Peer Reviews

Dr. Mark C. Paulk

SE 4367 – Software Testing, Verification, Validation, and Quality Assurance

Three Mechanisms for V&V

Testing

- **"test" can be a generic word – see Weinberg's Perfect Software and Other Illusions About Testing**
  - **does (not) meet need**
  - **does (not) meet all requirements**
  - **(un)acceptable costs or constraints**
  - **grossly unreliable**
  - **poor performance**
- **by testing, we typically assume dynamic execution of a program**
  - **black box, white box, unit, integration, system, regression, etc., testing**

Formal methods

- formal specifications
- model / property checking
- proofs of correctness
  - Correctness attempts to establish the program is error-free; testing attempts to find if there are any errors in the program.

Peer reviews

- A review of a software work product, following defined procedures, by peers of the producers of the product for the purpose of identifying defects and improvements. (Software CMM)
  - managers are not peers…
  - customers are not peers…

Peer Review

A review of a software work product, following defined procedures, by peers of the producers of the product for the purpose of identifying defects and improvements. (Software CMM)

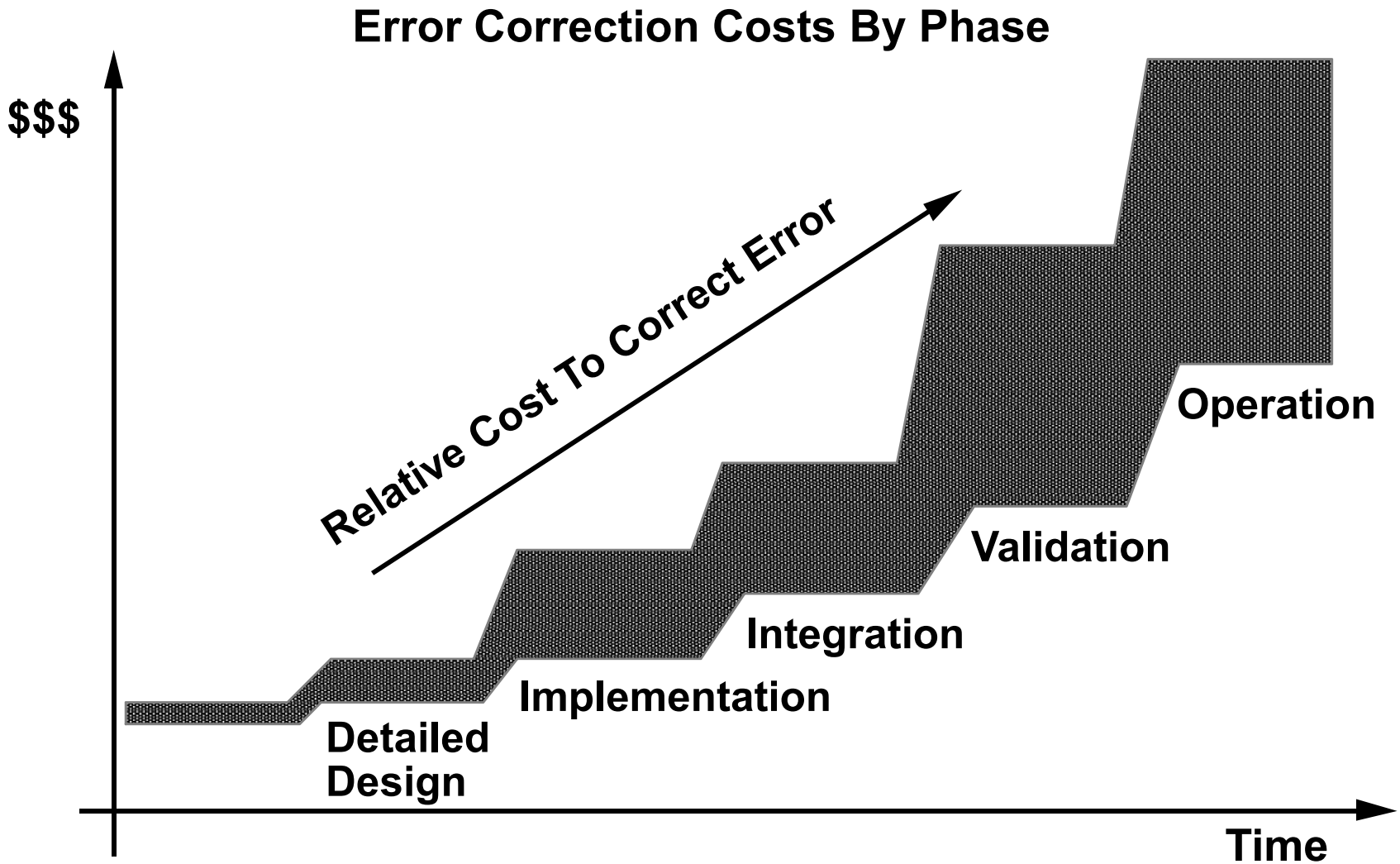
Variants of peer reviews include

- **(structured) walkthroughs**
- **(Fagan-style) inspections**
- **scenario-based inspections**
- **active design reviews**
- **…**

Alternative practices

- **pair programming**
- **formal methods, proof of correctness**

Value of Fixing Defects Early



Barry Boehm, Software Engineering Economics, 1981.

Walkthroughs

A static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a segment of documentation or code, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems.

(IEEE 610)

See Weinberg's The Psychology of Computer Programming for a discussion of walkthroughs and egoless programming.

Inspections

A static analysis technique that relies on visual examination of development products to detect errors, violations of development standards, and other problems.

- **types include code inspection; design inspection**

(IEEE 610)

Inspections are the most rigorous form of peer review… and the most effective.

Value of Peer Reviews

Any kind of peer review is better than none.

- **inspections (approximately 5:1 ROI)**
- **structured walkthroughs (approximately 3:1 ROI)**

No longer an argument over whether peer reviews are worthwhile.

- **debates are over "how"**
- **recognizing the value does not mean that we do them systematically**
- **knowing how to do them does not mean we do them correctly or consistently**

Empirical Data – The Good News

R.L. Glass, "Inspections - Some Surprising Findings," Communications of the ACM, April 1999.

The three best software engineering practices: inspections, inspections, and inspections.

