CITS5501 Software Testing and Quality Assurance
Software reviews

Unit coordinator: Arran Stewart

Reviews

Software reviews

- A catch all term for *filters* that can be applied at various points during the software development process

Static Testing Methods

- Code Inspections
- Code Walk-throughs
- Semantic Analysis of Code
- Fagan Inspections
- Analysis of Code Metrics

Code Inspection Methods

- Inspections are a formal, efficient and economical method of finding faults in design and code [Fagan,76]
- Code inspection amounts to "executing the code in your head" or on paper
- Code inspections are very effective at finding faults [statistics from Pfleeger p.291]
    - detect 67% of faults [Fagan 76]
    - detect 85% of faults [Jones 77]
    - detect 93% of errors [Ackerman et al 86]

Code Walk-through

- A Code Walk-through is an informal meeting where the programmer leads the review team through his/her code and the reviewers try to identify faults
- You can do this with your own code but it is more effective if the reviewer is not the same person as the programmer. See egoless programming in many SE texts
- Example: perform a code walk through of the getNumofDaysinMonth method

getNumOfDays

```
if (year<1) {
  throw new YearOutOfBounds(year);
}

if (month==1 || month==3 || month==5 || month=7 || month==10
    || month==12) {
  numDays = 32;
} else if (month==4 || month==6 || month==9 || month==11) {
  numDays = 30;
} else if (month==2) {
  if (isLeapYear(year)) { numDays = 29;
  }
  else { numDays = 28;
  }
} else {
  throw new MonthOutOfBounds(month);
}
return numDays;
```

Fagan Inspection

- is a more formal version of a code walk-through

- Procedure:

    1. Overview
    2. Preparation
    3. Inspection
    4. Rework
    5. Follow up

- meetings are chaired by a team moderator rather than the programmer

Checklists

- A set of questions to stimulate critical appraisal of all aspects of the system
- Questions are usually general in nature and thus applicable to many types of system
- Checklist examples will go up on website

Semantic Analysis

Analysis based on a model of the meaning (i.e. semantics) of a program

- Formal Proofs: prove a given program (model) satisfies a required property
- Control Flow Analysis: analysis of the directed graph of the control structure of a program to identify inaccessible code, infinite loops and poor structure
- Data Flow Analysis: analysis of a diagrammatic representation of the flow of data throughout a program
- Symbolic Execution: check for agreement between code and specification using algebraic vars in place of input data. Assignment statements produce algebraic output which can be compared with expected results.

Code Metrics

- Measures of properties of code which may predict how likely the code is to contain errors
- For example,
    - graph theoretic complexity (of the program's control graph)
    - module accessibility (how many ways a module may be accessed)
    - number of entry and exit points per module
    - for some Object Oriented Metrics see
      http://yunus.hun.edu.tr/~sencer/oom.html

Quality assurance

Overview

- Source: Pressman, R. Software Engineering: A Practitioner's
  Approach (McGraw-Hill, 2005)
- Aspects of quality
  - Definition, types of quality costs
- Organizational responsibility
  - Who is responsible for software quality?
- Software reliability, availability, safety
- Quality assurance plans
- Techniques
  - Software reviews

What is Quality Management

- Also called software quality assurance (SQA)
- Serves as an umbrella activity that is applied throughout the software process
- Involves doing the software development correctly versus doing it over again
- Reduces the amount of rework, which results in lower costs and improved time to market

What is Quality Management (cont'd)

SQA encompasses:

- A software quality assurance *process*
    - (i.e. What is our process for ensuring we maintain quality?)
- Specific quality assurance and quality control *tasks* (including formal technical reviews and a multi-tiered testing strategy)
    - (i.e. What tasks are involved?)
- Effective software engineering practices (methods and tools)
    - (e.g. Do we use revision control? Unit testing frameworks? OO analysis and design?)
- Control of all software work products and the changes made to them
    - (Do we know what and where our software artifacts are, and who can change them, and when and why?)
- A procedure to ensure compliance with software development standards
    - (e.g. What checks do we have in place?)
- Measurement and reporting mechanisms
    - (e.g. How do team leaders/management know how we're doing?)

