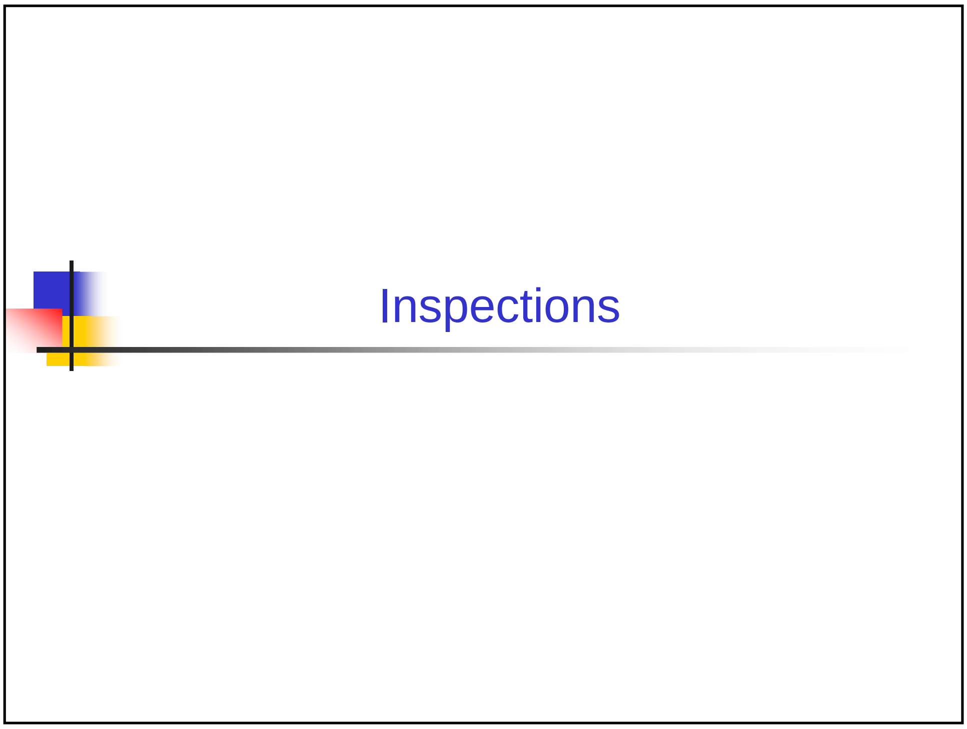
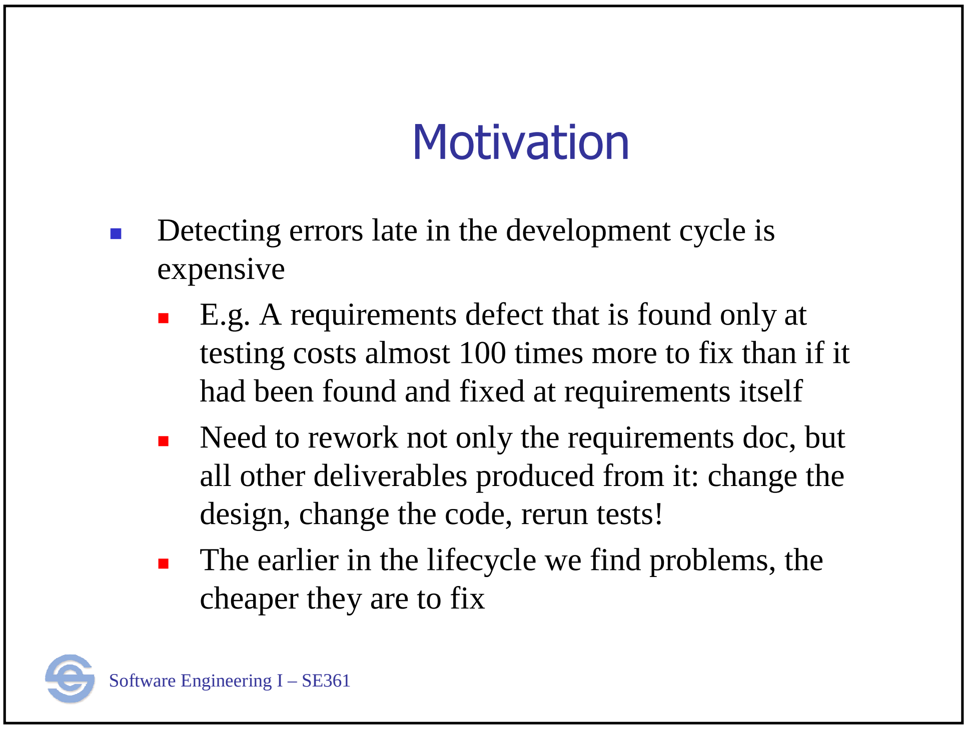
