

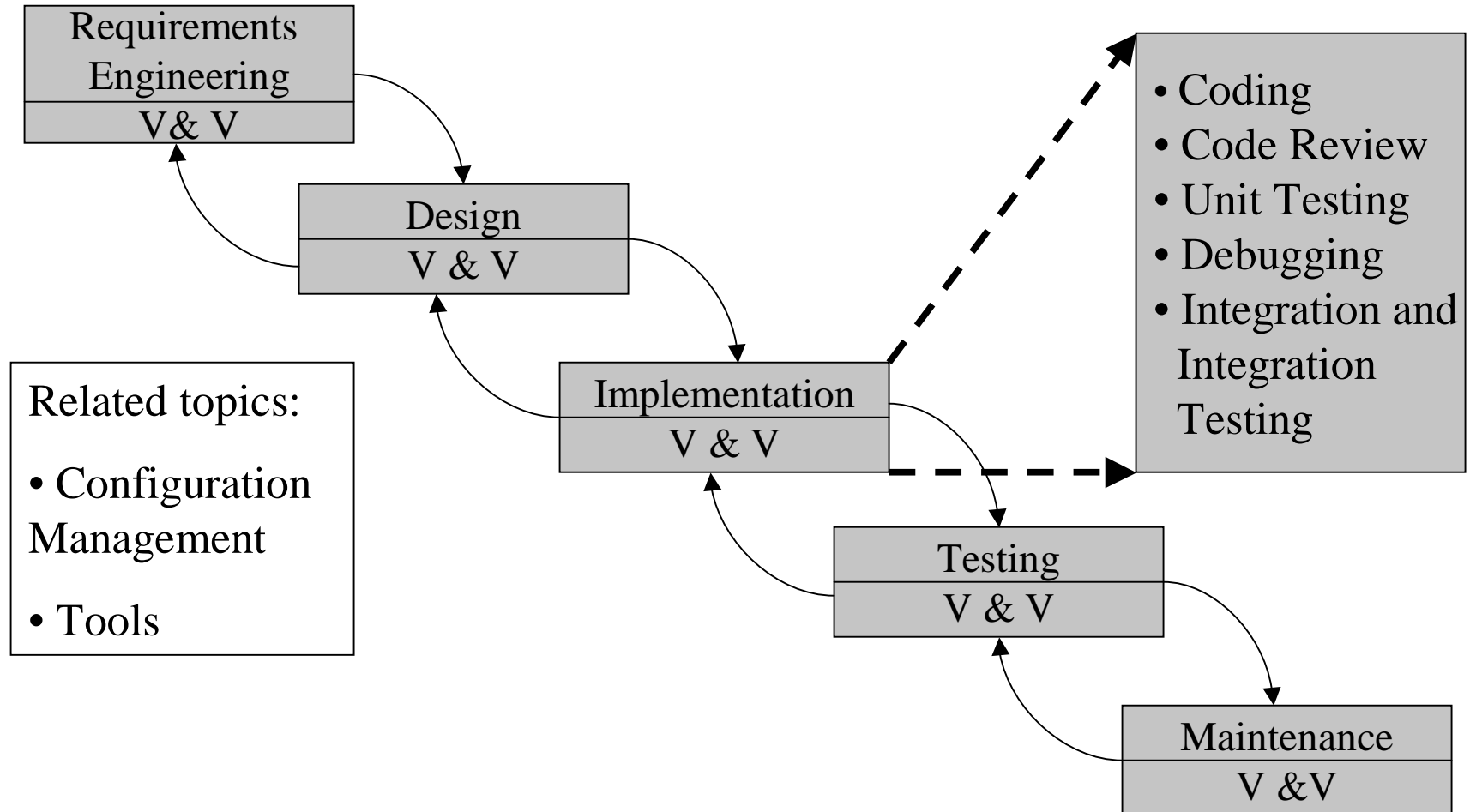
CSC444F: Software Engineering I

Mou Hu
mou.hu@utoronto.ca

Lecture 12: Implementation (I)

- Implementation Phase Activities
- Code Review, Inspection, Walkthrough
- Reading: Section 13.4

Implementation Phase Activities



Reviews, Inspections, Walkthroughs...

- Note: these terms are not widely agreed
 - formality
 - informal: *from* meetings over coffee, *to* regular team meetings
 - formal: scheduled meetings, prepared participants, defined agenda, specific format, documented output
 - “Walkthroughs”
 - developer technique (usually informal)
 - used by development teams to improve quality of product
 - focus is on finding defects
 - “(Fagan) Inspections”
 - a process management tool (always formal)
 - used to improve quality of the development process
 - collect defect data to analyze the quality of the process
 - written output is important
 - major role in training junior staff and transferring expertise

Source: Ref. [3]

