

Review Processes

- Quality and Reviews
 - efficacy of Q/A techniques
 - benefits of reviews
 - types of reviews
- Formal Reviews
 - the process
 - the roles
 - risks and how to manage them
- Informal Review Processes
 - differences and trade-offs

Review Processes

1

Ways to improve quality

- Try to be careful
 - follow established best practices
 - reduce number of mistakes we make
- Peer Reviews
 - get other skilled people to check our work
 - before we do further work based on it
- Testing
 - test for all the problems we can think of
 - try to find the mistakes after we make them
 - go back and fix them before we ship

Review Processes

2

Reviews

- get other sets of eyes to review our work
 - to find errors and omissions
 - to encourage developers to do better work
- review each major completed work product
 - fix requirements before we do architecture
 - fix architecture before we do the design
 - fix design before we write the code
 - understand how to test code before we write it
 - fix code before we test and ship it
- enabling us to ship better products
 - on-time, with lower development & support costs



Review Processes

3

Benefits of Reviews

- They can be better than testing
 - finds more problems than testing
 - finds problems sooner and more efficiently
- They are excellent training tools
 - process, methodology, standards, technique
- They improve information dissemination
 - reviewers learn other parts of the product
- They improve programming skills
 - as people learn from others' skills/mistakes



Review Processes

4

Many types of reviews

- Reviews for almost every project phase
 - Requirements reviews
 - Architectural reviews
 - Design reviews
 - Test Plan reviews
 - Code reviews
- Often used as acceptance criteria
 - before moving on to next project phase
- Different reviews ask different questions
 - but the process remains the same

Quality and Quality Assurance

5

Requirements Reviews

- Ensuring we are building the right thing
- user-level requirements
 - clear and well justified, widely agreed to
 - traceable and prioritized
 - relatively complete and stable
 - do we believe we can satisfy them?
- validate component-level requirements
 - reasonable, complete, consistent, testable
 - do they add up to the user-level requirements

Review Processes

6

Architectural Reviews

- Review architecture prior to design
- Is it capable of meeting requirements?
 - embraces all applicable standards
 - no performance or robustness issues
- Will it be practical to build & support?
 - all components well specified, look doable
 - reasonable use of off-the-shelf technology
 - good modularity, well abstracted interfaces
- Is there anything here we'll regret later?

Design Reviews

- Review Design prior to implementation
- Is the design reasonable?
 - it will satisfy all component requirements
 - no major concerns about it working
 - complete, correct, and relatively simple
- Is it clear how to build this component?
 - clearly achievable with existing technology
 - no significant open design questions
- Is the design testable?
 - adequately observable and controllable

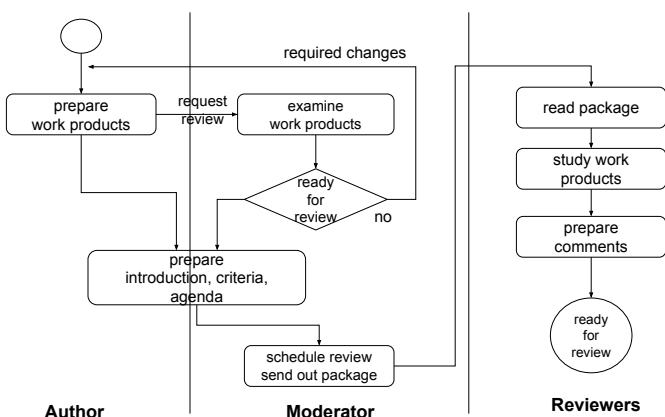
Test Plan Reviews

- Review proposed test cases
 - each clearly and adequately described
 - sufficient to cover all likely problems
 - avoid redundant or useless test cases
- Review proposed testing strategy
 - enables code to be tested as developed
 - clear how all tests will be implemented
 - good use of standard automation technology
- How much confidence will it give us?

Code Reviews

- Review Code (pre-testing vs pre-push)
- Does this code implement the design?
 - implements all specified functionality
 - appropriately handles all reasonable cases
- Is this code obviously correct?
 - un-obviousness often hides incorrectness
- Does it conform to applicable standards?
 - naming, commenting, layout conventions
 - portability, tool enabling conventions, etc.

Preparation for a Review



Author

- person who created the work product
- Preparatory tasks
 - prepare the work product for review
 - known problems have already been addressed
 - purpose of review is not to ask for help
 - prepare introductory & background materials
- During a traditional formal review
 - author does not present
 - may answer questions
 - clarify points that have been missed

Review Materials - Introduction

- background
 - what project/component are we discussing
 - what do reviewers need to know about it
 - history, key problems, important decisions, etc.
 - where can they find additional information
 - requirements, designs, issue analyses, tool docn
- goals of this review
 - specific work products will be reviewed
 - scope of this review (what is in/out of bounds)
 - what approval means

