SE 329 – Software Project Management

Project Quality Management

Lotfi ben Othmane

Course Outcome

Understand how to manage quality in software projects

Plan

1. What is quality in SE?
2. How to identify the quality requirements for an SE project?
3. Why should we care about quality?
4. How to develop a quality management plan for a project?

Resources

- A Guide to the Project Management Body of Knowledge (PMBOK), Chapter Project Quality management

- Case study at Bombardier - Laporte et al. “Measuring the Cost of Software Quality of a Large Software Project at Bombardier Transportation: A Case Study,” *Software Quality Professional Magazine,* 2012 (the figures are taken from the paper)

- Spinellis wt al. “Evaluating the Quality of Open Source Software,” Electronic Notes in Theoretical Computer Science 233 (2009) 5–28

Quality

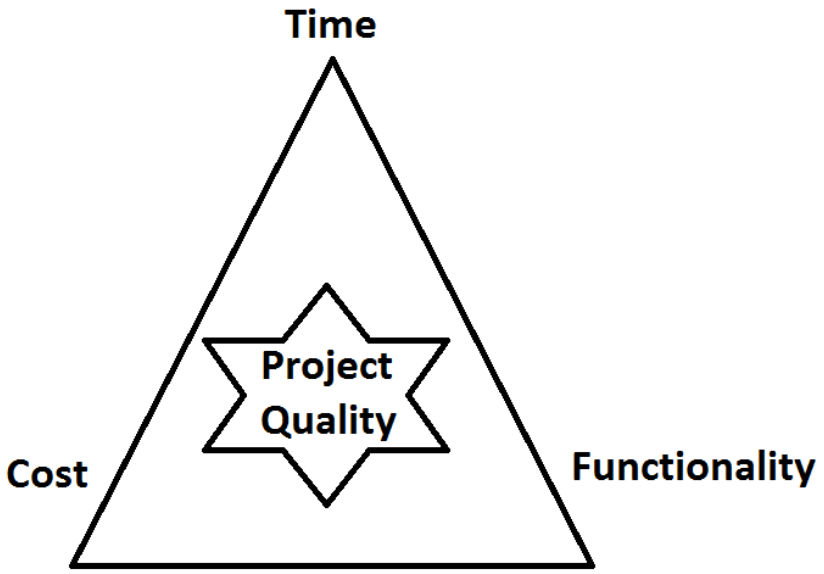
You are given two Android chat apps that have the same set of features and are both used by your friends for free to install on your phone.

Which one would you use and why?

Quality and Projects Management

Quality is a constrained optimization of

1) Time
2) Cost
3) Functionality



Quality

Quality is:

1. Meeting the needs and expectation of the customers

Quality is not:

1. A feature of a substance or material
2. How good or bad something is
3. Something of high standard

Importance of Quality

- Customer satisfaction
- Reputation
- Cost of maintenance

Quality Metrics

- We describe quality using a set of attributes
  - Example: #of defects/1000 LOC
- The quality is the aggregation of the measurements of the attributes

