Lecture 12: Chapter 15

- **Review Techniques**

*Slide Set to accompany*

*Software Engineering: A Practitioner’s Approach, 7/e*

*by Roger S. Pressman*


*For non-profit educational use only*

May be reproduced ONLY for student use at the university level when used in conjunction with *Software Engineering: A Practitioner's Approach, 7/e*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information MUST appear if these slides are posted on a website for student use.
... there is no particular reason why your friend and colleague cannot also be your sternest critic.

Jerry Weinberg
What Are Reviews?

- a meeting conducted by technical people for technical people
- a technical assessment of a work product created during the software engineering process
- a software quality assurance mechanism
- a training ground
What Reviews Are Not

- A project summary or progress assessment
- A meeting intended solely to impart information
- A mechanism for political or personal reprisal!
What Do We Look For?

- Errors and defects
  - *Error*—a quality problem found *before* the software is released to end users
  - *Defect*—a quality problem found only *after* the software has been released to end-users
- We make this distinction because errors and defects have very different economic, business, psychological, and human impact
- However, the temporal distinction made between errors and defects in this book is *not* mainstream thinking
Overall

- Effort expended with and without reviews

![Graph showing effort expended with and without reviews.](image)
Informal Reviews

Informal reviews include:
- a simple desk check of a software engineering work product with a colleague
- a casual meeting (involving more than 2 people) for the purpose of reviewing a work product, or
- the review-oriented aspects of pair programming

pair programming encourages continuous review as a work product (design or code) is created.
- The benefit is immediate discovery of errors and better work product quality as a consequence.
Formal Technical Reviews

- The objectives of an FTR are:
  - to uncover errors in function, logic, or implementation for any representation of the software
  - to verify that the software under review meets its requirements
  - to ensure that the software has been represented according to predefined standards
  - to achieve software that is developed in a uniform manner
  - to make projects more manageable

- The FTR is actually a class of reviews that includes *walkthroughs* and *inspections*. 
The Review Meeting

- Between three and five people (typically) should be involved in the review.
- Advance preparation should occur but should require no more than two hours of work for each person.
- The duration of the review meeting should be less than two hours.
- Focus is on a work product (e.g., a portion of a requirements model, a detailed component design, source code for a component)
The Players

- Review leader
- Standards bearer (SQA)
- Producer
- Oracle
- Recorder
- Reviewer
- User rep
- Maintenance
The Players

- **Producer**—the individual who has developed the work product
  - informs the project leader that the work product is complete and that a review is required
- **Review leader**—evaluates the product for readiness, generates copies of product materials, and distributes them to two or three reviewers for advance preparation.
- **Reviewer(s)**—expected to spend between one and two hours reviewing the product, making notes, and otherwise becoming familiar with the work.
- **Recorder**—reviewer who records (in writing) all important issues raised during the review.