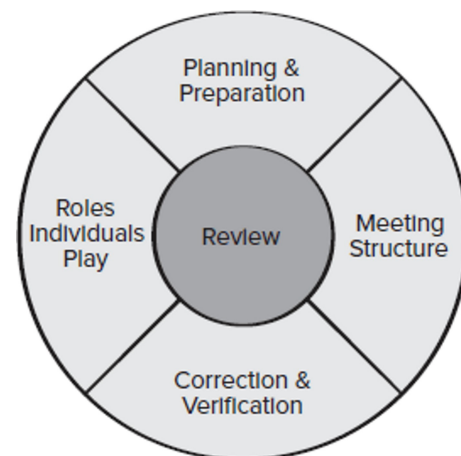


# Chapter 16: Reviews—A Recommended Approach

February 8, 2021 8:59 AM

- Reviews
  - ★ ○ meetings conducted by technical people for technical people
    - A technical assessment of a work product
  - ★ ○ A software quality assurance mechanism
    - A training ground
- Reviews are not
  - A project summary or progress assessment
  - A meeting intended to solely to impart information
  - A mechanism for political or personal reprisal
- What do we look for?
  - ★ ○ Error - a quality problem found before release
  - ★ ○ Defect - a quality problem found after release
    - These distinctions are not mainstream thinking
- Defect Amplification and Removal
  - Defect amplification - how a defect introduced early in the software engineering work flow and undetected, can and often will be amplified into multiple errors during the design and more errors in the construction
  - Defect propagation - the impact an undiscovered defect has on future development activities or product behavior
  - Technical debt - the costs incurred by failing to find and fix defects early or failing to update documentation following software changes
- Metrics - are measures
  - Effort,  $E$  - in-person hours
  - Preparation effort,  $E_p$  - the effort required to review a work product prior to the review meeting
  - Assessment effort,  $E_a$  - the effort that is expending during the actual review
  - Rework effort,  $E_r$  - the effort that is dedicated to the correction of those errors uncovered during the review
  - ★ ○ Work Product Size, WPS - lines of code or number of pages
    - Minor errors found,  $Err_{minor}$  - the number of errors found that can be categorized as minor
    - Major errors found,  $Err_{major}$  - the number of errors found that can be categorized as major
  - The total review effort
    - $E_{review} = E_p + E_a + E_r$
    - $E_{tot} = Err_{minor} + Err_{major}$
  - Defect density represents the errors found per unit of WPS
    - Defect density =  $\frac{Err_{tot}}{WPS}$
  - Example
    - The average defect density for a requirements model is 0.68 errors per page, and a new requirements model is 40 pages long,  $0.68 \times 40 =$  approximately 27 errors
    - If you only find 9 errors, you've done an extremely good job in developing the requirements model or your review approach was not thorough enough
  - Example
    - The effort required to correct a minor model error will require 4 person-hours
    - The effort required for major requirements error is 18 person-hours
    - Minor errors occur about 6 times more frequently than major errors
    - The average effort required to correct 1 error will be 6 hours  $(4 \text{ Hours} \times 6 + 18 \text{ Hours}) / 7 = 6 \text{ hours}$
  - ★ ○ Effort saved per error =  $E_{testing} - E_{reviews}$
- Informal Reviews - the benefit is immediate discovery of errors and better work product quality
  - A simple desk check with a colleague
  - A casual meeting (2 or more people)
  - The review-oriented aspects of pair programming



- ★ ▪ pair programming - encourages continuous review as a work product is created
  - Quality can suffer
- Formal Technical Reviews, FTRs, Code Inspections, Code Walkthroughs - 5 key objectives
  - ★ ○ To uncover errors in functions, logic, implementation in any representation of the software
  - ★ ○ To verify that the software meets its requirements
  - ★ ○ To ensure that the software adheres to a standard
  - ★ ○ To achieve software that is developed in a uniform manner
  - ★ ○ To make projects more manageable
- Review Meeting
  - Between three and five people should be involved in the review
  - Advance preparation - should require no more than two hours of work per person
  - The duration should be less than two hours
  - Focus is on a specific work product
- Review Players
  - Producer - the individual who has developed the work product
  - Review leader - evaluates the product for readiness, generates copies of product materials, and distributes them to two or three reviewers for advanced preparation and facilitates the meeting discussion
  - Reviewers - expected to spend between one and two hours reviewing the product, making notes, and becoming familiar with the work
    - Come prepared to evaluate
  - Recorder - records (in writing) all important issues
- Review Outcome - a decision must be made to:
  - ★ ○ Accept the product without modification
  - ★ ○ Reject the product due to severe errors
  - ★ ○ Accept the product conditionally (correct errors and no additional review)
- Review Reporting and Record Keeping
  - The recorder records all issues raised and summarizes these in an action list for the producer (formal technical review summary report) - this answers three questions:
    - What was reviewed?
    - Who reviewed it?
    - What were the issues and conclusions (Review Outcome)?
  - You should establish a follow-up procedure to ensure that items on the issues list have been properly corrected
- Review Guidelines
  - Review the product, not the producer
  - Stay on-topic
  - Limit the debate
  - Enunciate problems, but don't try to solve every problem
  - Take written notes
  - Limit the number of participants and insist doing advance preparation
  - Develop a checklist for each product that is likely to be reviewed
  - Allocate proper resources and time for FTRs
  - Conduct meaningful training for all reviewers
  - Review your early reviews
- Post-mortem Evaluations, PME
  - A mechanism to determine what went right and what went wrong
  - Attended by members of the software team and stakeholders who focus on excellences and challenges
- Agile reviews
  - User stories are reviewed and ordered by priority
  - Daily scrum meetings to ensure members are working to try to catch and defects
  - Sprint review meetings using guidelines
  - Sprint retrospective meetings to capture the lessons learned from previous meetings