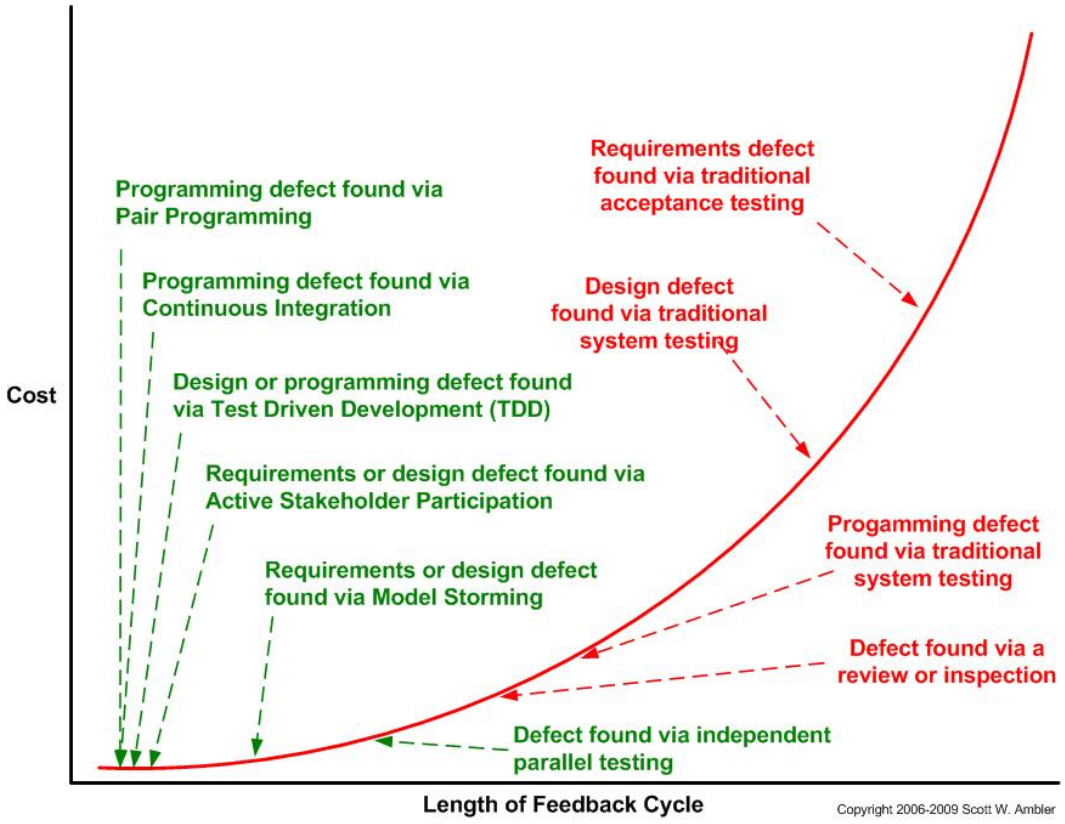
Quality in Software Engineering

1. Reliability/defect
2. Quality attributes performance, security, etc.
3. Maintainability

Software Defects

1. Code bugs
2. Compliance with regulations
3. Compliance with requirements

- Metrics:
  - # of bugs /1000 lines of code
  - Etc.



http://www.agilemodeling.com/essays/costOfChange.htm

Quality Attributes

- These include:
  - Performance
  - Scalability
  - Security
  - Usability
- Metrics: List of unsatisfied quality attributes
- Corrections activities implies redesign of the software and major REWORK.

Maintainability

- Maintainability qualify the easiness to maintain the software
- Metrics include
  - Coupling
  - Cohesion
  - Etc.

Discussion - Compare the Qualities of Four Operating Systems

- The paper “Evaluating the Quality of Open Source Software” compares a set of quality metrics for four Oss: FreeBSD, Linux, Solaris, Windows (on Canvas)

1. Which OS has the best quality?

2. How confident are you about the quality metrics that you choose?

Examples of Quality Requirements

1. All severe defects must be addressed
2. All test cases related the major requirements must run successful before delivering the product to the customer.
3. The software must be tested wrt to SQL injection before delivering it to the customer
4. The performance of the system must comply with the customers' requirements
5. There should be a trace between all the requirements and the software modules
6. The developers must apply the coding style of the company
7. All code must be reviewed before pushed to the main software repository

Sources of Quality Requirements

1. Requirement documents
2. Project scope
3. Organization policies
4. Regulations
5. Risk

1. How to assess the cost of software quality?
2. How much does software quality cost?
3. How to control the cost of quality?

Quality Cost

- Cost of non conformance
  - Returns
  - Bugs
- Reward of conformance
  - Customers satisfied
- High quality work – do it right!

Cost of Software Quality

1. Cost of prevention
2. Appraisal or evaluation cost
3. Cost of anomalies and non-compliance/repair

Quality Control Activities

1. Process improvements
2. Requirement traceability
3. Prototyping, test, debug
4. Design review
5. Testing activities
6. Software validation and verification
7. Maintenance rework
8. Release preparation
9. Software problem correction
10. Code review
11. Etc.

Practice 7 - Coding Style

Google Java Style Guide:
https://google.github.io/styleguide/javaguide.html#s6.2-caught-exceptions

1. What are the naming practices for Java?

2. What are the good practices of error handling?

Identify Bad Code Automatically

Linting
Coding style

THE FIREFOX BUILD SYSTEM
Mach
Try Server
Build System
TaskCluster Task-Graph Generation
Managing Documentation

TESTING & TEST INFRASTRUCTURE
Marionette
geckodriver
web-platform-tests documentation
Fuzzing
Sanitizer
Performance Testing
mozperftest
Code coverage
Testing & Debugging Rust Code

