

Reviews & Inspections



Agenda

- Quality Management
- Software Quality
- Validation & Verification
- Reviews & Inspections

Software Reviews

- Collaborative techniques for evaluating software development products
 - generic term for a variety of techniques
- Aim to detect errors and improve quality

Reviews vs. Testing

- Reviews
 - analysis of static system representation to discover problems
 - ❖ static verification
 - supplemented by tool-based document and code analysis
- Testing
 - exercise and observe product behaviour
 - ❖ dynamic verification
 - execute system with test data

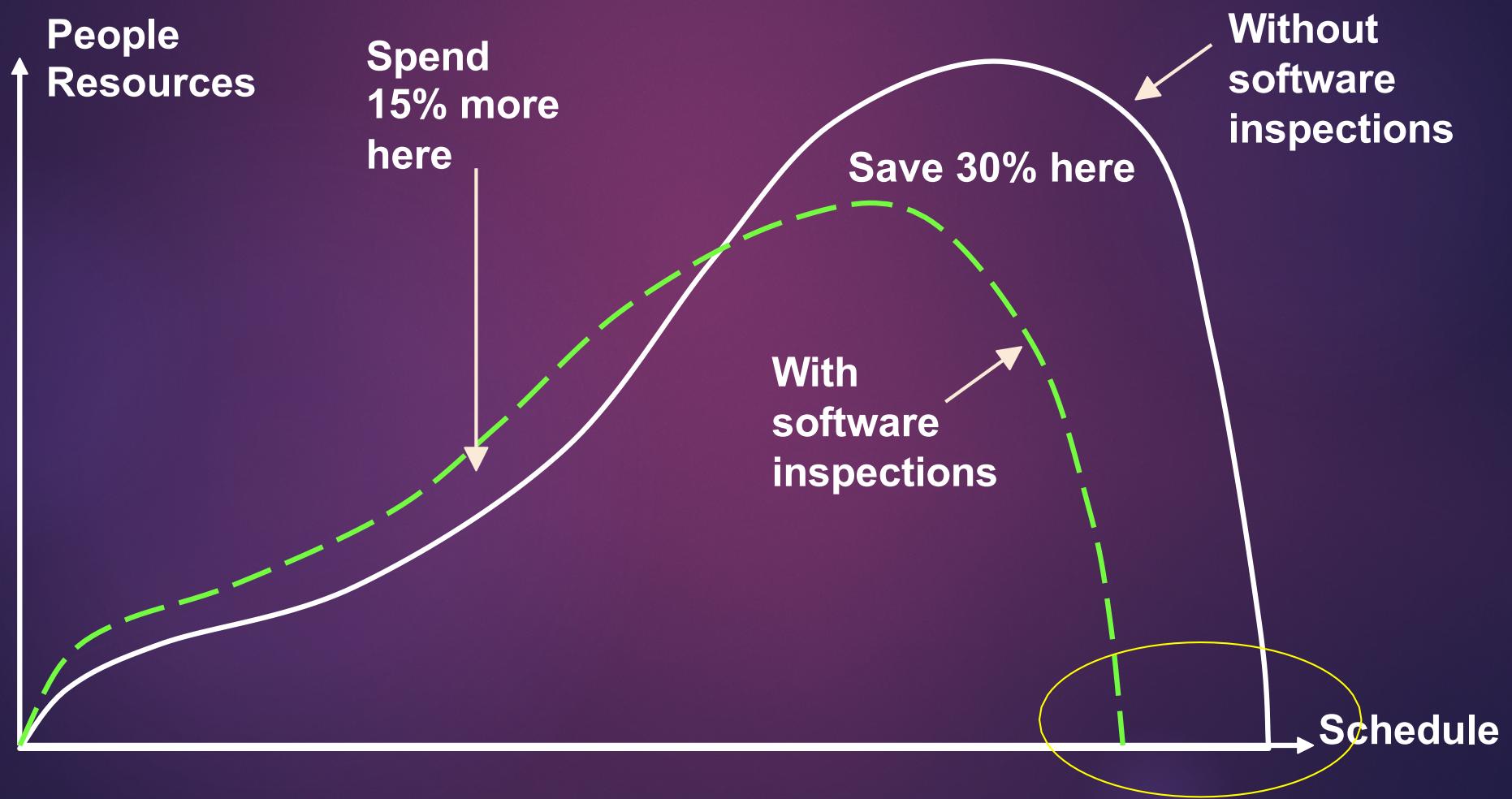
Benefits of Reviews

- Cost benefit from early error detection
 - author receives timely feedback
- Can be applied to untestable products
 - e.g. requirements specification
- During dynamic testing errors can hide other errors
 - static verification means no interaction between errors
- Distributes knowledge

Cost of Reviews

- Typically 15% of project budget
 - extra effort early in process with expectation of payoff later
- Significant learning curve