Quality Defined

- "*a* quality" is defined as a characteristic or attribute of something
- Refers to measurable characteristics that we can compare to known standards
- "quality" (in general) is "the degree of excellence of something"
- Software is more difficult in some ways to measure than other things
- But there are still many attributes which we can measure

Software quality – a definition

Definition: "Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software"

Software quality – a definition (cont'd)

- This definition emphasizes three points
    - Software requirements are the foundation from which quality is measured; lack of conformance to requirements is lack of quality
    - Specified standards define a set of development criteria that guide the manner in which software is engineered; if the criteria are not followed, lack of quality will almost surely result
    - A set of implicit requirements often goes unmentioned; if software fails to meet implicit requirements, software quality is suspect
- Software quality is no longer the sole responsibility of the programmer
    - It extends to software engineers, project managers, customers, salespeople, and the SQA [Software Quality Assurance] group
    - Software engineers apply solid technical methods and measures, conduct formal technical reviews, and perform well-planned software testing

Software quality – a definition (cont'd)

- query: What if there are no (or few) formal requirements?
- query: What if it meets the requirements, but customers are unhappy with it?

Quality Defined (continued)

Some sub-types of quality, taken from manufacturing:

- Quality of design (the quality we're *intending* to offer)
  - The characteristic that designers specify for an item
  - This encompasses requirements, specifications, and the design of the system
- Quality of conformance (i.e., implementation)
  - The degree to which the design specifications are followed during manufacturing
  - This focuses on how well the implementation follows the design and how well the resulting system meets its requirements

Quality – a user perspective

- Robert Glass's formulation:

```
user satisfaction = compliant product +
                    good quality +
                    delivery on time & within budget
```

Quality – a user perspective (cont'd)

- What if the user wanted something that's inadvisable? Or illegal?

Quality – a user perspective (cont'd)

- What if the user wanted something that's inadvisable? Or illegal?
    - "Store all user-names and passwords in a text file, so we can easily manage and change them."

Quality Control

- Involves a series of inspections, reviews, and tests used throughout the software process
- Ensures that each work product meets the requirements placed on it
- Includes a feedback loop to the process that created the work product
  - This is essential in minimizing the errors produced
- Combines measurement and feedback in order to adjust the process when product specifications are not met
- Requires all work products to have defined, measurable specifications to which practitioners may compare to the output of each process

The Cost of Quality

MS Word – it sometimes crashes

- could it be made better quality?

The Cost of Quality

MS Word – it sometimes crashes

- could it be made better quality?
- could Microsoft apply, say, formal methods to get NASA-level quality from MS Word?

The Cost of Quality

MS Word – it sometimes crashes

- could it be made better quality?
- could Microsoft apply, say, formal methods to get NASA-level quality from MS Word?
- should they?

The Cost of Quality

MS Word – it sometimes crashes

- could it be made better quality?
- could Microsoft apply, say, formal methods to get NASA-level quality from MS Word?
- should they?
- What would be the costs? What would be the benefits?

The Cost of Quality

Cost of quality . . .

- includes all costs incurred in the pursuit of quality or in performing quality-related activities
- is studied to
    - Provide a baseline for the current cost of quality
    - Identify opportunities for reducing the cost of quality
    - Provide a normalized basis of comparison (which is usually dollars)
- involves various *kinds* of quality costs (see next slides)
- increases dramatically as the activities progress from
    - Prevention $\Rightarrow$ Detection $\Rightarrow$ Internal failure $\Rightarrow$ External failure

Kinds of Quality Costs

- Prevention costs *(ensure mistakes never creep in)*
    - Quality planning, formal technical reviews, test equipment, training
- Appraisal costs *(check whether they have)*
    - Inspections, equipment calibration and maintenance, testing
- Failure costs *(oops)*
    - subdivided into internal failure costs and external failure costs
    - Internal failure costs
        - Incurred when an error is detected in a product prior to shipment
        - Include rework, repair, and failure mode analysis
    - External failure costs
        - Involves defects found after the product has been shipped
        - Include complaint resolution, product return and replacement, help line support, and warranty work

The SQA (Software Quality Assurance) Group

In an organisation with end-users/customers:

- Serves as the customer's in-house representative
- Assists the software team in achieving a high-quality product
- Views the software from the customer's point of view
  - Does the software adequately meet quality factors?
  - Has software development been conducted according to pre-established standards?
  - Have technical disciplines properly performed their roles as part of the SQA activity?
- Performs a set of of activities that address quality assurance planning, oversight, record keeping, analysis, and reporting (See next slide)

SQA Activities

- Prepares an SQA plan for a project
- Participates in the development of the project's software process description
- Reviews software engineering activities to verify compliance with the defined software process
- Audits designated software work products to verify compliance with those defined as part of the software process
- 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
- Coordinates the control and management of change
- Helps to collect and analyze software metrics