Code quality

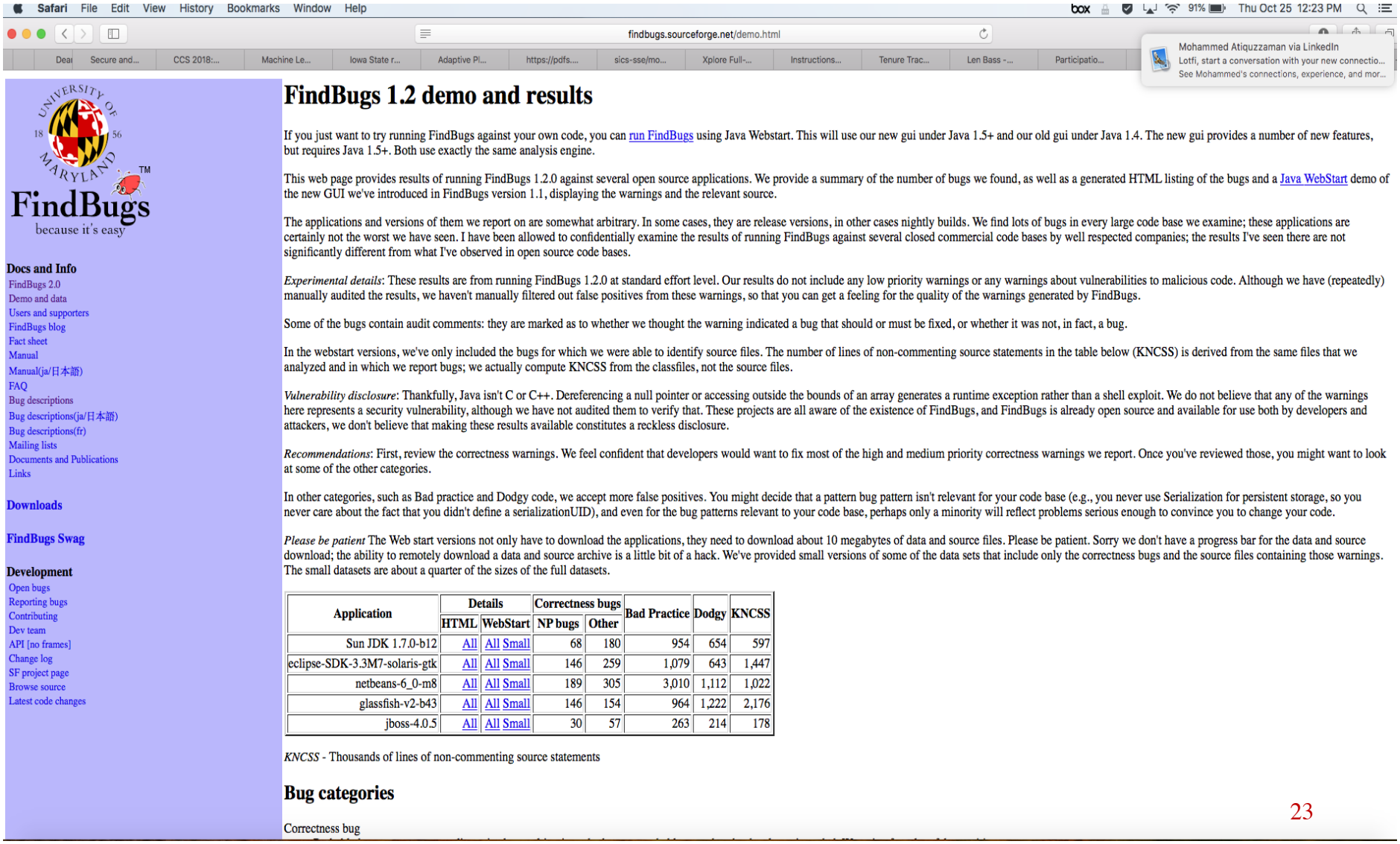
Because Firefox is a complex piece of software, a lot of tools are executed to identify issues at development phase. In this document, we try to list these all tools.

- Static analysis
- Linting
- Coding style

C/C++

Tools	Has autofixes	Meta bug	More info
Custom clang checker			Source
Clang-Tidy	Yes	bug 712350	Static analysis
Clang analyzer		bug 712350	
Coverity		bug 1230156	
cpp virtual final			cpp virtual final
Semmle/LGTM		bug 1458117	
clang-format	Yes	bug 1188202	Formatting C++ Code With cl

Identify Bad Code Automatically



Safari File Edit View History Bookmarks Window Help — Thu Oct 25 12:23 PM

findbugs.sourceforge.net/demo.html

Mohammed Atiquzzaman via LinkedIn
Lotfi, start a conversation with your new connectio...
See Mohammed's connections, experience, and mor...

FindBugs
because it's easy

Docs and Info
- FindBugs 2.0
- Demo and data
- Users and supporters
- FindBugs blog
- Fact sheet
- Manual
- Manual(ja/日本語)
- FAQ
- Bug descriptions
- Bug descriptions(ja/日本語)
- Bug descriptions(fr)
- Mailing lists
- Documents and Publications
- Links

Downloads

FindBugs Swag

Development
- Open bugs
- Reporting bugs
- Contributing
- Dev team
- API [no frames]
- Change log
- SF project page
- Browse source
- Latest code changes

FindBugs 1.2 demo and results

If you just want to try running FindBugs against your own code, you can run FindBugs using Java Webstart. This will use our new gui under Java 1.5+ and our old gui under Java 1.4. The new gui provides a number of new features, but requires Java 1.5+. Both use exactly the same analysis engine.

This web page provides results of running FindBugs 1.2.0 against several open source applications. We provide a summary of the number of bugs we found, as well as a generated HTML listing of the bugs and a Java WebStart demo of the new GUI we've introduced in FindBugs version 1.1, displaying the warnings and the relevant source.

The applications and versions of them we report on are somewhat arbitrary. In some cases, they are release versions, in other cases nightly builds. We find lots of bugs in every large code base we examine; these applications are certainly not the worst we have seen. I have been allowed to confidentially examine the results of running FindBugs against several closed commercial code bases by well respected companies; the results I've seen there are not significantly different from what I've observed in open source code bases.