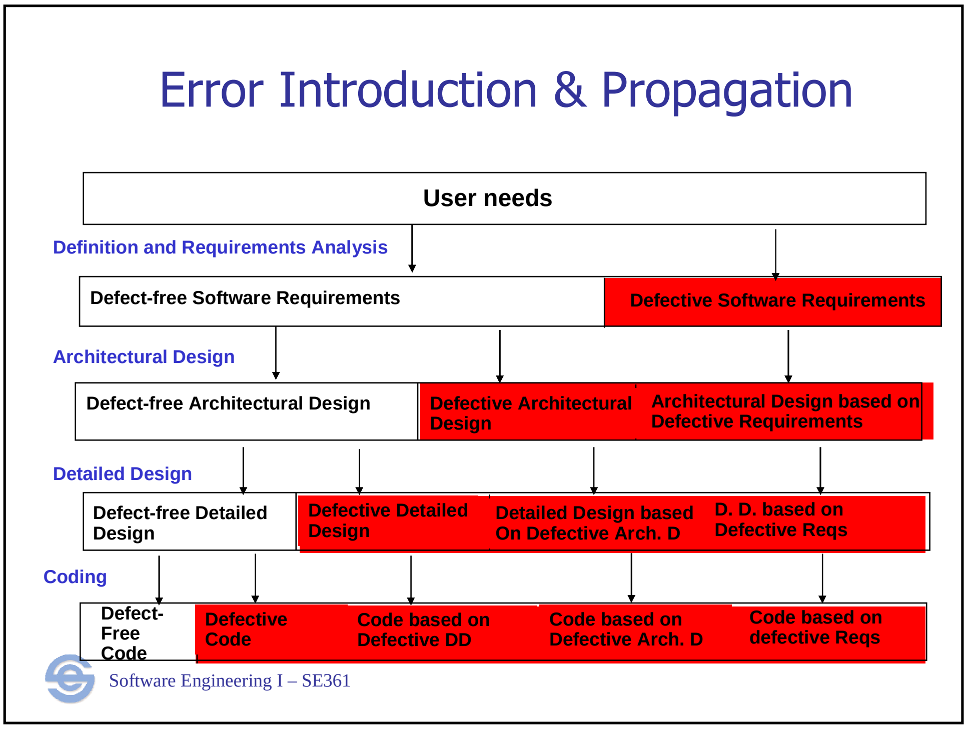


