improved Peer Review Method



Qualities of Effective Review

- Ensures adequate coverage of artifact in breadth and depth
- Reviewers review only issues on which they have expertise
- Individual responsibilities are clear and fulfilling them is evidence of a job well done.
  - Review process is active: i.e., performing the review produces visible output
  - 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 Review Process

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

1. Identify several types of reviews targeting different types of errors
2. Identify appropriate classes of reviewers for each type of review
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 *using the artifact* to fill out questionnaire
6. Review process: overview, review, meet
   - One-on-one or small, group
   - Discuss issues identified in review
   - Track and respond to issues

Examples

- 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
  - For each access program in the module, what exceptions that can occur?
  - 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”)

ConventionalGood Design Review Questions	ActiveBetter 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}.

Applying 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 actions 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
- Largest part of review process (filling out questionnaires) is conducted independently and in parallel
- Reviewers actively use the artifact
- Cost: more work for QA team but…
  - Can be started early and in parallel
  - Can be reused for many artifacts

For Your Projects

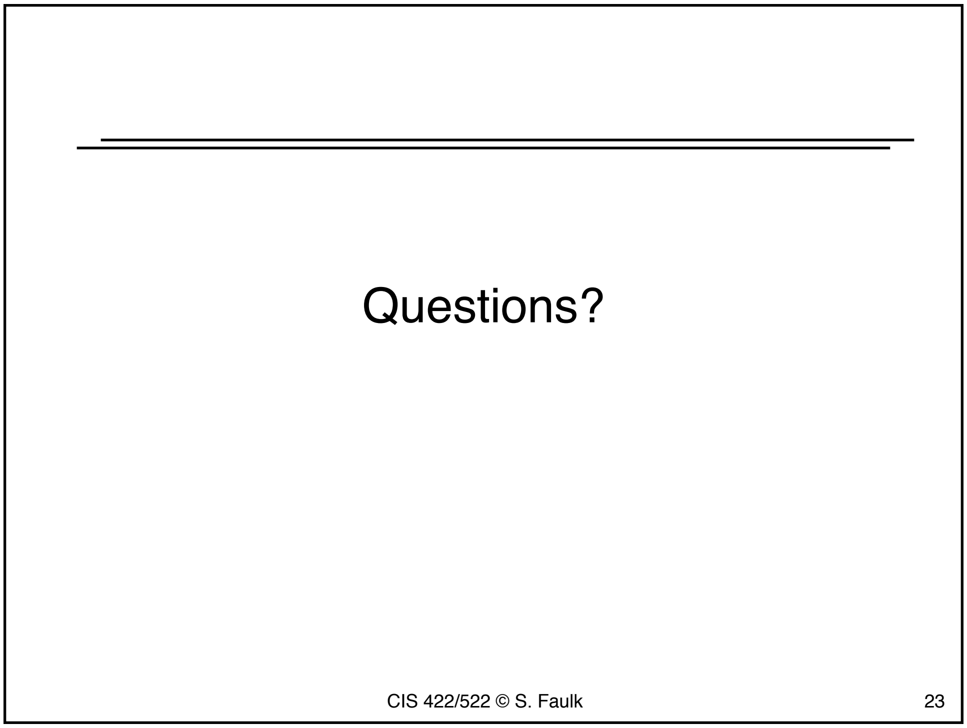
- Create a plan
- For non-code artifacts
  - Identify which artifacts will be reviewed, by whom, and when
  - Define the goal of each review
  - Specify the review method
  - Record the results
- Types of reviews
  - Good: Standard review
  - Better: Review with checklist
  - Best: Active review

Example: Requirements

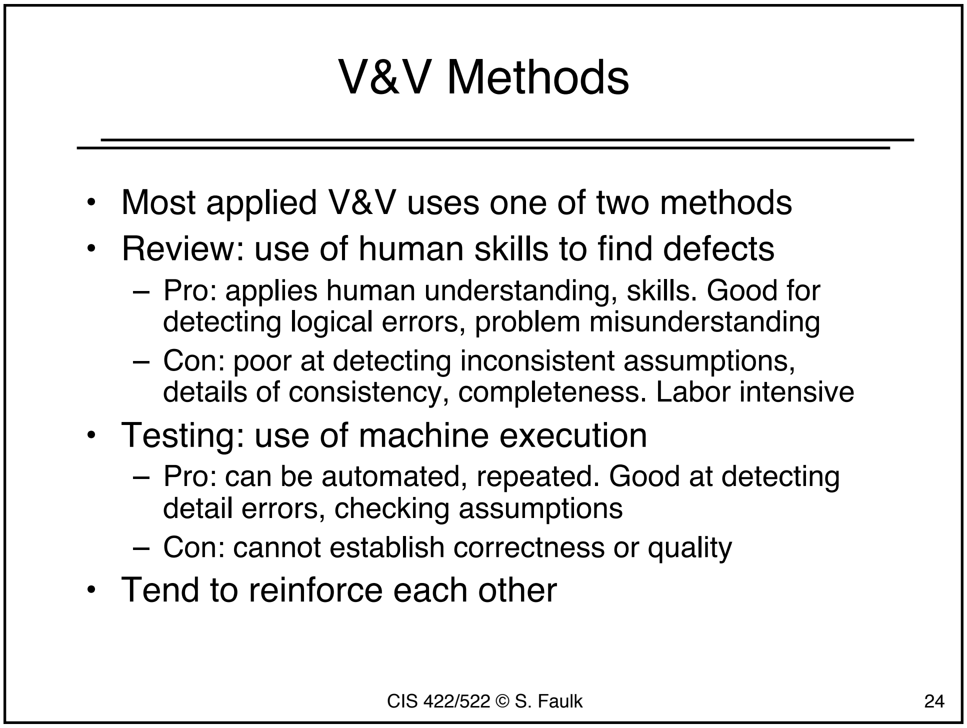
- ConOps review
  - When should we review it?
  - Who should review it?
  - What qualities or properties do we review for?
  - Which review method should we use?
  - What should we do with the results?

Summary

- Reviews are usually only practical method for
  - Early artifacts (requirements, etc.)
  - Defects in understanding, some qualities, etc
- Effective method of defect detection
- Active reviews are more effective than standard inspections



Questions?



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