Quality Assurance

José Costa

Software for Embedded Systems

Departamento de Engenharia Informática (DEI)
Instituto Superior Técnico

2015-10-03

Outline

- Quality assurance
- Quality assurance techniques
- Verifying the specification
- Design reviews
- Measurement-driven quality assurance

Quality of a Product

- Can be judged by how well it satisfies its intended function

Reasons for low quality

- shoddily manufactured
- components improperly designed
- architecture poorly conceived
- product's requirements poorly understood

- You can't test out enough bugs to deliver a high-quality product!

Quality Assurance

- Quality judged by how well product satisfies its intended function
  - may be measured in different ways for different kinds of products

- Quality assurance (QA) makes sure that all stages of the design process help to deliver a quality product

Case Study: Therac-25 Medical Imager

- Six known accidents: radiation overdoses leading to death and serious injury
- Radiation gun controlled by PDP-11 minicomputer

Findings

- changes to parameters made by operator may show on screen but not be sensed by data entry task
- one accident caused by entering mode/energy, changing mode/energy, returning to command line in 8 seconds
- skilled operators typed faster, more likely to exercise bug

Case Study: Therac-25 Medical Imager

TÉCNICO LISBOA

Observations

- Performed limited safety analysis
  - guessed at error probabilities, etc.
- Did not use mechanical backups to check machine operation
  - logs
  - cross-checks of the states of the external sensors and actuators
- Used overly complex programs written in unreliable styles

Quality Metrics

- Metrics are important to the quality control process

- How to know whether we have achieved high levels of quality?
  - must be able to measure aspects of the system
    - e.g., execution speed, coverage of test patterns
  - must be able to measure aspects of the design process
    - e.g., rate at which bugs are found

ISO 9000 (1/2)

- Developed by International Standards Organization
- Applies to a broad range of industries
- Concentrates on process
  - ability of the processes defined to evaluate "product quality" or "user satisfaction"
  - adherence of the processes to the quality manual
- Validation based on extensive documentation of organization's process

ISO 9000 (2/2)

- Process is crucial
  - haphazard development leads to haphazard products and low quality
- Documentation is important
  - helps internal quality monitoring groups to ensure that the required processes are being followed
  - helps outside groups understand the processes
- Communication is important
  - people in the organization should understand not only their specific tasks but also how their jobs affect overall quality

CMU Capability Maturity Model

- Developed by Carnegie Mellon University's Software Engineering Institute
- Way of measuring the quality of the software development process
- Provides a model for judging an organization

CMM Levels of Organization Maturity

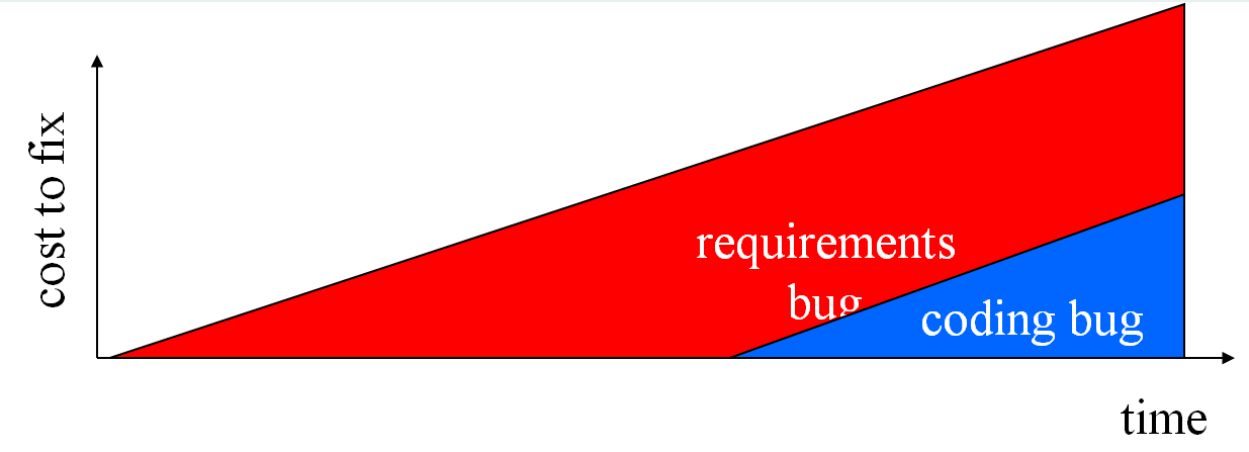
1. Initial
   - poorly organized process
   - success depends on individuals
2. Repeatable
   - basic tracking mechanisms to understand cost, scheduling and how the system meets goals
3. Defined
   - processes documented and standardized
4. Managed
   - makes detailed measurements of process and product quality
5. Optimizing
   - measurements used for improvement

Very few organizations meet the highest level!