Experimental details: These results are from running FindBugs 1.2.0 at standard effort level. Our results do not include any low priority warnings or any warnings about vulnerabilities to malicious code. Although we have (repeatedly) manually audited the results, we haven't manually filtered out false positives from these warnings, so that you can get a feeling for the quality of the warnings generated by FindBugs.

Some of the bugs contain audit comments: they are marked as to whether we thought the warning indicated a bug that should or must be fixed, or whether it was not, in fact, a bug.

In the webstart versions, we've only included the bugs for which we were able to identify source files. The number of lines of non-commenting source statements in the table below (KNCSS) is derived from the same files that we analyzed and in which we report bugs; we actually compute KNCSS from the classfiles, not the source files.

Vulnerability disclosure: Thankfully, Java isn't C or C++. Dereferencing a null pointer or accessing outside the bounds of an array generates a runtime exception rather than a shell exploit. We do not believe that any of the warnings here represents a security vulnerability, although we have not audited them to verify that. These projects are all aware of the existence of FindBugs, and FindBugs is already open source and available for use both by developers and attackers, we don't believe that making these results available constitutes a reckless disclosure.

Recommendations: First, review the correctness warnings. We feel confident that developers would want to fix most of the high and medium priority correctness warnings we report. Once you've reviewed those, you might want to look at some of the other categories.

In other categories, such as Bad practice and Dodgy code, we accept more false positives. You might decide that a pattern bug pattern isn't relevant for your code base (e.g., you never use Serialization for persistent storage, so you never care about the fact that you didn't define a serializationUID), and even for the bug patterns relevant to your code base, perhaps only a minority will reflect problems serious enough to convince you to change your code.

Please be patient The Web start versions not only have to download the applications, they need to download about 10 megabytes of data and source files. Please be patient. Sorry we don't have a progress bar for the data and source download; the ability to remotely download a data and source archive is a little bit of a hack. We've provided small versions of some of the data sets that include only the correctness bugs and the source files containing those warnings. The small datasets are about a quarter of the sizes of the full datasets.

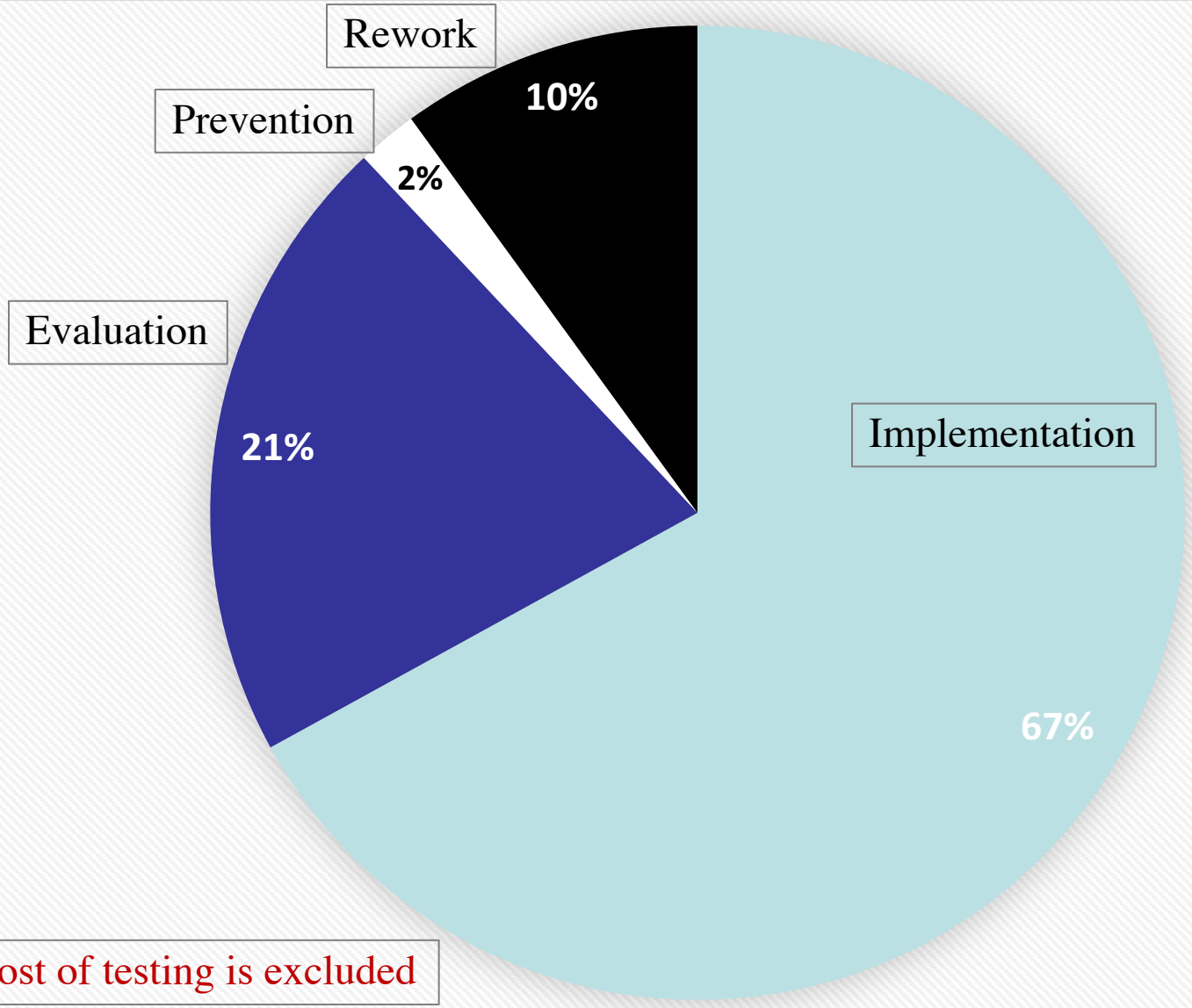
Application	Details		Correctness bugs		Bad Practice	Dodgy	KNCSS
	HTML	WebStart	NP bugs	Other			
Sun JDK 1.7.0-b12	All	All Small	68	180	954	654	597
eclipse-SDK-3.3M7-solaris-gtk	All	All Small	146	259	1,079	643	1,447
netbeans-6_0-m8	All	All Small	189	305	3,010	1,112	1,022
glassfish-v2-b43	All	All Small	146	154	964	1,222	2,176
jboss-4.0.5	All	All Small	30	57	263	214	178