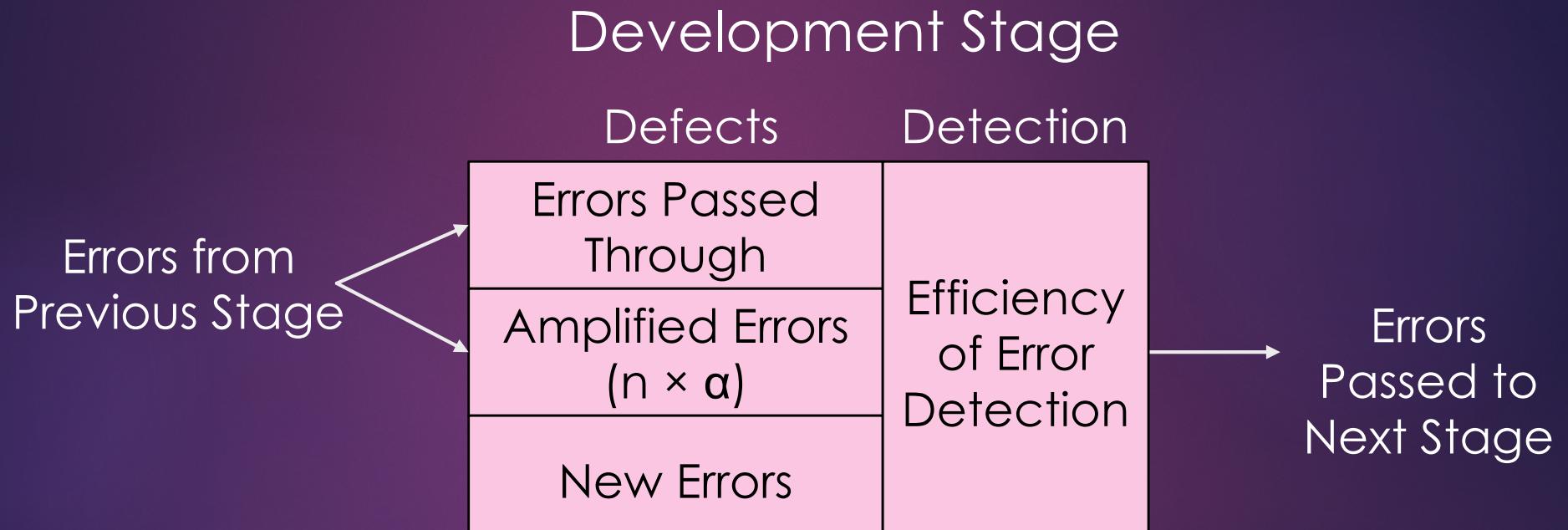
Inspection Experience



Defect Amplification

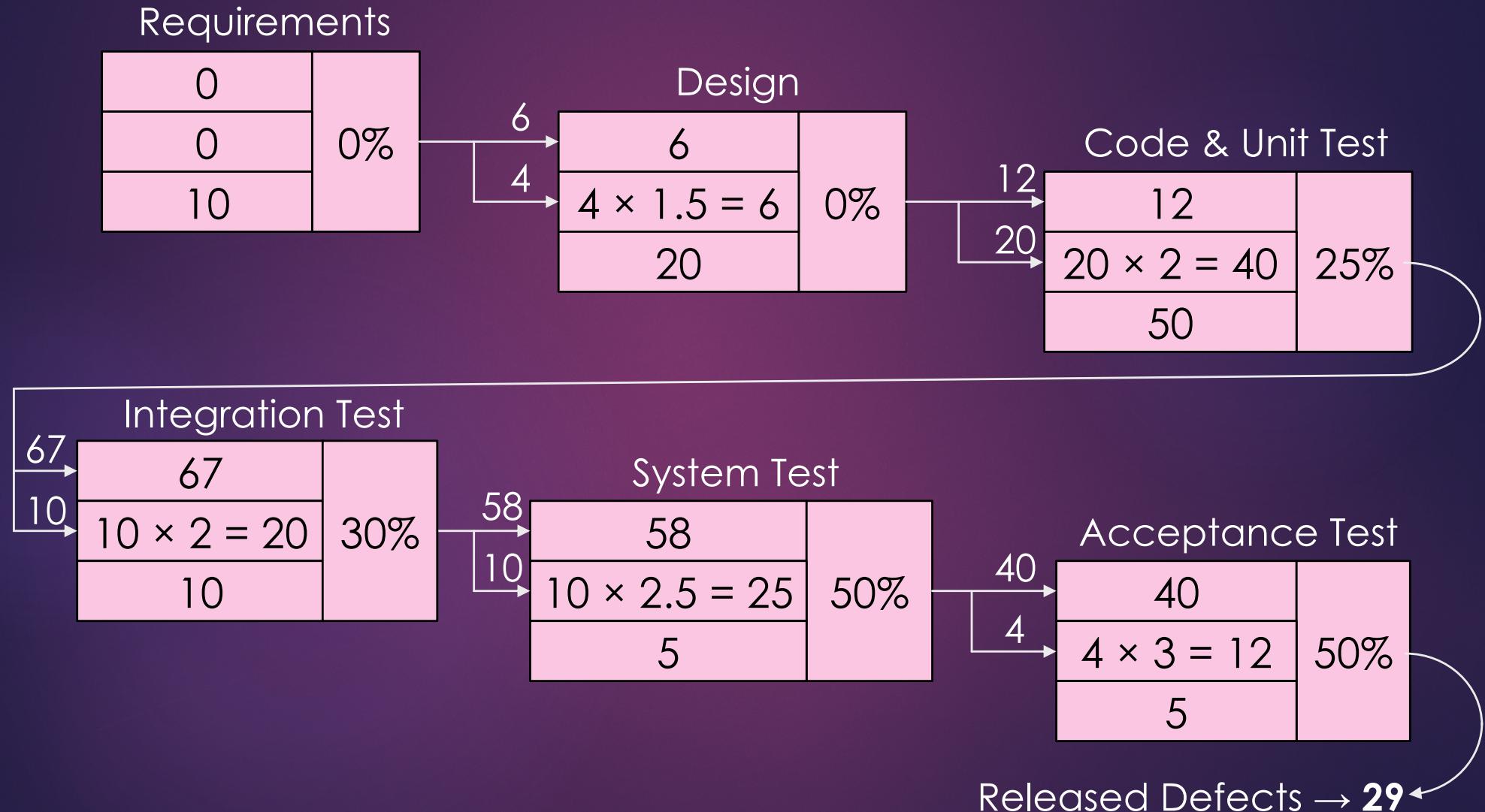
- ◆ Errors from previous activity
- ◆ Some errors are passed through
- ◆ Some get amplified
- ◆ New errors are generated
- ◆ Errors passed on to next phase
- ◆ Reviews filter out errors (up to 75%)