Fully 90% of software errors can be found by inspections before the first test case is run.

Because of all the hard work in doing inspections, most companies don't do many inspections, and some do none at all. At best, the practice is "we inspect our key components."

Inspections are the most useful, most cost-effective form of error removal.

Inspect Requirements and Design

(Software Program Managers Network – Critical Software Practice #14)

All products that are placed under CM and are used as a basis for subsequent development need to be subjected to successful completion of a formal inspection prior to its release to CM.

The inspection needs to follow a rigorous process defined in the software development plan and should be based on agreed-to entry and exit criteria for that specific product.

At the inspection, specific metrics should be collected and tracked which will describe defects, defect removal efficiency, and efficiency of the inspection process.

All products to be placed under CM should be inspected as close to their production as feasible.

Inspections should be conducted beginning with concept definition and ending with completion of the engineering process.

The program needs to fund inspections and track rework savings.

Elements of Inspections

Six well-defined inspection steps

- **planning**
- **overview**
- **preparation**
- **meeting**
- **rework**
- **follow-up**

Four well-defined inspection roles

- **moderator**
- **recorder**
- **reader**
- **producer**

Formal collection of process and product data

The product being inspected

A supporting infrastructure

A.F. Ackerman, L.S. Buchwald, and F.H. Lewski, "Software Inspections: An Effective Verification Process," IEEE Software, May 1989.

"Rules" for Inspections

Optimum number of inspectors is four.

Review rate should be about 140 lines of text / hr (no more than 280 LOT/hr) for design documents.

Review rate should be about 125 SLOC/hr (no more than 250 SLOC/hr) for code.

Inspection meetings should not last more than two hours.

No more than two inspection meetings per day.

M.E. Fagan, "Design and Code Inspections to Reduce Errors in Program Development," IBM Systems Journal, 1976.
M.E. Fagan, "Advances in Software Inspections," IEEE Transactions on Software Engineering, July 1986.

Key Properties of Inspections

Formal moderator training

Definite participant roles

Moderator "drives" the inspection

Use of "how to find errors" checklists

Use distribution of error types to look for

Follow-up to reduce bad fixes

Less future errors because of detailed error feedback to the individual programmer

Improved inspection efficiency from analysis of results

Analysis of data identifies process problems which leads to improvement – *systemic defects*

Evidence-Based Inspections

Ron Radice, High Quality Low Cost Software Inspections, 2002.