KNCSS - Thousands of lines of non-commenting source statements

Bug categories

Correctness bug

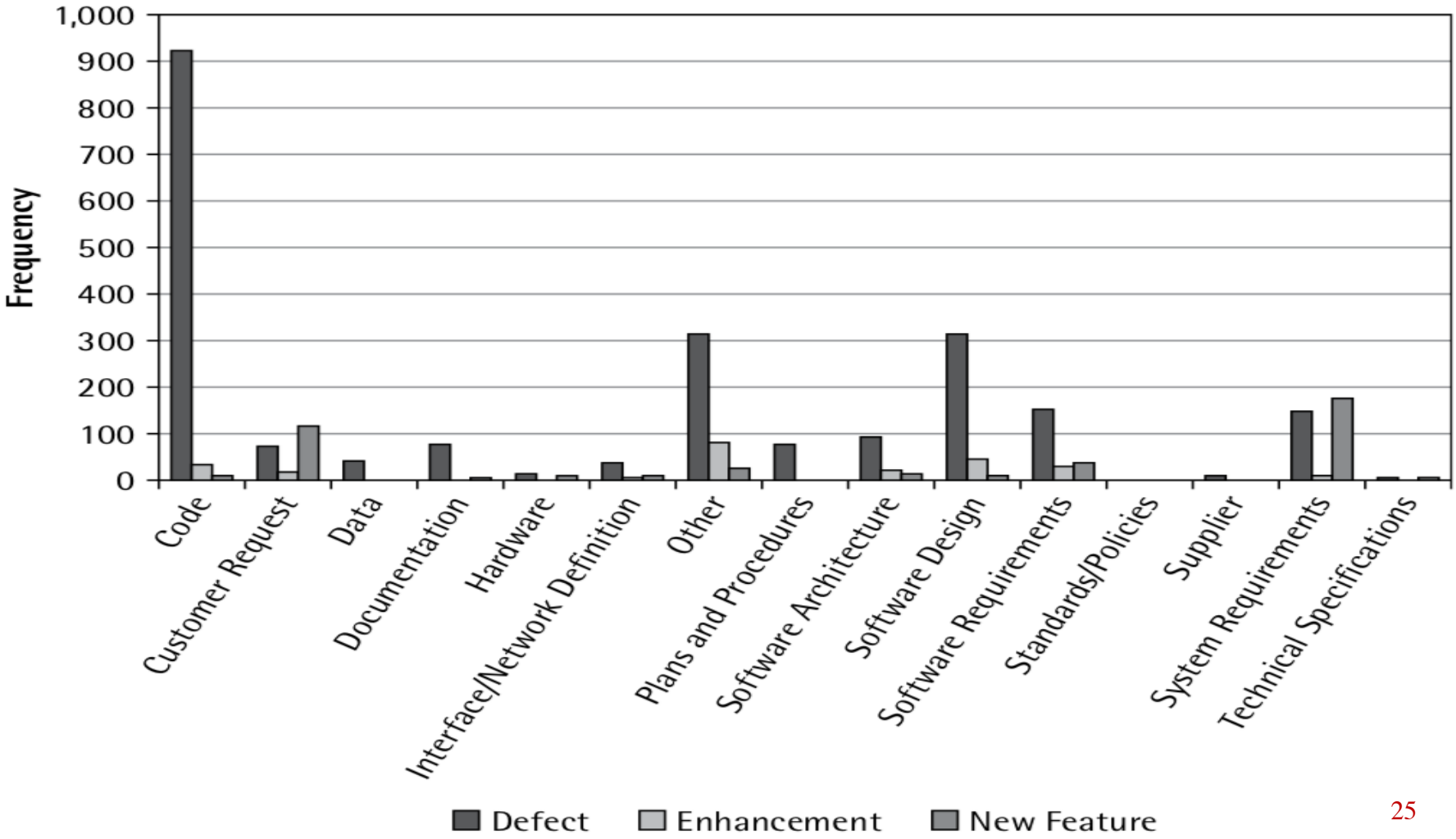
Distribution of Efforts
Case of Bombardier - 88,000 hours project