Defect Amplification Model

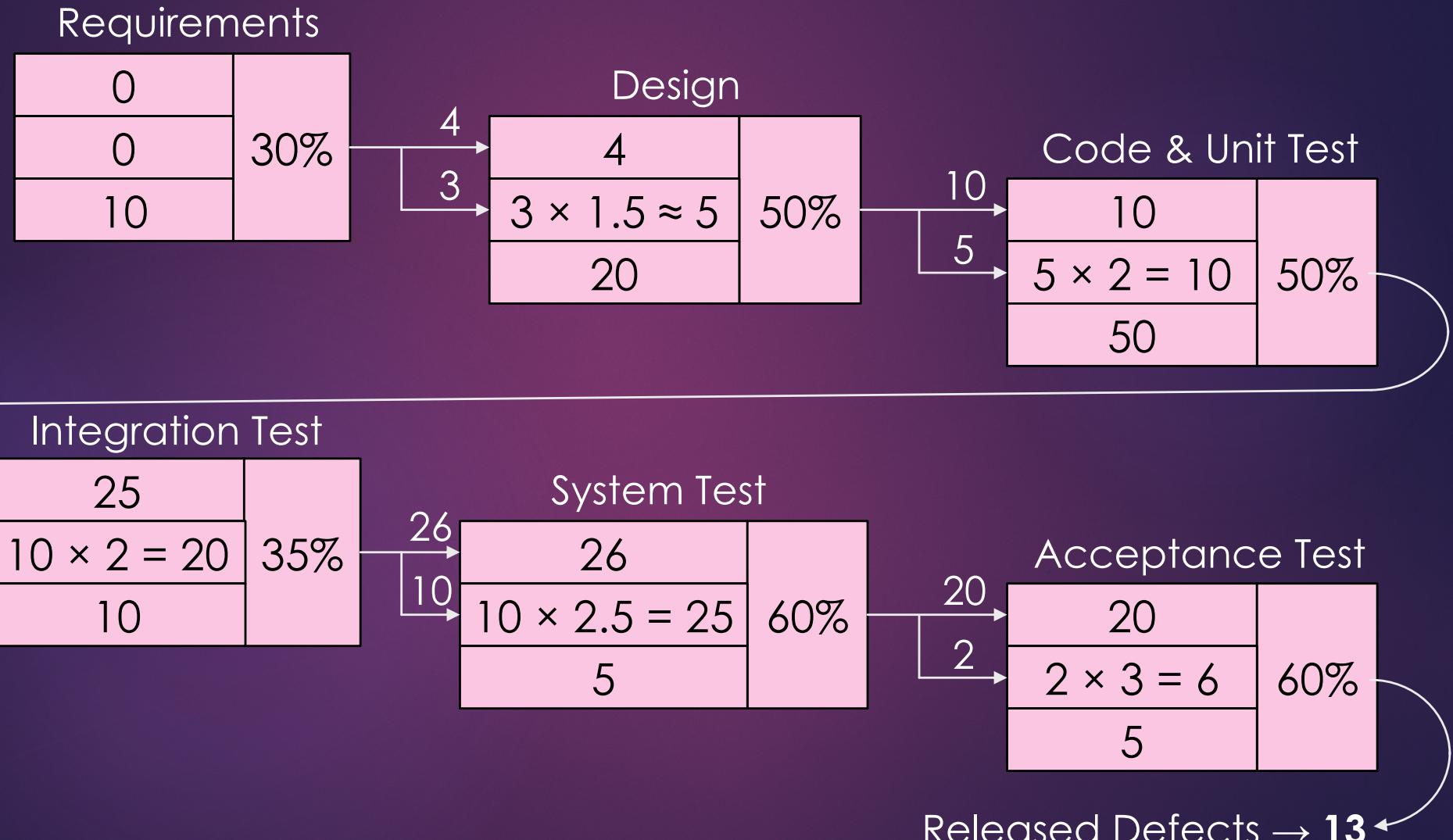


α – amplification multiplier

No Reviews Example



Reviews Example



Bell Northern Research



- ◆ 2.5M lines of code over 8 releases
- ◆ Found 1 defect per staff hour
 - 2-4 times faster than testing
- ◆ 80% of all defects found
- ◆ Avoided 33 hours of maintenance effort per defect

