Peer Reviews

Presented By: Mark Paulk

Brought To You By:

ASQ Software Division
The Global Voice of Quality™

Sponsored By:

the WestfallTeam
Partnering for Software Excellence

ASQ Software Division Invites You to Attend

Institute for Software Excellence

Held concurrently with the ASQ
World Conference on Quality and Improvement

May 6 – 8, 2013
in Indianapolis, Indiana
at the Indiana Convention Center

For ongoing information:

- Visit the ISE website at: asq.org/conferences/institute-for-software-excellence/index.html
- Visit the ASQ Software Division website at: asq.org/software/

Logistics

- Attendees are on mute
- Type your questions into the Chat area
  - Louise will ask questions between slides
- A recording of this webinar will be available online
- Certificates are available for RUs, PUs, etc.
- You will receive an email tomorrow telling you:
  - How to request a certificate of attendance
  - How to access the recording

Counting Defects

When programmers make mistakes, the defects injected are on the order of 10% of the SLOC.

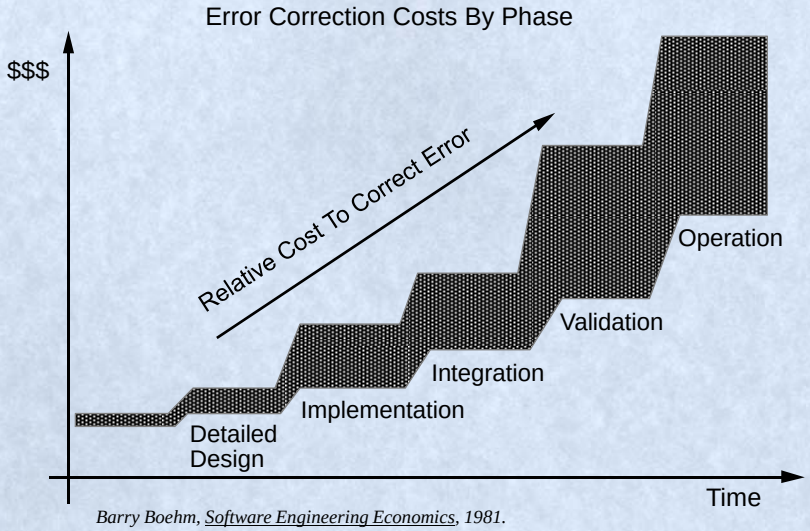
- over 100 defects / KSLOC from PSP

The "normal" development process will remove ~50% of defects before they are ever counted.

Defects will be distributed among requirements, design, code, ...

- there will be ripple effects from defects injected earlier in the process

Value of Fixing Defects Early



Barry Boehm, Software Engineering Economics, 1981.

Empirical Data – The Good News

Robert 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.

Software CMM v1.1

Level	Focus	Key Process Areas
5 Optimizing	*Continuous process improvement*	Defect Prevention Technology Change Management Process Change Management
4 Managed	*Product and process quality*	Quantitative Process Management Software Quality Management
3 Defined	*Engineering processes and organizational support*	Organization Process Focus Organization Process Definition Training Program Integrated Software Management Software Product Engineering Intergroup Coordination Peer Reviews
2 Repeatable	*Project management processes*	Requirements Management Software Project Planning Software Project Tracking & Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management
1 Initial	*Competent people (and heroics)*	

Software CMM v1.1

Peer Reviews (PR)

The purpose of Peer Reviews is to remove defects from the software work products early and efficiently.

- *An important corollary effect is to develop a better understanding of the software work products and of defects that might be prevented.*

Goals

- Peer review activities are planned.
- Defects in the software work products are identified and removed.

CMMI for Development v1.3

Level	Process Characteristics	Process Areas	
5 Optimizing	*Focus is on quantitative continuous process improvement*	Causal Analysis & Resolution Organizational Performance Management	
4 Quantitatively Managed	*Process is measured and controlled*	Organizational Process Performance Quantitative Project Management	
3 Defined	*Process is characterized for the organization and is proactive*	Requirements Development Technical Solution Product Integration Verification Validation	Organizational Process Focus Organization Process Definition Organizational Training Integrated Project Management Risk Management Decision Analysis & Resolution
2 Managed	*Process is characterized for projects and is often reactive*	Requirements Management Project Planning Project Monitoring & Control Supplier Agreement Management Product & Process Quality Assurance	Configuration Management Measurement & Analysis
1 Initial	*Process is unpredictable, poorly controlled, and reactive*		

CMMI-DEV v1.3 Engineering – Level 3 VER

Verification

Ensure that selected work products meet their specified requirements.

Specific Goals

1) **Prepare for verification.**
2) **Perform peer reviews.**
3) **Verify selected work products.**

Inspect Requirements and Design

(SPMN – 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.

…

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.

Implementing Peer Reviews

Variants of peer reviews include

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

Alternative practices

- pair programming
- formal methods, proof of correctness

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

Elements of Inspections

Six well-defined inspection steps

Four well-defined inspection roles

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*

Inspection Effectiveness Factors

Moderator effectiveness

Material meets entry criteria

Domain knowledge

Language knowledge

Phantom inspector (synergy of meeting)

Meeting length within two hours

Size of team is four

Inspection rate (code = 125 LOC/hr, max 250 LOC/hr)

Preparation rate (code = 100 LOC/hr, max 200 LOC/hr)

Michael Fagan, 2002

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

Typically between 8% and 20% of project budget for inspections

Radice, page 9-295 & 9-296

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

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

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 (1976): Four people constitute a good-sized inspection team.
- Buck found 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*.**

- You may have to sacrifice some data accuracy to make data collection easier.
- 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.
- Good inspection results can create false confidence. Inspections are not a silver bullet. Be sure to inspect all basic design documents.
- Inspections can improve the quality of maintenance fixes.
- 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

Reminders

- A recording of this webinar will be available online
- Certificates are available for RUs, PUs, etc.
- You will receive an email tomorrow telling you:
  - How to request a certificate of attendance
  - How to access the recording

Questions and Answers

Copyrights and Trademarks

® ***Capability Maturity Model, Capability Maturity Modeling, CMM Integration, and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.***