Preconditions

- **clear and visible management support**
- **defined policy**
- **good training for all**
- **effective procedures**
- **proper planning**
- **adequate resources**

Cost of Inspections

Software CMM Level	Inspection as % Cost
1	8.5
2	10.2
3	11.4
4	13.6
5	15.3

Radice, page 9-295 & 9-296

Return-on-Investment *(1 of 3)*

Defects Found Without Inspections			
	Found	**Relative Cost/Defect**	**Full Cost**
Inspections	**0**	**1**	**0**
All Tests	**90**	**10**	**900**
Users	**10**	**100**	**1000**
Total	**100**	**-**	**1900**

Radice, pages 9-298 to 9-299

Return-on-Investment *(2 of 3)*

Defects Found With Inspections at 50% Effectiveness			
	Found	**Relative Cost/Defect**	**Full Cost**
Inspections	**50**	**1**	**50**
All Tests	**45**	**10**	**450**
Users	**5**	**100**	**500**
Total	**100**	**-**	**1000**

Return-on-Investment *(3 of 3)*

Defects Found With Inspections at 90% Effectiveness			
	Found	**Relative Cost/Defect**	**Full Cost**
Inspections	**90**	**1**	**90**
All Tests	**9**	**10**	**90**
Users	**1**	**100**	**100**
Total	**100**	**-**	**280**

Case Studies of Inspections

IBM inspections showed up

- **82% of all detected faults (1976)**
- **70% of all detected faults (1978)**
- **93% of all detected faults (1986)**

Switching system

- **90% decrease in the cost of detecting faults (1986)**

JPL

- **four major faults, 14 minor faults per 2 hours (1990)**
- **savings of $25,000 *per inspection***
- **number of faults decreased exponentially by phase (1992)**

Inspection Effectiveness and Maturity

Software CMM Level	Inspection Effectiveness
1	<50%
2	50-65%
3	65-75%
4	75-90%
5	>90%

Radice, page 1-40

Preparation and Meeting Rates

Work Product Type	Rates
Architecture & requirements documents	2-3 pages/hr
High-level & low-level design	3-4 pages/hr
Code & test cases	100-150 LOC/hr
Unit test plan	4-5 pages/hr
All test plans	5-7 pages/hr
User documentation	8-12 pages/hr
Fixes & changes	50-75 LOC/hr

Radice, page 3-95

Team Size

Fagan: Four people constitute a good-sized inspection team.

Buck: Little difference in effectiveness for teams of 3, 4, and 5 participants.

Freedman and Weinberg: Select the reviewers to ensure that the material is adequately covered.

Porter and Votta: Inspections with two reviewers were no less effective than those with four.

Systemic Defects

Causal analysis at the end of the inspection

What "systemic defects" should be prevented by process changes?

Target: 90% of systemic defects fixed within five working days.

Michael Fagan, 2002

Some Lessons Learned

Ed Weller, "Lessons from Three Years of Inspection Data," IEEE Software, September 1993.

- **Inspect before unit test.**
- **An inspection team's effectiveness and efficiency depend on how familiar they are with the product and what their inspection-preparation rate is.**
- **Expect a drop in effectiveness as you transition from pilot to general use, and be prepared to address the problems with process metrics.**
- **Good inspection results can create false confidence. Inspections are not a silver bullet. Be sure to inspect all basic design documents.**
- **Investigate the work product before deciding that process metrics indicate an ineffective inspection process.**
- **Inspections can replace unit test with significant cost savings.**
- **No matter how well they are executed, inspections cannot overcome serious flaws in the development process.**

Barriers to Peer Reviews

Belief that peer reviews are too expensive

- **four engineers per review**
- **150 SLOC/hr for code reviews**
- **max of 2 hours per review**
- **max of 2 reviews per day for an engineer**

Schedule pressure leaves no time for reviews

Hostile reviews – poor interpersonal skills, poor facilitation

Use of review results for performance evaluations

Summary – Things to Remember

3:1 ROI for walkthroughs, 5:1 ROI for inspections

Peer reviews early in the life cycle maximize ROI

Four roles in inspections

Rules for doing inspections right

Questions and Answers