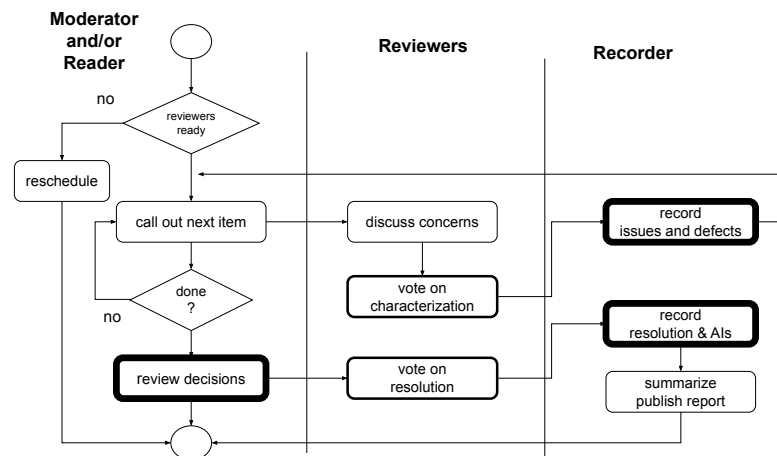
Materials - the Work Products

- the work products to be reviewed
 - specifications, designs, code, test plans, etc.
 - these should speak for themselves
 - wasteful to review them before they're ready
- a plan to structure the review
 - a table-of-contents for the work product
 - what will be reviewed, in what order
 - correct order is often critical to understanding
 - what types of issues will be covered when
 - this is the basis for the review agenda

Materials - Approval Criteria

- requirements to be satisfied
 - customer, organizational, standards
- review check-lists
 - many organizations have review check-lists
 - questions to asked, problems to consider
 - they are evolved based on experience
 - e.g. at the end of McConnell's chapters
 - these can help the reviewers
 - by reminding them of things to consider
 - they can't substitute for thought/experience

The Review Process



The Review Moderator

- experienced person, other than the author
- Preparatory tasks
 - schedule the review
 - review & distribute the preparatory materials
 - prepare and distribute an agenda
- During the review
 - keep review moving per the agenda (w/o digressions, rat-holes, scope-excursions)
 - ensure all voices heard, no key points lost
 - ensure decisions & action items are agreed to

Reader

- experienced person, other than the author
 - could be moderator
- during the review
 - guide the discussion through the code
 - following the prepared review materials
 - calling out each interesting element
 - asking for observations and issues

Reviewers (2-6)

- adequate technical experience
 - all reviewers must understand work products
 - others may attend for training purposes
 - to learn the technology or review process
 - but these people are not there as reviewers
- breadth of relevant expertise
 - people familiar with the problem domain, related products, or components
- take the process seriously
 - prepare, fully participate, there to help

Recorder

- take notes during the review
 - record all defects discovered
 - it is useful to assign a severity to each
 - record all issues raised
 - questions, suggestions, escalations, etc.
 - record decision and action items
 - accepted, major/minor revisions, further review
 - summarize all decisions at end of review
- publish a report of the review
 - recorder is often a process person, observing process & collecting metrics

Issue/Defect Resolutions

- Issue characterizations:
 - issues (to be resolved)
 - defects (to be fixed)
- Disposition must be agreed to for each:
 - major (must) ... would not work as written
 - minor (should) ... measurable improvement
 - advice ... things to be considered
 - answered questions have been dealt with

Decision and Report

- At end of meeting
 - review each outstanding defect/issue
 - clearly state the issue
 - confirm the agreed upon status
 - decide/announce the state of the project
 - **approved** (w/enumerated changes)
 - **requires another review**
 - to review non-straight-forward fixes
 - to review proposed resolutions to large issues
- These must also be in the report

Potential Problems

- scope issues
 - digressions and rat-holes
 - revisiting past decisions
- productivity issues
 - materials difficult to understand
 - reviewers haven't done the preparation
 - reviewer burn-out
- ego issues
 - discussing people rather than problems
 - telling the author how to write his/her code
 - author defends his/her decisions

Less Formal Processes

- 90% of bugs are found during preparation
 - is the formal meeting a price performer?
 - how much value do moderator and scribe add?
- Structured Walk Through
 - conducted by author, perhaps w/o preparation
 - no check lists, moderator, or written report
- Code Reading
 - give code to reviewers (similar to review)
 - reviewers send feedback directly to author
 - no meeting, moderator, or written report
- Pair Programming
 - code reviewed by partner, as it is written

How do they work?

- they can work well with a good author
- advantages
 - fewer people, less overhead, faster, cheaper
 - has potential to find most of the problems
- disadvantages
 - no moderator may mean poor agenda control
 - no formal discussion may mean lost input
 - no scribe may mean lost issues and concerns
 - less opportunity for emergent insights
 - less opportunity for training and learning

For Next Lecture

- Usability and Usability Testing:
 - Wikipedia: Usability Testing
 - usability.net: User Centered Design
 - Talin: User Interface Design Principles
 - Bay: Designing Games that Don't Suck
 - Thompson: Halo 3 – a New Science of Play
- Different Types of User Interfaces:
 - Nielsen: Web vs GUI interfaces
 - Kampe: Content Architecture
 - Kampe: CLI design

Supplementary Slides

Review Scope

- review must be kept at designated level
 - requirements ... don't design the system
 - architectural ... don't design the components
 - design ... don't over-constrain implementor
- avoid descending to lower level issues
 - may be needed to illustrate potential issues
 - suggestions can be made outside the review
- avoid re-specifying/designing the system
 - escalations can be included in the report

Too much ego involvement

- author is too defensive to get input
 - hence author can't moderate/review/record
 - ensure senior engineers set a good example
 - exclude management from review meetings
- reviewers focusing on the wrong things
 - different approaches, style, personality
 - clearly defined review scope
 - written standards and check-lists
 - moderator must manage the egos & scope