Introduction to Software Engineering

(CS350)

Lecture 14

Jongmoon Baik

Software Quality Assurance

Comment on Quality

- Phil Crosby once said:
  - "The problem of quality management is not what people don't know about it. The problem is what they think they do know . . . In this regard, quality has much in common with sex."
  - *Everybody is for it.* (Under certain conditions, of course.)
  - *Everyone feels they understand it.* (Even though they wouldn't want to explain it.)
  - *Everyone thinks execution is only a matter of following natural inclinations.* (After all, we do get along somehow.)
  - *And, of course, most people feel that problems in these areas are caused by other people.* (If only they would take the time to do things right.)

Four Variables of SW Dev.

- Scope
- Time
- Resources (or cost)
- Quality

Who defines Quality?

Is quality a viable variable?

Can Quality be "bought"?

Two critical Parts for QA

- The Product delivered –human injection of defects
  - Bug / *defectfree*
  - Answers the problem –Requirements
  - Activities: Testing, reviews, inspections

- The Process that builds the product
    - –to limit human variation
  - How to consistently make a high quality product
  - Obviously closely tied to the product
  - Activities: Audits, inspections
  - CMM, standards, formalized

Defect Defined

Pressman (SEPA ’01)

“A defect occurs when quality assurance activities (e.g. formal technical reviews) fail to uncover an error in a ***work product*** produced during the software ***process***”

“A quality problem found only after release to the end user.”

Error defined Pressman 2005

"Quality problem found before release to the end user."

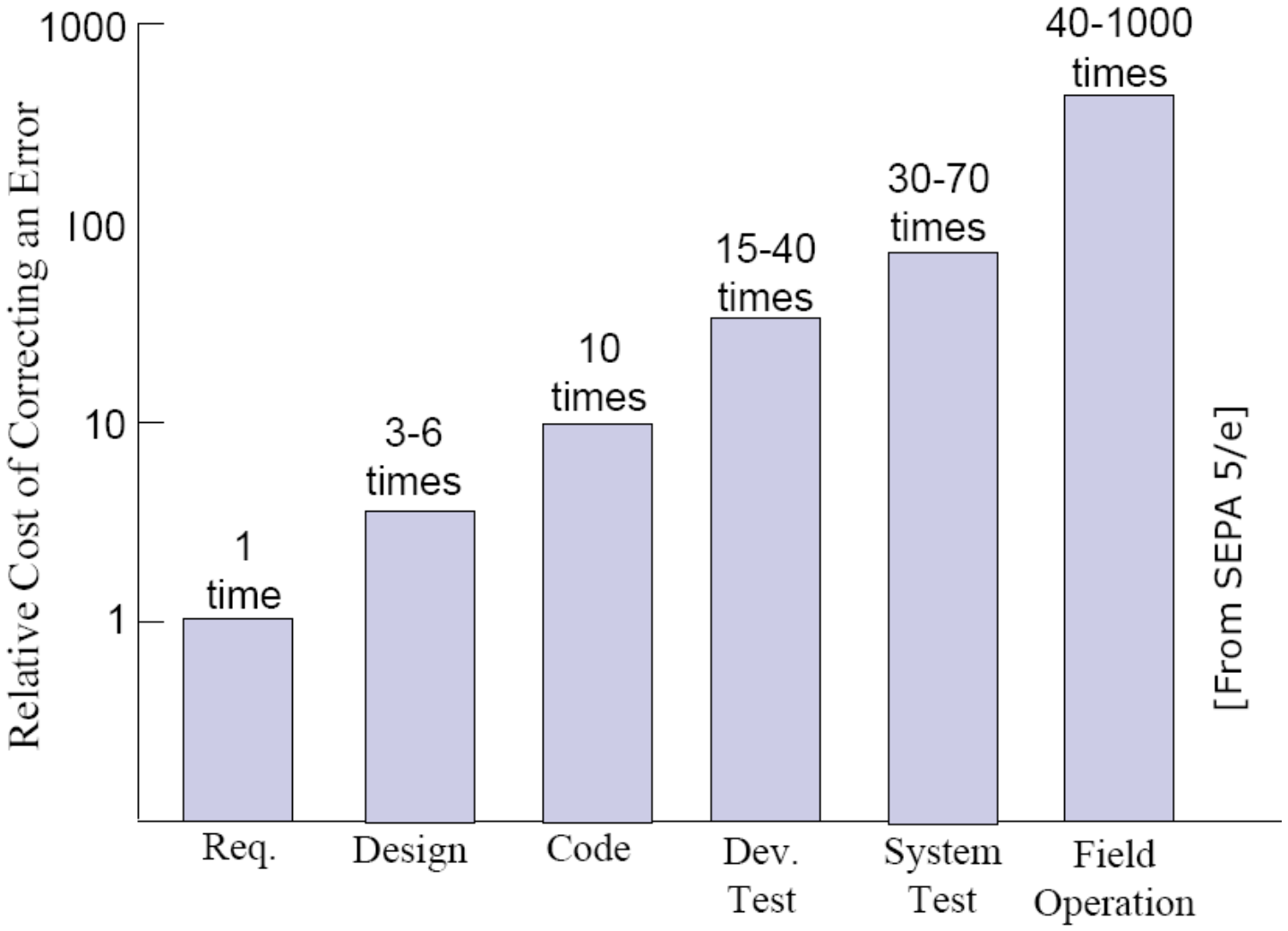
Or are errors the same as defects and bugs?