Inspections



Motivation

- Detecting errors late in the development cycle is expensive
  - E.g. A requirements defect that is found only at testing costs almost 100 times more to fix than if it had been found and fixed at requirements itself
  - Need to rework not only the requirements doc, but all other deliverables produced from it: change the design, change the code, rerun tests!
  - The earlier in the lifecycle we find problems, the cheaper they are to fix



Error Introduction & Propagation

User needs

Definition and Requirements Analysis

Defect-free Software Requirements | Defective Software Requirements

Architectural Design

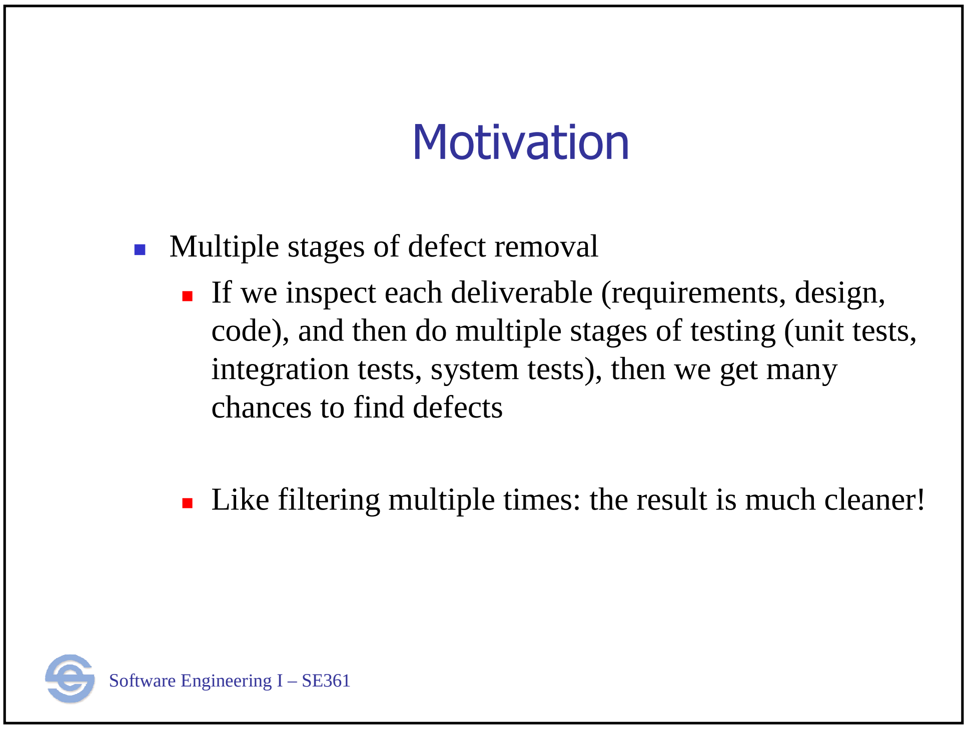
Defect-free Architectural Design | Defective Architectural Design | Architectural Design based on Defective Requirements

Detailed Design

Defect-free Detailed Design | Defective Detailed Design | Detailed Design based On Defective Arch. D | D. D. based on Defective Reqs

Coding

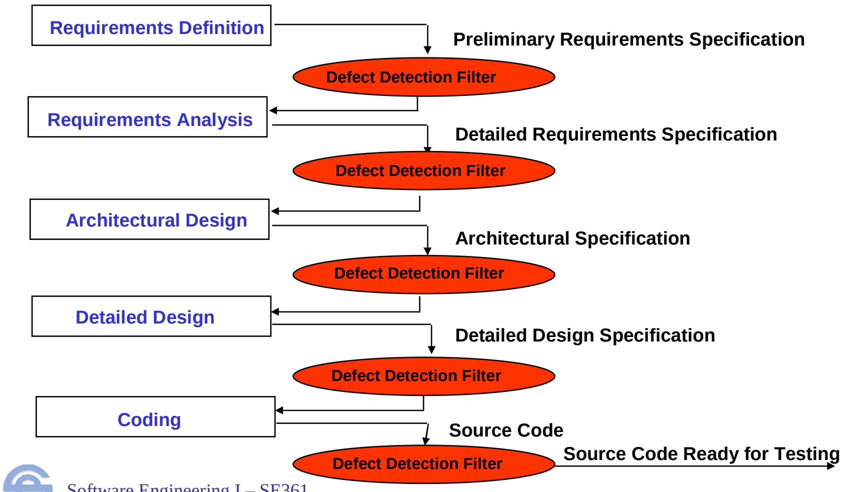
Defect-Free Code | Defective Code | Code based on Defective DD | Code based on Defective Arch. D | Code based on defective Reqs



Motivation

- Multiple stages of defect removal
  - If we inspect each deliverable (requirements, design, code), and then do multiple stages of testing (unit tests, integration tests, system tests), then we get many chances to find defects
  - Like filtering multiple times: the result is much cleaner!