NB: Cost of testing is excluded

Distribution of Effort by Type of Maintenance

What are the top 3 activities for addressing defects?



We know:

- What is quality?
- Three types of quality aspects
- Quality metrics
- The quality control activities
- The importance and cost of quality

- How to plan for it? How to control it?

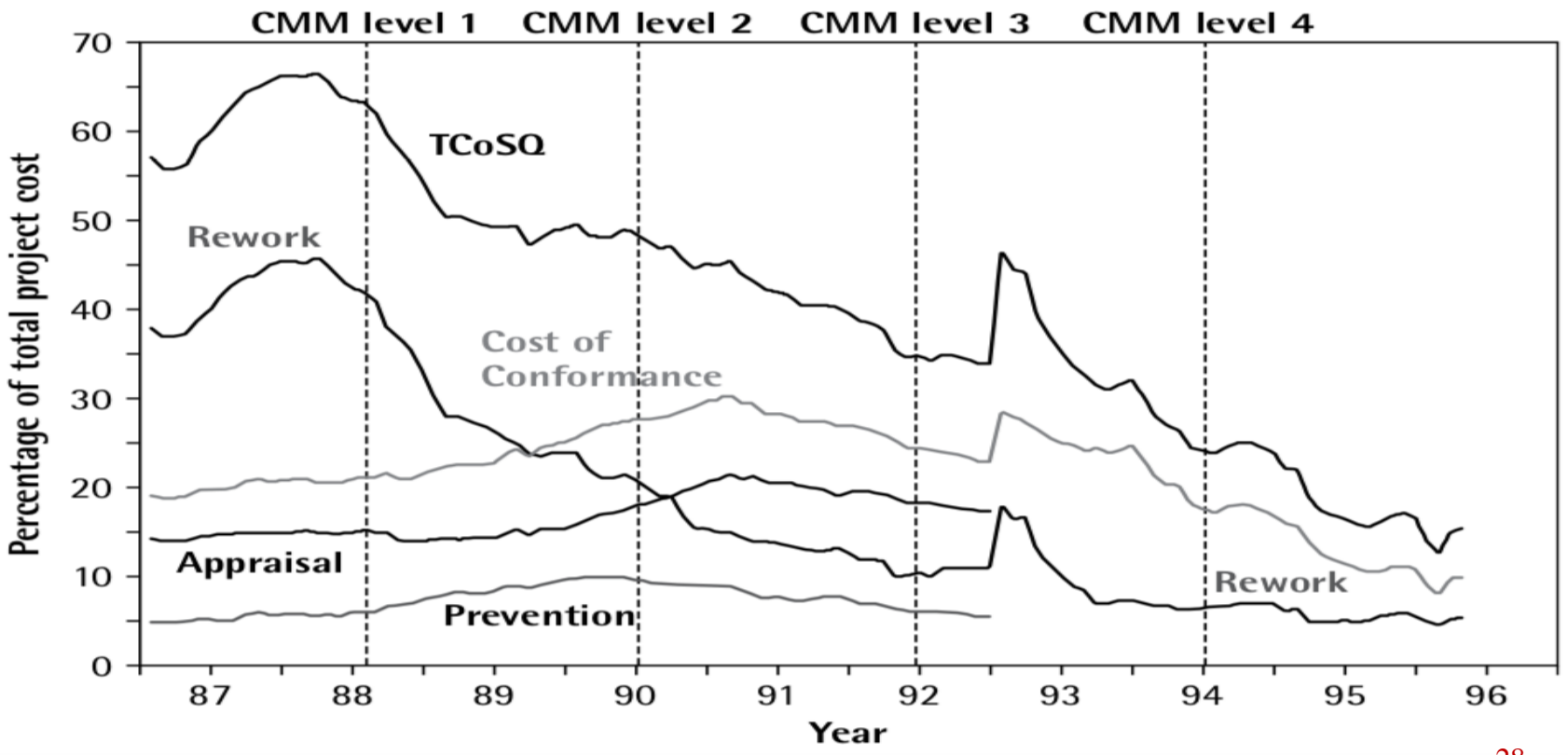
Is There a Relationship Between CMM Levels and Quality Cost?