Define, define, define -for all stakeholders

Problems with Selling QA (Justifying Cost)

Up front cost with no perceived return

- Resource Use –How much do I divert
  - Hard to sell
- Return on Investment
  - Hard sell of up front costs
- Planning, reviews, tools, training
- Appraisal cost –setting baseline
- Setting standards to measure from


Relative Cost of Correcting an Error
1000
100
10
1
1 time
3-6 times
10 times
15-40 times
30-70 times
40-1000 times
[From SEPA 5/e]
Req.
Design
Code
Dev. Test
System Test
Field Operation

Simple Answer

1. The earlier in the process that defects are discovered and removed, the cheaper the development costs!

2. Inspections/reviews work

Elements of SQA

- Standards
- Reviews and Audits
- Testing
- Error/defect collection and analysis
- Change management
- Education
- Vendor management
- Security management
- Safety
- Risk management

QA & Standards

Ian Sommerville 2000

- Key to effective quality management
- International, national, organizational and local
- Define characteristics that all components should exhibit
  - e.g. a common programming style
- Process standards define how the software process should be enacted

Importance of standards

Ian Sommerville 2000

- Best practices-avoids repetition of past mistakes
- Framework for quality assurance process -it involves checking standard compliance
- Provide continuity -understand the organization by the standards applied
- Reduce costs –reinventing the wheel
  - Easier to edit than create

Problems with standards

Ian Sommerville 2000

- Not seen as relevant and up-to-date by software engineers
  - Review and fix
- Involve too much bureaucratic form filling
- Unsupported by software tools so tedious manual work is involved to maintain standards
- Setting standards for “subjective” items
  - QA activities
  - Forming teams…
  - Others?

Standards development

Ian Sommerville 2000

- Involve developers
  - Understand the rationale underlying a standard
- Review standards and their usage **regularly**
  - Standards can quickly become outdated and this reduces their credibility amongst practitioners
  - Set review dates: Annual, bi-annual, etc.
- Detailed standards should have associated tool support
  - Excessive clerical work ?
- Don’t reinvent the wheel: ISO, IEEE….
  - Benchmark –Easier to edit than create, but do CHECK them

Name some standards…

(you may use or want)

First look back on when you did have to reinvent the wheel

Role of the SQA Group-I

- **Prepares an SQA plan for a project.**
  - The plan identifies
    - evaluations to be performed
    - audits and reviews to be performed
    - standards that are applicable to the project
    - procedures for error reporting and tracking
    - documents to be produced by the SQA group
    - amount of feedback provided to the software project team
- **Participates in the development of the project's software process description.**
  - The SQA group reviews the process description for compliance with organizational policy, internal software standards, externally imposed standards (e.g., ISO-9001), and other parts of the software project plan.

Role of the SQA Group-II

- **Reviews software engineering activities to verify compliance with the defined software process.**
  - identifies, documents, and tracks deviations from the process and verifies that corrections have been made.
- **Audits designated software work products to verify compliance with those defined as part of the software process.**
  - reviews selected work products; identifies, documents, and tracks deviations; verifies that corrections have been made
  - periodically reports the results of its work to the project manager.
- **Ensures that deviations in software work and work products are documented and handled according to a documented procedure.**
- **Records any noncompliance and reports to senior management.**
  - Noncompliance items are tracked until they are resolved.

QA group

- Separate – reporting chain
- Communication with developers
- Acceptance as effective
- Fix the problem not the blame
  - Call use “assistance”
- Expertise
  - Moderators for inspections
  - Assistance with plans, how to check them

Generic QA Plan

- Evaluation –e.g. QA of the QA plan
- Audits / Reviews (code and plans?)
- Standards
- Metrics defined
- Procedures for Error Reporting
- Analysis of Data
- Artifacts / Documents to be produced
- Feedback to Developers
- Example: IEEE 730TM-**2002**