Types of Reviews

- Technical Review
 - conformance to standards
 - achievement of milestones
 - led by team leader
- (Fagan) Inspection
 - peer review with formal process
 - led by independent moderator
 - systematic data collection
 - focus on defect detection
 - process improvement goal

Type of Reviews (cont.)

- Structured Walkthrough (Yourdan)
 - less formal than inspection
 - ❖ no data collection
 - ❖ no preparation
 - led by producer
- Audit
 - external review of artefacts
 - independently managed
 - usually late in process

Inspection Principles

- Defined roles for participants
 - requires technical knowledge
 - preparation in advance
- Checklists used to identify defects
 - focus on finding problems
 - ❖ not solving problems
- Inspection data is recorded
 - effectiveness of inspection process
 - relationship with product quality

Participants

- Moderator
 - responsible for leading inspection process
 - schedules and conducts meeting
 - prepares reports and follows up action items
- Recorder
 - keeps record of inspection results
 - helps prepare reports
- Producer
 - responsible for work under inspection

Participants (cont.)

- Reader
 - presents work in lieu of producer
- Reviewers
 - directly concerned with and aware of work
 - required to prepare for inspection
 - should be objective and accountable
- Typically 3 to 7 people

Inspection Process

- Request
 - producer requests inspection of artefact
 - moderator is selected
- Entry
 - moderator checks artefact is ready
- Planning
 - moderator plans inspection meeting
 - ❖ e.g. who will participate
- Overview (optional)
 - meeting to distribute documents

Inspection Process (cont.)

- Preparation
 - reviewers work alone, using checklists to find defects
- Inspection Meeting
 - consolidate defects found by reviewers
 - classify defects by severity
 - find additional defects via synergy
- Rework
 - producer resolves issues from meeting

Inspection Process (cont.)

- Follow-up
 - moderator checks rework
 - schedules additional inspection if needed
- Exit
 - moderator checks artefact against exit criteria
- Release
 - artefact is released

Issue Classification

- Major
 - defect likely to
 - ❖ cause incorrect behaviour
 - ❖ require external resolution before development is complete
- Minor
 - defect likely to cause limited or no loss of functionality

Issue Classification (cont.)

- Grammatical
 - spelling, grammar, typographical errors
- Questions
 - potential defects a reviewer is unsure about
 - ❖ wants to discuss at the meeting

Inspection Preparation

- Product must be ready for inspection
- Inspection team must be selected, briefed and supplied with artefact
- Checklists and standards give guidance to reviewers
- Reviewers must prepare individually
 - recording time and errors identified
- Reviewers may be given special roles

Inspection Preparation

- Reviewers try to find as many defects as possible
 - concentrating on major defects
 - ❖ actual classification is not critical
 - re-assessed at meeting
- Queries about the artefact should be recorded for meeting
- Experience shows about 75% of defects are identified during individual preparation

Checklists

- Checklist of common errors used to drive inspection
 - more useful with artefacts in a formal language
 - ❖ e.g. code, structured specifications, formal diagrams, mathematical specifications, ...
 - notation dependent
 - ❖ reflect characteristic errors that are likely to arise in the language
- In code, the 'weaker' the type checking, the larger the checklist
 - e.g. initialisation, constant naming, loop termination, array bounds, etc.