Verifying the Specification

- Requirements and specification are generated very early in the design process
- Bugs introduced in these phases are extremely expensive to fix later on
- The longer a bug survives in the system, the more expensive it will be to fix it

Early bugs are more expensive to fix



Verifying the Specification

- The goal of validating the requirements and specification is to ensure that they are:
  - correct
  - unambiguous
  - complete
  - verifiable
  - consistent
  - modifiable
  - traceable

Verifying the requirements and specification is very important!

Requirements Validation

- Requirements validation may seem like a challenge
  - they come somewhat informal from the customer

- But many things can be done

How to validate the requirements

- Prototypes
- Prototyping languages
- Pre-existing systems

Prototypes

- Very useful tool
- Can let you see, hear and touch
- Not fully functional
  - design work has not been done
- User interfaces are well suited to prototyping
  - allowing user testing
- Can help the end user critique numerous functional and nonfunctional requirements:
  - e.g., data displays, speed of operation, size, weight

Prototyping/Specification laguages

- Certain programming languages are well suited to prototyping
  - e.g, matlab
- May be able to perform functional atributes
- Usualy not good in doing nonfunctional atributes

Preexisting Systems

- Helps the user articulate the needs
- Much easier to know if someone likes or not some design
- It may be possible to construct a prototype of the new system using the preexisting

Specification Validation

- Could use the same techniques in the requirements validation
  - prototypes
  - prototyping languages
  - pre-existing systems

- Other tools
  - usage scenarios
  - formal techniques

Usage Scenarios

- Work through usage scenarios
- Helps designers fill out details
- Ensures completeness
- Ensures correcteness

Formal Techniques

Design techniques that make use of mathematical proofs

- Proofs may be done either manually or automatically
- Proving that a particular condition can or can not occur according to the specification is important
- Automated proofs are particular useful
  - in certain types of complex systems

Design Review

Design review

A meeting in which team members discuss a design, reviewing how a component of the system works

- Critical component of any quality assurance process
- Simple, low-cost way of catching bugs early in the process
- Uses meetings to catch design flaws
- Proven by experiments to be effective
- Use other people in the project/company to help spot design problems

Design Review - Players

- Designers
  - central in this process
  - present design to rest of team, make changes
- Review leader
  - coordinates process
- Review scribe
  - takes notes of meetings
- Review audience
  - studies the component
  - looks for bugs
  - includes other members of the project for which this component is being designed

Design Review - Before

The design review begins before the meeting itself.

- Design team
  - prepares documents used to describe the design
  - distributes them
- Leader
  - recruits audience, coordinates meetings, distributes handouts, etc.
- Audience members
  - familiarize themselves with the documents before they go to the meeting

Design Review - Meeting

- Leader keeps meeting moving
- Scribe takes notes
- Designers present the design
  - use handouts
  - explain what is going on
  - go through details

Design Review - Audience

Audience looks for any problems.

- Is the design consistent with the specification?
- Is the interface correct?
- How well is the component's internal architecture designed?
- Did they use good design/coding practices?
- Is the testing strategy adequate?

Design Review - Follow-up

- Uses notes by scribe
- Designers make suggested changes
  - document changes
- Leader checks on results of changes
  - may distribute to audience for further review or additional reviews

Measurement-Driven Quality Assurance

TÉCNICO LISBOA

- Design review is an inspection-based system
  - it cannot be fully automated
  - no formal criteria for finding all types of bugs
- Inspection process is important
  - there will always be creative ways to introduce bugs that require human analysis
- But we can automate the process of finding bugs
  - measuring bug rates
  - use those statistics

Measurements

- Measurements help ground our beliefs:
  - Do our practices really work?
  - Do they work where we think they work?

Types of measurements

- bugs found at different stages of design
- bugs as a function of time
- bugs in different types of components
- how bugs are found

Outline

- Quality assurance
- Quality assurance techniques
- Verifying the specification
- Design reviews
- Measurement-driven quality assurance

References

- Computers as Components: Principles of Embedded Computing System Design. Marylin Wolf. Morgan Kaufman. Ch 9.6

Next Class

- Smart Cards