
A Primer on Software Inspections

Khaled El Emam
National Research Council of Canada

What Are Software Inspections ?

-
- A formal process for early defect detection (it is not a walkthrough nor a review)
 - Has other names (e.g., Formal Technical Review)
 - No managers involved
 - A manual activity (labor intensive)
 - Can be applied to any software engineering artifact
 - In practice, most commonly applied to source code

Benefits of Software Inspections - I

- One of the most effective and efficient methods for defect detection and correction in software engineering
- Can be used for training new staff
- A vehicle for enforcing standards (e.g., coding standards)
- Helps to create a “team spirit”
- Facilitates information (process and design/programming) transfer amongst the members of the team

Benefits of Software Inspections - II

- An excellent source of data for defect management activities
- Can lead to improved maintainability
- The same principles are applicable to many document types (e.g., requirements, design, code, and test cases) irrespective of the paradigm (procedural and object-oriented)

Calculating Return on Investment (ROI)

Principles

- Defect detection and correction in later phases of a project are much more expensive (some quoted numbers are 100:1 or even 200:1 for post-release defects)
- Effort spent now to detect and correct defects will lead to savings later on (so you invest now to save money later on)



Calculating Return on Investment (ROI)

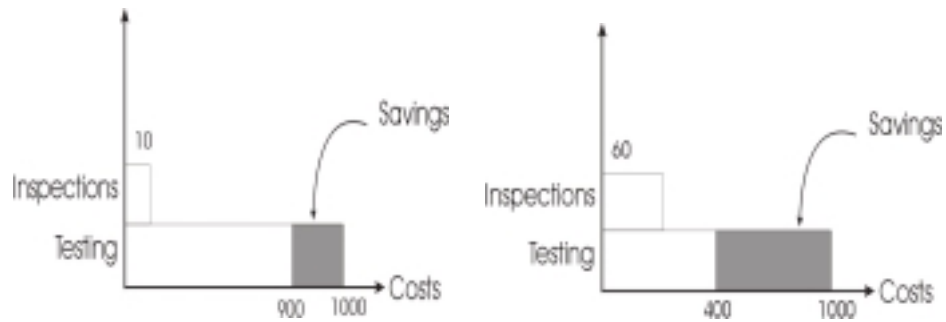
Basic Calculation

$$ROI = \frac{\text{cost saved}}{\text{cost consumed}}$$



Calculating Return on Investment (ROI)

Problems with Basic ROI



Calculating Return on Investment (ROI)

Improved Calculation

$$ROI = \frac{\text{cost saved}}{\text{virtual testing cost}}$$

Benchmarks - I

Proportion of Defects Found

	Minimum	Most Likely	Maximum
Code	0.25	0.57	0.84
Design	0.19	0.57	0.70

Benchmarks - II

Effort per Defect Found (hours/defect)

	Minimum	Most Likely	Maximum
Code	0.58	1.58	2.9
Design	0.67	1.46	2.7
Testing	4.5	6	17

Benchmarks - III

ROI

Level (% above)	CT	DT	DCT
5 (20%)	>0.48	>0.53	>0.46
4 (40%)	>0.43	>0.47	>0.39
3 (60%)	>0.37	>0.41	>0.33
2 (80%)	>0.31	>0.35	>0.27
1	<=0.31	<=0.35	<=0.27

Inspection Process

-
- Planning
 - [Overview]
 - Preparation
 - Meeting
 - Correction
 - Followup

Inspection Meetings

- Conflicting evidence (and opinions) as to their utility (meeting gains, soft benefits, interval stretch)
- Defects are better found during preparation (an individual activity)
- Reading techniques can dramatically improve individual preparation
 - ad-hoc
 - checklist
 - scenario-based reading

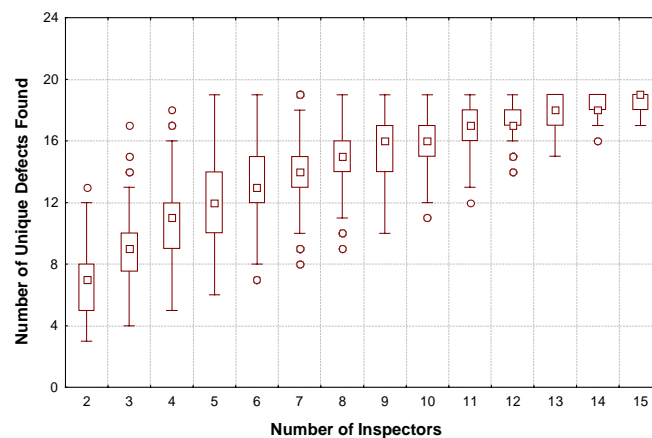
Managing Inspections - I

- Choosing the size of the inspection team
- Choosing the size of the inspected document
- How do you partition documents for inspection ?
- How much time should be spent in preparation and meeting ?
- Should a document be reinspected ?
- Who should receive moderator training ?

Managing Inspections - II

- Proximity of inspectors within the organization
- The necessary source material
- Do we have to inspect everything ?
- Definition of defects
- Inspecting changes

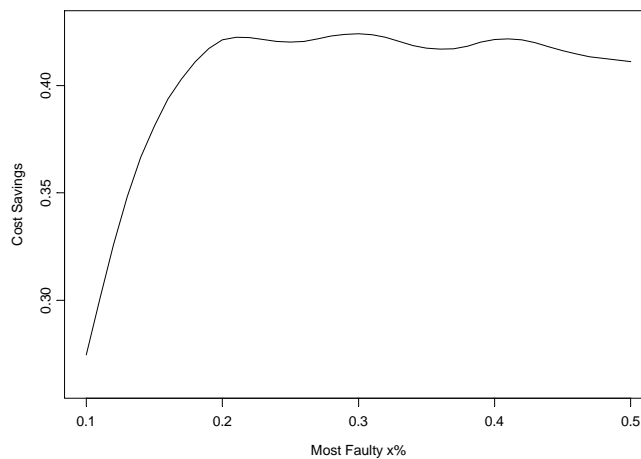
Optimal Team Size - I



Optimal Team Size - II

ϕ	Optimal Team Size							
	Without Fixing Effort				With Fixing Effort			
	No meeting	1 hr	1.5 hr	2 hrs	No Meeting	1 hr	1.5 hr	2 hrs
1	2	2	2	2	2	2	2	2
2	3	2	2	2	3	2	2	2
3	5	4	3	3	3	2	2	2
4	6	5	5	5	5	4	3	3
5	8	7	6	5	6	5	4	5
6	10	9	8	7	8	7	6	5
7	11	9	9	9	11	9	8	7
8	13	9	9	9	11	9	9	9
9	13	11	9	9	13	9	9	9
10	15	15	15	9	15	11	9	9
11	15	15	15	15	15	15	11	9
12	15	15	15	15	15	15	15	11
13	15	15	15	15	15	15	15	15
14	15	15	15	15	15	15	15	15
15	15	15	15	15	15	15	15	15

Inspecting An Optimal Subset



Further Information

- Object-oriented software quality:

<http://www.object-oriented.org>