Java Checklist

Area of Concern	Type Code	Description
Complete	CP1	Code is missing design functions
Understandable	US1	Code is hard to understand
	US2	Class/method is too big and part of it would be better split into another class/method
	US3	Nesting of loops is too deep
	US4	The solution is too tricky. Simplify.
Comments	CT1	Not clear
	CT2	Not correct
	CT3	Should focus on why, and not how
Types	TP1	Incorrect object instantiation
	TP2	Incorrect variable declaration
	TP3	Unsafe casting (e.g. long to short)

Java Checklist

Area of Concern	Type Code	Description
Names	NM1	Incorrect spelling
	NM2	Inconsistent use of names
	NM3	Should be more descriptive
	NM4	Not consistent with standards for classes, methods, variables and constants
Initialisation	IT1	Variable/argument not initialized. (Look at class/method entry and at the start of loops.)
Calls	CL1	Incorrect arguments. (Look at object instantiation and method calls.)
Output Format	OF1	Improper line spacing
	OF2	Improper spacing

Java Checklist

Area of Concern	Type Code	Description
Standards	SD1	Not to coding standards
Logic Operators	OP1	Improper use of logic operator. (e.g. look for ==, =, &, &&, , , ...)
	OP2	() not properly used
File Open & Close	FL1	File not properly defined
	FL2	File not opened
	FL3	File not closed
Overall	OV1	System issue
	OV2	Unexpected problem

Req Spec Checklist

Area of Concern	Type Code	Description
Organisation	OR1	Does not conform to template
	OR2	Internal cross-references are incorrect
Completeness	CP1	Use case details are inconsistent
	CP2	Inappropriate level of detail
Completeness	CP3	Necessary information is missing (and not marked as TBD in draft)
	CP4	Alternative scenarios are not documented
Other Issues	OI1	Not enough detail to start design
	OI2	Design detail in requirements document

Req Spec Checklist

Area of Concern	Type Code	Description
Correctness	CO1	Requirements conflict with or duplicate others
	CO2	Language is not clear, concise or unambiguous
	CO3	Requirement is not verifiable by testing, demonstration, review or analysis
	CO4	Requirement is not in scope
	CO5	Domain description is incorrect
Quality Attributes	QA1	Non-functional requirement is not measurable and testable
	QA2	Performance objective is missing
	QA3	Security and safety considerations missing
	QA4	Other quality goals missing

Inspection Meeting

- Moderator confirms all reviewers are prepared
 - otherwise meeting is delayed
- Reader leads meeting through artefact
 - reviewers raise defects and questions
- Recorder logs each defect
 - location, description, category
 - any unresolved questions
- Any new issues are discussed and logged

Inspection Meeting (cont.)

- Meeting concludes when all participants are satisfied that all issues have been raised and decision on review outcome is agreed
 - accept, verify rework, re-inspect
- Two hours is maximum duration for one meeting
 - reschedule as necessary

Reporting

- Preparation logs from reviewers
- Combined defect list from meeting
- Management summary
 - activities: who did what, when
 - summary of defects

Two Person Review

- Eliminates the moderator
 - just author and reviewer
- Immediate benefits in program quality and productivity
- Recommended as a transition technique for introducing inspections or in small organisations
- Pair programming?

Agile & Reviews

- Pair Programming
 - review code as it is written
 - ❖ shortening the loop
- Collective Code Ownership
- Collaborative Design
 - CRC Cards
 - whiteboard sketches
- Refactoring

Reading

- Sommerville
 - chapter 24
- Wiegers
 - chapter 17
- Larman
 - chapter 37

Further Reading

- Sommerville
 - chapter 13
- Larman
 - chapter 21
- *Peer Reviews in Software : A Practical Guide*
 - Karl E. Wiegers
- *Software Inspection*
 - Tom Gilb & Dorothy Graham
- *Software Inspection : An Industry Best Practice*
 - David A. Wheeler et al.
- *Software Inspection Process*
 - Robert G. Ebenau & Susan H. Strauss

Next Steps

- Lecture
 - Estimation techniques
- Tutorial
 - Review requirements specification