# Applied Software Project Management

## Reviews

## When are reviews needed?

- A *review* is any activity in which a work product is distributed to reviewers who examine it and give feedback.
  - Reviews are useful not only for finding and eliminating defects, but also for gaining consensus among the project team, securing approval from stakeholders, and aiding in professional development for team members.
  - Reviews help teams find defects soon after they are injected making them cost less to fix than they would cost if they were found in test.
  - All work products in a software project should be either reviewed or tested.
    - Software requirements specifications, schedules, design documents, code, test plans, test cases, and defect reports should all be reviewed.

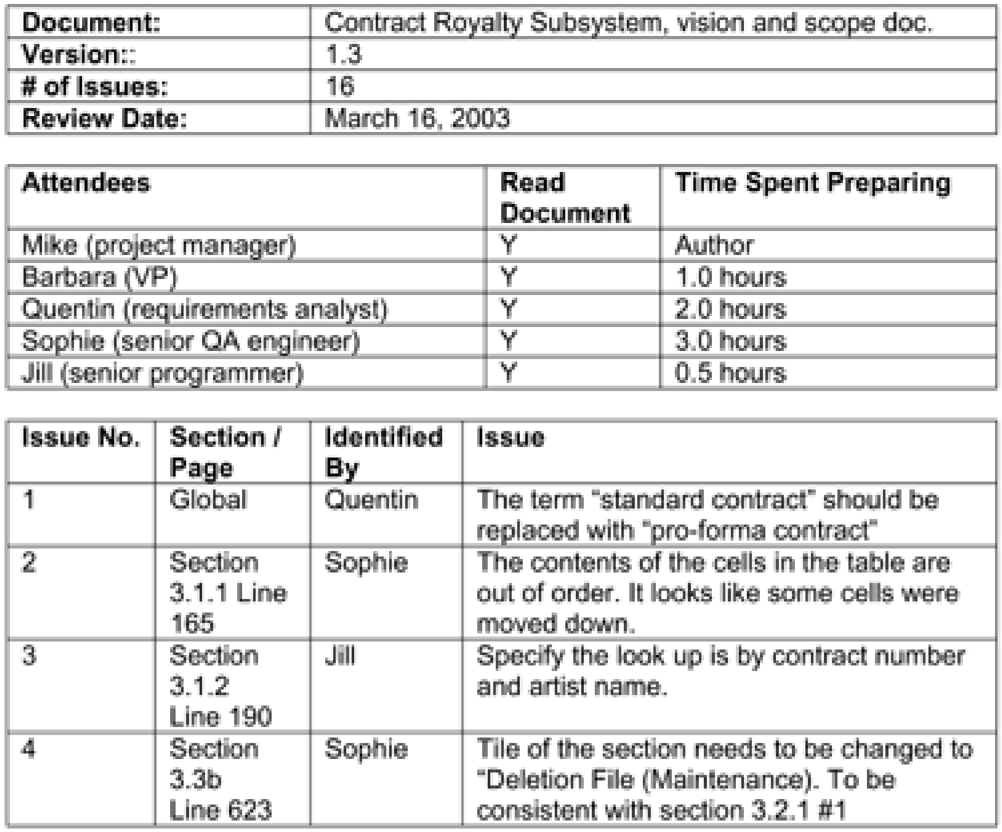
## Types of Review: Inspections

- *Inspections* are moderated meetings in which reviewers list all issues and defects they have found in the document and log them so that they can be addressed by the author.
- The goal of the inspection is to repair all of the defects so that everyone on the inspection team can approve the work product.
  - Commonly inspected work products include software requirements specifications and test plans.