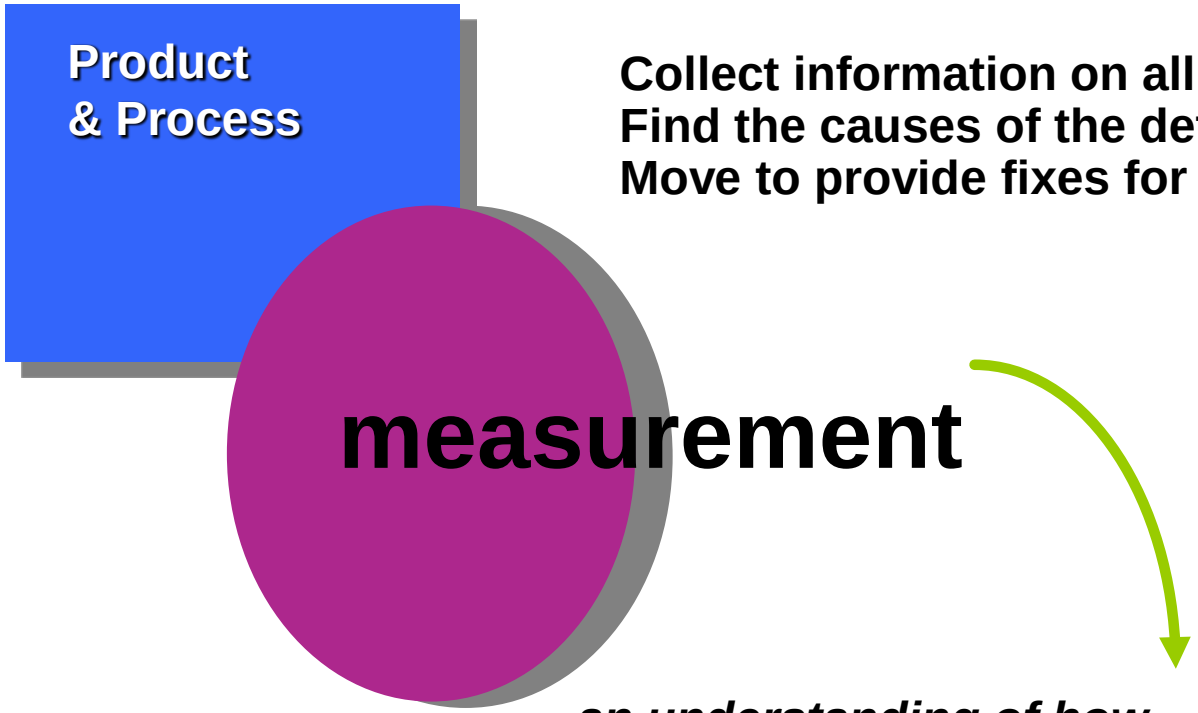
QA plan Recommendations

- Involve all stakeholders
  - Avoid the QA group “enemy”syndrome
- Benchmark
- Don’t reinvent the wheel
- Standards
- Plan for review of the plan

SQA Goals (see Figure 16.1)

- **Requirements quality.** The correctness, completeness, and consistency of the requirements model will have a strong influence on the quality of all work products that follow.
- **Design quality.** Every element of the design model should be assessed by the software team to ensure that it exhibits high quality and that the design itself conforms to requirements.
- **Code quality.** Source code and related work products (e.g., other descriptive information) must conform to local coding standards and exhibit characteristics that will facilitate maintainability.
- **Quality control effectiveness.** A software team should apply limited resources in a way that has the highest likelihood of achieving a high quality result.

Statistical SQA



Collect information on all defects
Find the causes of the defects
Move to provide fixes for the process

... an understanding of how

to improve quality ...

Statistical SQA Steps

- Information about software errors and defects is collected and categorized.
- An attempt is made to trace each error and defect to its underlying cause (e.g., non-conformance to specifications, design error, violation of standards, poor communication with the customer).
- Using the Pareto principle (80 percent of the defects can be traced to 20 percent of all possible causes), isolate the 20 percent (the *vital few*).
- Once the vital few causes have been identified, move to correct the problems that have caused the errors and defects.

Six-Sigma for Software Engineering

- The term “six sigma” is derived from six standard deviations—3.4 instances (defects) per million occurrences—implying an extremely high quality standard.
- The Six Sigma methodology defines three core steps:
  - *Define* customer requirements and deliverables and project goals via well-defined methods of customer communication
  - *Measure* the existing process and its output to determine current quality performance (collect defect metrics)
  - *Analyze* defect metrics and determine the vital few causes.
  - *Improve* the process by eliminating the root causes of defects.
  - *Control* the process to ensure that future work does not reintroduce the causes of defects.

Software Reliability

- A simple measure of reliability is *mean-time-between-failure* (MTBF), where

$$MTBF = MTTF + MTTR$$

- The acronyms MTTF and MTTR are *mean-time-to-failure* and *mean-time-to-repair*, respectively.
- *Software availability* is the probability that a program is operating according to requirements at a given point in time and is defined as

$$Availability = [MTTF/(MTTF + MTTR)] \times 100\%$$

Software Safety

- *Software safety* is a software quality assurance activity that focuses on the identification and assessment of potential hazards that may affect software negatively and cause an entire system to fail.
- If hazards can be identified early in the software process, software design features can be specified that will either eliminate or control potential hazards.

ISO 9001:2000 Standard

- ISO 9001:2000 is the quality assurance standard that applies to software engineering.
- The standard contains 20 requirements that must be present for an effective quality assurance system.
- The requirements delineated by ISO 9001:2000 address topics such as
  - management responsibility, quality system, contract review, design control, document and data control, product identification and traceability, process control, inspection and testing, corrective and preventive action, control of quality records, internal quality audits, training, servicing, and statistical techniques.

Q & A