Multiple Stages of Defect Removal



Requirements Definition

Preliminary Requirements Specification

Defect Detection Filter

Requirements Analysis

Detailed Requirements Specification

Defect Detection Filter

Architectural Design

Architectural Specification

Defect Detection Filter

Detailed Design

Detailed Design Specification

Defect Detection Filter

Coding

Source Code

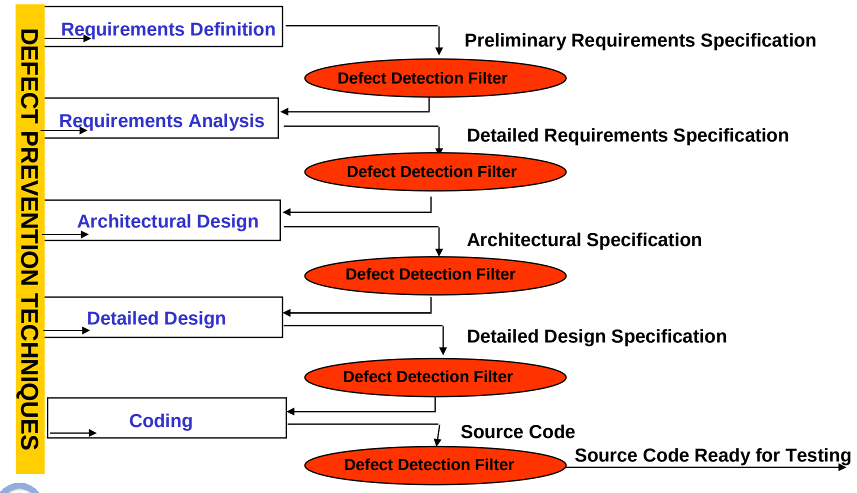
Defect Detection Filter

Source Code Ready for Testing

Defect Prevention

- In addition to removing defects through inspections, we can eliminate defects using
  - Checklists: common mistakes, concerns to address, activities to do
  - Templates: standard document formats that list the different aspects to be covered
    - Reduce work and avoid incompleteness
  - Tools and workflow automation
    - Avoid errors, inconsistencies and missing steps
    - Reduce effort too!