1. Initial - The software process is characterized as ad hoc
2. Repeatable - Basic project management processes are established to track cost, schedule, and functionality
3. Defined - The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process
4. Managed - Detailed measures of the software process and product quality are collected
5. Optimizing - Continuous process improvement is enabled

Cost of Quality at Raytheon - 1996

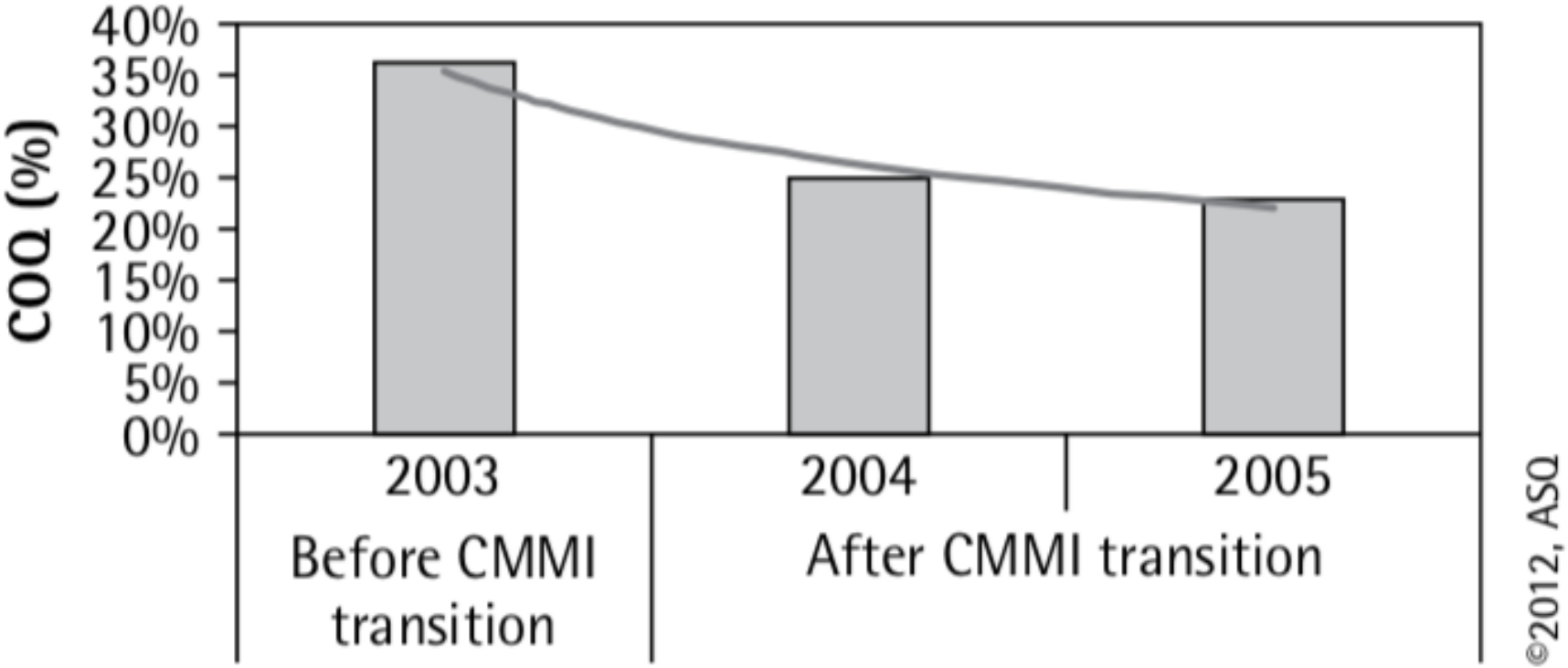
What is the relationship between the cost of quality and CMM for RAYTHEON? And Why?

FIGURE 5 CoSQ data at Raytheon (adapted from Haley 1996)



Cost of Quality at Motorola

Is this relationship specific for one company? And Why?