Benefits of Formal Inspection

- For applications programming:
 - most reviewed programs run correctly first time
 - compare: 10-50 attempts for test/debug approach
- Data from large projects
 - Data from Bell-Northern Research:
 - Inspection cost: 1 hour per defect.
 - Testing cost: 2-4 hours per defect.
 - Post-release cost: 33 hours per defect.
 - error reduction by a factor of 5; (10 in some reported cases)
 - improvement in productivity: 14% to 25%
 - percentage of errors found by inspection: 58% to 82%
 - cost reduction of 50%-80% for V&V (even including cost of inspection)
- Effects on staff competence:
 - increased morale, reduced turnover
 - better estimation and scheduling (more knowledge about defect profiles)
 - better management recognition of staff ability

Source: Ref. [3]

Constraints

- Size
 - “enough people so that all the relevant expertise is available”
 - min: 3 (4 if author is present)
 - max: 7 (less if leader is inexperienced)
- Duration
 - never more than 2 hours
- Outputs
 - all reviewers must agree on the result: accept or re-work or re-inspect
 - all findings should be documented
 - summary report (for management)
 - detailed list of issues
- Scope
 - ↳ focus on small part of a design, not the whole thing
 - ↳ Fagan recommends rates:
 - 130-150 SLOC per hour
- Timing
 - ↳ Examines a product once its author has finished it
 - ↳ not too soon
 - product not ready - find problems the author is already aware of
 - ↳ not too late
 - product in use - errors are now very costly to fix

Source: Ref. [3]

Choosing Reviewers

- Possibilities
 - specialists in reviewing (e.g. QA people)
 - people from the same team as the author
 - people invited for specialist expertise
 - people with an interest in the product
 - visitors who have something to contribute
 - people from other parts of the organization
- Exclude
 - anyone responsible for reviewing the author
 - i.e. line manager, appraiser, etc.
 - anyone with known personality clashes with other reviewers
 - anyone who is not qualified to contribute
 - all management
 - anyone whose presence creates a conflict of interest

Source: Ref. [3]

Roles

Formal Walkthrough

- Review Leader
 - chairs the meeting
 - ensures preparation is done
 - keeps review focussed
 - reports the results
- Recorder
 - keeps track of issues raised
- Reader
 - summarizes the product piece by piece during the review
- Author
 - should actively participate (may be the reader)
- Other Reviewers
 - task is to find and report issues

Fagan Inspection

- Moderator
 - must be a competent programmer
 - should be specially trained
 - could be from another project
- Designer
 - programmer who produced the design being inspected
- Coder/Implementor
 - programmer responsible for translating the design to code
- Tester
 - person responsible for writing/executing test cases

Source: Ref. [3]

Guidelines

- Prior to the review
 - schedule Formal Reviews into the project planning
 - train all reviewers
 - ensure all attendees prepare in advance
- During the review
 - review the product, not its author
 - keep comments constructive, professional and task-focussed
 - stick to the agenda
 - leader must prevent drift
 - limit debate and rebuttal
 - record issues for later discussion/resolution
 - identify problems but don't try to solve them
 - take written notes
- After the review
 - review the review process

Source: Ref. [3]

Opening Moments

- 1) Don't start until everyone is present
- 2) Leader announces:
 - “We are here to review product X for purpose Y”
- 3) Leader introduces the reviewers, and explains the recording technique
- 4) Leader briefly reviews the materials
 - check that everyone received them
 - check that everyone prepared
- 5) Leader explains the type of review

Note: The review should not go ahead if:

- some reviewers are missing
- some reviewers didn't receive the materials
- some reviewers didn't prepare

Source: Ref. [3]

Structuring the Inspection

- Checklist
 - uses a checklist of questions/issues
 - review structured by issue on the list
- Walkthrough
 - one person presents the product step-by-step
 - review is structured by the product
- Round Robin
 - each reviewer in turn gets to raise an issue
 - review is structured by the review team
- Speed Review
 - each reviewer gets 3 minutes to review a chunk, then passes to the next person
 - good for assessing comprehensibility!

Source: Ref. [3]

Fagan Inspection Process

1 Overview

- communicate and educate about product
- circulate materials
- Rate: 500 SLOC per hour

2 Preparation

- All participants perform individually
- review materials to detect defects
- Rate: 100-125 SLOC per hour

3 Inspection

- a reader paraphrases the design
- identify and note problems (don't solve them)
- Rate: 130-150 SLOC per hour

4 Rework

- ↳ All errors/problems addressed by author
- ↳ Rate: 16-20 hours per 1000 SLOC

5 Follow-up

- ↳ Moderator ensures all errors have been corrected
- ↳ if more than 5% reworked, product is re-inspected by original inspection team

Source: Ref. [3]

References

- [1] Hans van Vliet, “Software Engineering: Principles and Practice”, John Wiley and Sons, Ltd., 2000.
- [2] Stephen R. Schach, “Object-Oriented and Classical Software Engineering”, McGraw-Hill Companies, Inc., 2002.
- [3] Steve Easterbrook, “Lecture Notes”, University of Toronto, 2001.