Quality-Centered Development



DEFECT PREVENTION TECHNIQUES

Requirements Definition

Preliminary Requirements Specification

Defect Detection Filter

Requirements Analysis

Detailed Requirements Specification

Defect Detection Filter

Architectural Design

Architectural Specification

Defect Detection Filter

Detailed Design

Detailed Design Specification

Defect Detection Filter

Coding

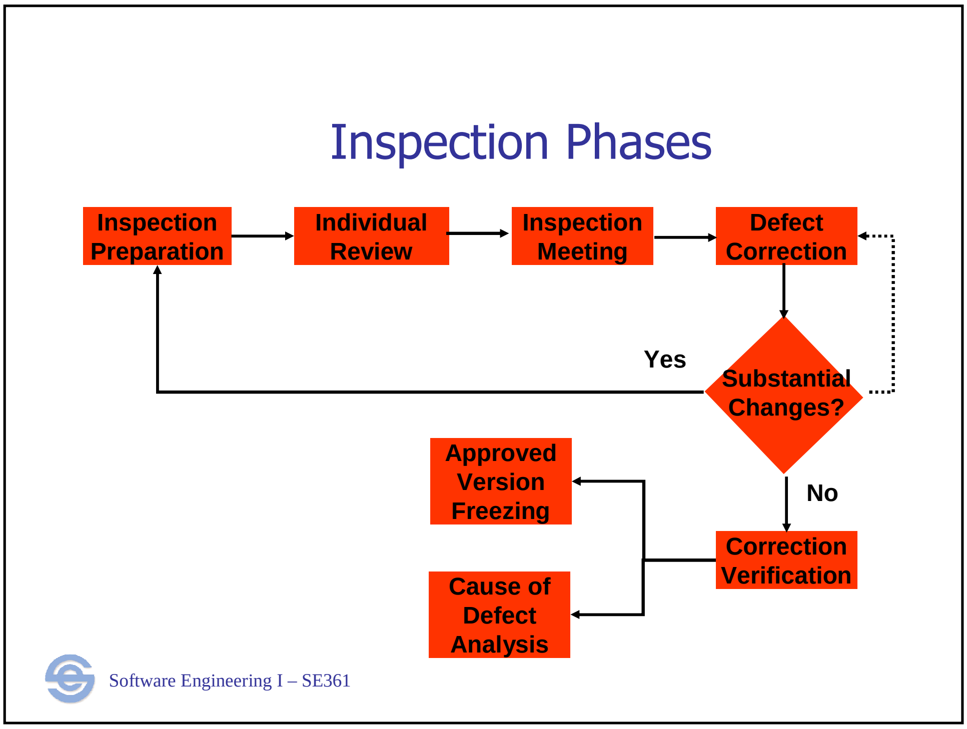
Source Code

Defect Detection Filter

Source Code Ready for Testing

Inspections

- A group of people review an artifact (code, documents) to find defects and identify opportunities for improvement
- Can be used for any document or code produced during the development
  - Preferably, all major development artifacts should be inspected
- Each reviewer spends several hours going through the artifact and finding problems and possible improvements
- Hold a review meeting to discuss the inputs from each reviewer and identify the problems that need fixing
- Fix problems, re-review if necessary



Inspection Phases

Inspection Preparation

Individual Review

Inspection Meeting

Defect Correction

Substantial Changes?