Control Cost of Quality

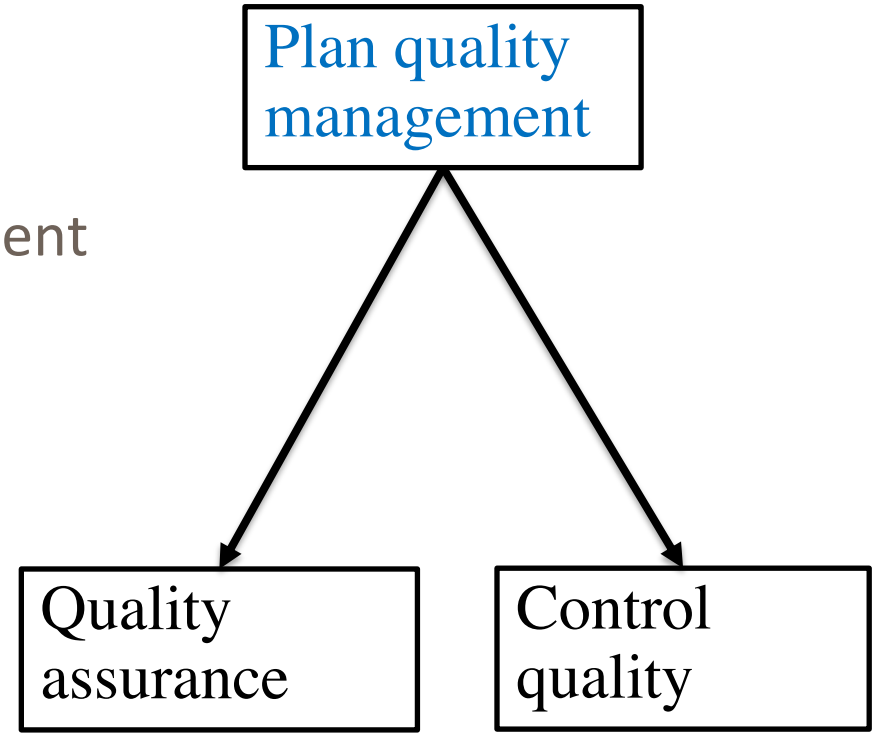
- Cost of software quality is correlated with maturity of processes.
- Reducing the cost implies improves the maturity of the organization.

Again - Quality Control Activities

1. Process improvements
2. Requirement traceability
3. Prototyping, test, debug
4. Design review
5. Testing activities
6. Software validation and verification
7. Maintenance rework
8. Release preparation
9. Software problem correction
10. Code review
11. Etc.

Project Quality Management

- Plan quality management
- Control quality
- Quality assurance



Quality Management Plan

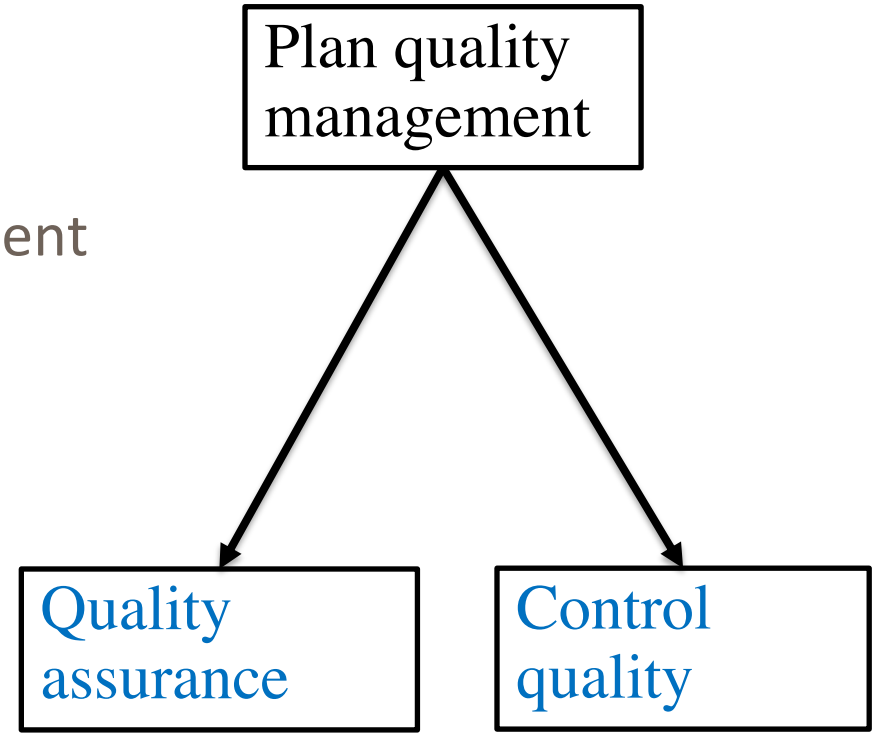
- Plan quality management – Identify quality requirements and how to demonstrate compliance
- Project quality program – Action plan such that the quality of the software meets the planned one

Techniques for Planning Quality Management

- Cost-benefit analysis
- Benchmarking
- Simulate changes
- Brainstorming

Project Quality Management

- Plan quality management
- Control quality
- Quality assurance



Quality Assurance

- Quality assurance is: audit the quality requirements and results to ensure appropriate standards are used.

- Objectives of quality audit:
  - Identify good practices
  - Identify non-conformity gaps and shortcomings
  - Learn lessons from the audit
- Process analysis aims to identify needed process improvements to address existing problems.

Control Quality

- Control quality – monitor and record results of executing quality activities.

- Control quality uses a set of techniques and tasks to verify that the deliverables meet the expectation of the customers

- Control quality techniques:
  1. Seven basic tools: affinity diagram, cause-effect diagram, etc.
  2. Statistical sampling
  3. Inspection
  4. Review approved change requests

Self-check

- What is quality?
- What are the components of quality cost?
- What is the relationship between CMM and quality cost?
- What is quality management program?
- Why quality management control is important?

Please post your questions on Piazza.