# Types of Review: Inspections

- Running an inspection meeting:
  1. A work product is selected for review and a team is gathered for an inspection meeting to review the work product.
  2. A moderator is chosen to moderate the meeting.
  3. Each inspector prepares for the meeting by reading the work product and noting each defect.
  4. In an inspection, a defect is any part of the work product that will keep an inspector from approving it.
  5. Discussion is focused on each defect, and coming up with a specific resolution.
     - It’s the job of the inspection team to do more than just identify the problems; they must also come up with the solutions.
  6. The moderator compiles all of the defect resolutions into an *inspection log*

# Inspection Log Example



| | |
|---|---|
| **Document:** | Contract Royalty Subsystem, vision and scope doc. |
| **Version::** | 1.3 |
| **# of Issues:** | 16 |
| **Review Date:** | March 16, 2003 |

| **Attendees** | **Read Document** | **Time Spent Preparing** |
|---|---|---|
| Mike (project manager) | Y | Author |
| Barbara (VP) | Y | 1.0 hours |
| Quentin (requirements analyst) | Y | 2.0 hours |
| Sophie (senior QA engineer) | Y | 3.0 hours |
| Jill (senior programmer) | Y | 0.5 hours |

| **Issue No.** | **Section / Page** | **Identified By** | **Issue** |
|---|---|---|---|
| 1 | Global | Quentin | The term "standard contract" should be replaced with "pro-forma contract" |
| 2 | Section 3.1.1 Line 165 | Sophie | The contents of the cells in the table are out of order. It looks like some cells were moved down. |
| 3 | Section 3.1.2 Line 190 | Jill | Specify the look up is by contract number and artist name. |
| 4 | Section 3.3b Line 623 | Sophie | Tile of the section needs to be changed to "Deletion File (Maintenance). To be consistent with section 3.2.1 #1 |

## Types of Review: Deskchecks

- A *deskcheck* is a simple review in which the author of a work product distributes it to one or more reviewers.
  - The author sends a copy of the work product to selected project team members. The team members read it, and then write up defects and comments to send back to the author.

# Types of Review: Deskchecks

- Unlike an inspection, a deskcheck does not produce written logs which can be archived with the document for later reference.
- Deskchecks can be used as predecessors to inspections.
  - In many cases, having an author of a work product pass his work to a peer for an informal review will significantly reduce the amount of effort involved in the inspection.

## Types of Review: Walkthroughs

- A *walkthrough* is an informal way of presenting a technical document in a meeting.
  - Unlike other kinds of reviews, the author runs the walkthrough: calling the meeting, inviting the reviewers, soliciting comments and ensuring that everyone present understands the work product.
  - Walkthroughs are used when the author of a work product needs to take into account the perspective of someone who does not have the technical expertise to review the document.
  - After the meeting, the author should follow up with individual attendees who may have had additional information or insights. The document should then be corrected to reflect any issues that were raised.

## Types of Review: Code Review

- A *code review* is a special kind of inspection in which the team examines a sample of code and fixes any defects in it.
  - In a code review, a defect is a block of code which does not properly implement its requirements, which does not function as the programmer intended, or which is not incorrect but could be improved
    - For example, it could be made more readable or its performance could be improved

## Types of Review:
## Code Review

- It’s important to review the code which is most likely to have defects. This will generally be the most complex, tricky or involved code.
- Good candidates for code review include:
  - A portion of the software that only one person has the expertise to maintain
  - Code that implements a highly abstract or tricky algorithm
  - An object, library or API that is particularly difficult to work with
  - Code written by someone who is inexperienced or has not written that kind of code before, or written in an unfamiliar language
  - Code which employs a new programming technique
  - An area of the code that will be especially catastrophic if there are defects

## Types of Review: Pair Programming

- *Pair programming* is a technique in which two programmers work simultaneously at a single computer and continuously review each others' work.
- Although many programmers were introduced to pair programming as a part of Extreme Programming, it is a practice that can be valuable in any development environment.
- Pair programming improves the organization by ensuring that at least two programmers are able to maintain any piece of the software.

## Types of Review: Pair Programming

- In pair programming, two programmers sit at one computer to write code. Generally, one programmer will take control and write code, while the other watches and advises.
  - Some teams have found that pair programming works best for them if the pairs are constantly rotated; this helps diffuse the shared knowledge throughout the organization. Others prefer to pair a more junior person with a more senior for knowledge sharing.
- The project manager should not try to force pair programming on the team; it helps to introduce the change slowly, and where it will meet the least resistance.
  - It is difficult to implement pair programming in an organization where the programmers do not share the same nine-to-five (or ten-to-six) work schedule.
  - Some people do not work well in pairs, and some pairs do not work well together.