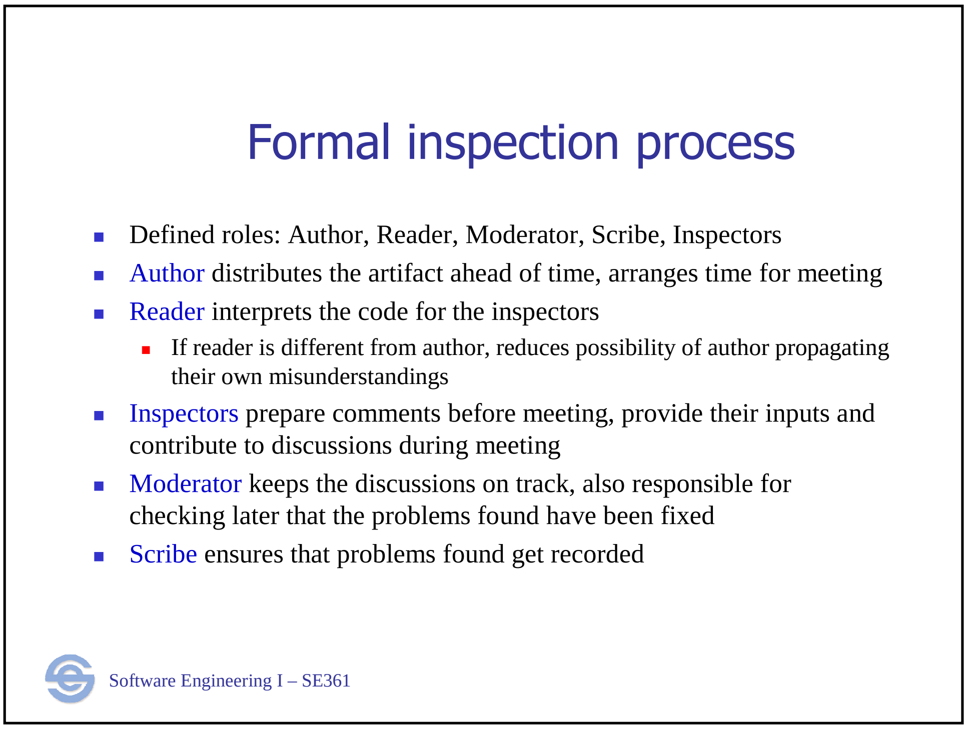
Yes

No

Correction Verification

Approved Version Freezing

Cause of Defect Analysis



Formal inspection process

- Defined roles: Author, Reader, Moderator, Scribe, Inspectors
- Author distributes the artifact ahead of time, arranges time for meeting
- Reader interprets the code for the inspectors
  - If reader is different from author, reduces possibility of author propagating their own misunderstandings
- Inspectors prepare comments before meeting, provide their inputs and contribute to discussions during meeting
- Moderator keeps the discussions on track, also responsible for checking later that the problems found have been fixed
- Scribe ensures that problems found get recorded

Inspection Meeting Preparation Process

- Author prepares the artifact to be inspected and makes it available to the moderator
  - Preferably at least 2 days prior to meeting
  - Give inspectors time to prepare
- Moderator obtains the inspection checklist and other support material
- Moderator distributes materials to all inspectors
- Inspectors inspect the artifact ***prior*** to coming to the meeting and make their own notes

Inspection Meeting Process

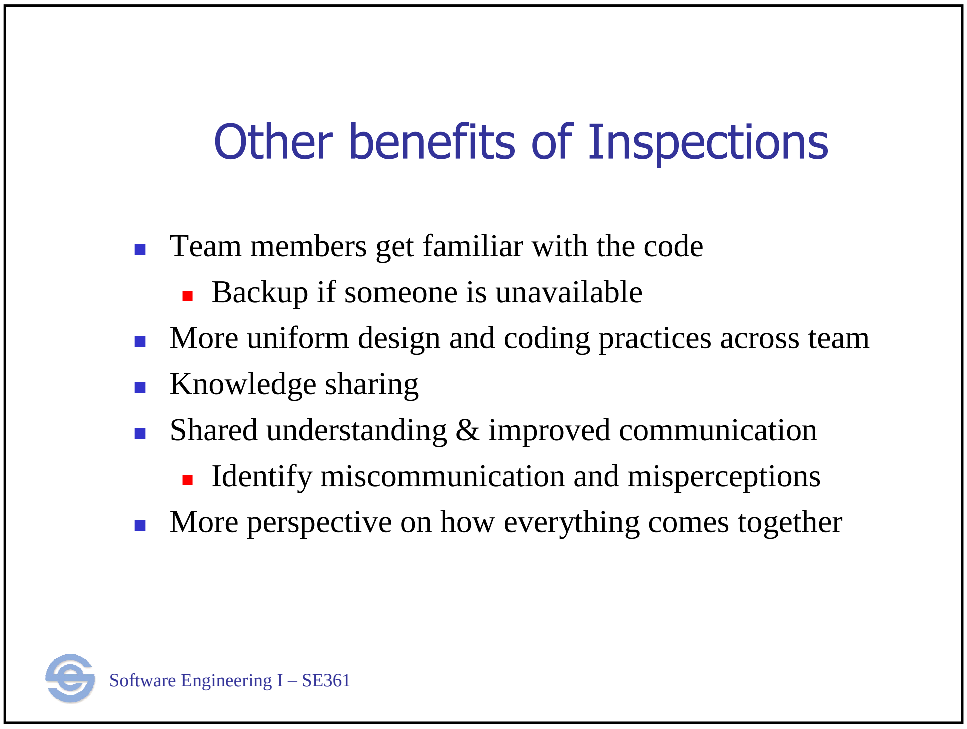
- The meeting is called to order by the moderator
- The reader interprets for the team as he/she understands what is in the artifact being inspected
  - In code inspections, the reader paraphrases the code as opposed to reading it line by line
- Based on their own reviews, inspectors question, add value, and contribute to the meeting
- The scribe records all issues raised on the appropriate inspection report form
- The author answers questions when necessary

