Dependability Properties

Dr. Paul West

Department of Computer Science
College of Charleston

January 16, 2014
Analysis

- analysis includes
  - manual inspection techniques
  - automated analyses
- can be applied at any development stage
- particularly well suited at the early stages of specifications and design (when there is little code)
- works well later as well: code reviews
Inspection

- can be applied to essentially any document
  - requirements statements
  - architectural and detailed design documents
  - test plans and test cases
  - program source code
- may also have secondary benefits
  - spreading good practices
  - instilling shared standards of quality.
- takes a considerable amount of time
- re-inspecting a changed component can be expensive
- used primarily
  - where other techniques are inapplicable
  - where other techniques do not provide sufficient coverage
Automatic Static Analysis

- More limited in applicability
  - can be applied to some formal representations of requirements models
  - not to natural language documents
- are selected when available
  - substituting machine cycles for human effort makes them particularly cost-effective.
**Testing**

- Executed late in development
- Start as early as possible
- Early test generation has several advantages
  - Tests generated independently from code, when the specifications are fresh in the mind of analysts
  - The generation of test cases may highlight inconsistencies and incompleteness of the corresponding specifications
  - Tests may be used as compendium of the specifications by the programmers
Improving the Process

- Long lasting errors are common
- It is important to structure the process for
  - Identifying the most critical persistent faults
  - tracking them to frequent errors
  - adjusting the development and quality processes to eliminate errors
- Feedback mechanisms are the main ingredient of the quality process for identifying and removing errors
Organizational Factors

- Different teams for development and quality?
  - separate development and quality teams is common in large organizations
  - indistinguishable roles is postulated by some methodologies (extreme programming)

- Different roles for development and quality?
  - test designer is a specific role in many organizations
  - mobility of people and roles by rotating engineers over development and testing tasks among different projects is a possible option
Allocation Responsibilities Example

Allocating tasks and responsibilities is a complex job: we can allocate
- Unit testing
to the development team (requires detailed knowledge of the code)
but the quality team may control the results (structural coverage)
- Integration, system and acceptance testing
to the quality team
but the development team may produce scaffolding and oracles
- Inspection and walk-through
to mixed teams
- Regression testing
to quality and maintenance teams
- Process improvement related activities
to external specialists interacting with all teams
Rewarding Mechanisms Case A

- allocation of responsibilities
  - Development team responsible for development measured with LOC per person month
  - Quality team responsible for quality
- possible effect
  - Development team tries to maximize productivity, without considering quality
  - Quality team will not have enough resources for bad quality products
- result
  - Product of bad quality and overall project failure
Rewarding Mechanisms Case B

- allocation of responsibilities
  - Development team responsible for both development and quality control

- possible effect
  - the problem of case A is solved
  - but the team may delay testing for development without leaving enough resources for testing

- result
  - delivery of a not fully tested product and overall project failure
Summary

- Test and Analysis are complex activities that must be suitably planned and monitored.

- A good quality process obeys some basic principles:
  - visibility
  - early activities
  - feedback

- aims at
  - reducing occurrences of faults
  - assessing the product dependability before delivery
  - improving the process
Chapter 4 Assignment

Choose and complete any two Chapter 4 Exercises (pg 51-52), except 4.8
Due in the dropbox by January 23, 2014 2359
• Evaluate H/FOSS projects and select 1st, 2nd, 3rd candidates that you wish to join/build a test suite for.

• Create a presentation for next class that summarizes why you chose each one. (EX: You like the goals of the project, you enjoy the technologies employed, ...)

• Criteria: the project must compile and run in Linux (no Windows-only project).

• We will assign a project to each individual during class.

• Note that this is vitally important as much of your oral exam will be from what work you accomplish with the project!

• A list of H/FOSS project can be found at http://www.xcitigroup.org/softhum/doku.php?id=g:hfoss_and_oss_projects

• Please place the presentation in the dropbox or email me.