Inspection Meeting Guidelines

- The objective of an inspection *is not* to correct defects but to set plausible course of action
- Author is there to clarify and to answer questions *not to* justify decisions he/she made
- Avoid personal attacks on the producer
- In code reviews *avoid* discussions of style, concentrate on important issues
- Inspection meetings must not last more than two hours

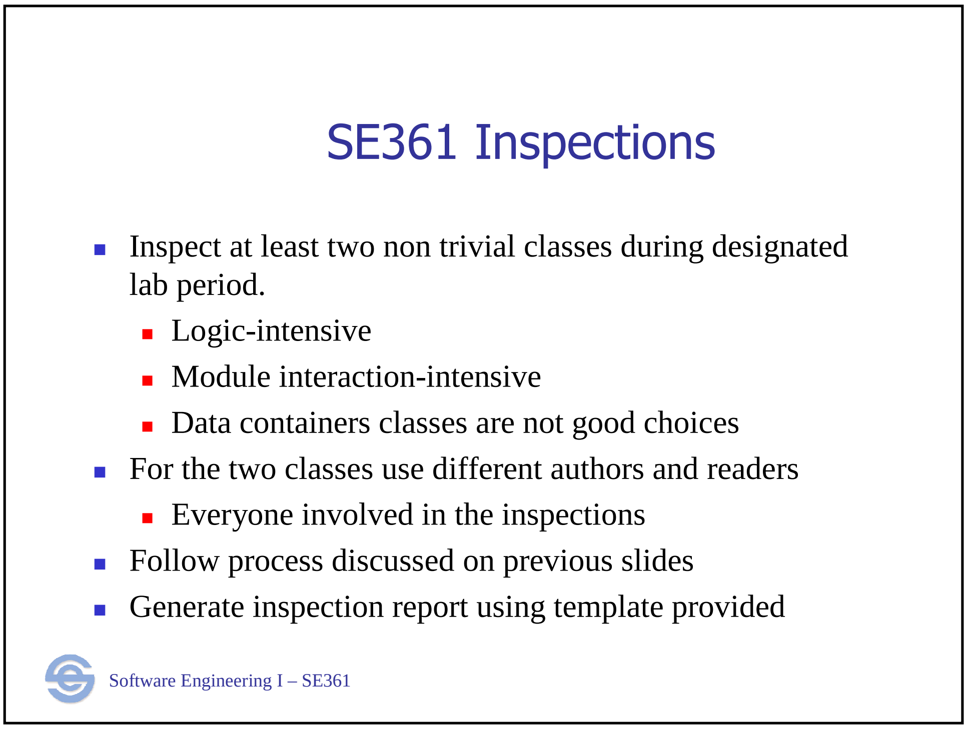
Tracking Process

- The scribe collects the defect report forms from all inspectors and uses them to finish up the inspection report form
- Author develops a response for each defect found during the meeting
- Author corrects all defects
- Author submits the corrected product to the moderator
- Moderator makes sure all defects were satisfactorily corrected and, if warranted, calls for a second inspection
- Moderator approves the final version so it can be frozen



Other benefits of Inspections

- Team members get familiar with the code
  - Backup if someone is unavailable
- More uniform design and coding practices across team
- Knowledge sharing
- Shared understanding & improved communication
  - Identify miscommunication and misperceptions
- More perspective on how everything comes together



SE361 Inspections

- Inspect at least two non trivial classes during designated lab period.
  - Logic-intensive
  - Module interaction-intensive
  - Data containers classes are not good choices
- For the two classes use different authors and readers
  - Everyone involved in the inspections
- Follow process discussed on previous slides
- Generate inspection report using template provided