Course Review

Stuart Anderson
Overview

- Review of course.
- Preparing for the exam.
- Identify most-favoured revision topics.
- Feedback questionnaire
It cannae be done captain!

• Program proof is impractical, complete testing ditto for similar reasons plus the fact that a software system in context is much more complex than a “simple” program.

• So grim reality is that we must do the best we can.
Well how, then?

Systematise our testing — what can we test, and can we come up with reusable methods of testing?

- Functional testing
- Structural testing – carries risks and benefits
- Higher level testing – more and more context-sensitive
And when!?

Testing exists in a software lifecycle

- Where in the lifecycle can testing have an impact?
- Design for tests
- Test your designs and specifications
- Static:
  - Black-box: examine specifications
  - White-box: code review
- Dynamic:
  - Black-box: functional testing
  - White-box: structural testing
Isn’t easy

Some things can be hard to test:

- System properties such as usability
- GUIs
- Interaction with physical systems
Your tests better than mine?

Can we say some tests are better than others?

- Adequacy, coverage: running theme throughout – structural, integration, GUI
- Mutations
How does it work?

- Understanding
- Specification
- Implementation
- Lifecycle

- Tools are largely essential, and helped by reflection in Java — Lively field, with lots of new and recent work
Achievements, Challenges, Dreams

Software testing research roadmap

Achievements

- Testing process
  - Reliability testing
    - Defect-oriented testing
  - Protocol testing
    - Test criteria
    - Comparison among test criteria

Challenges

- Why
  - Education of software testers
- How
  - Understanding the costs of testing
    - Test input generation
    - Model-based testing
    - Anti-model-based testing
    - Explicit test hypotheses
      - Test effectiveness
      - Empirical body of evidence
    - Compositional testing
- How much
- What
  - Testing patterns
    - Controlling evolution
    - Leveraging user population and resources
  - Domain-specific test approaches
    - On-line testing
- Where
- When

Dreams

- Efficacy-maximized test engineering
- 100% automatic testing
- Test-based modeling
- Universal test theory

[Bertolino, 2007]
Slide 8: Achievements, Challenges, Dreams

**Required Readings**


  [http://dx.doi.org/10.1109/FOSE.2007.25](http://dx.doi.org/10.1109/FOSE.2007.25)
Exam

- No surprises if you pay attention to previous exams
  http://www.inf.ed.ac.uk/teaching/exam_papers/

- General topics:
  - Functional testing
  - Structural testing
  - Lifecycle, higher level testing
  - Not exclusive though, and anyway cross-topic knowledge and context is very good for rounding out your answers so don’t try ignoring one of these three.

- Aspects which don’t get proportional emphasis in the tutorials, so pay attention:
  - Definitions (need to be precise, but not necessarily mathematical)
  - Areas where there’s more content than method (lifecycle, integration/regression/GUI/higher level testing)
  - General “big picture” aspects: context, interrelations, themes, etc.
Feedback

Please fill in the feedback questionnaire:

http://www.inf.ed.ac.uk/cgi-bin/ito/course_questionnaire.pl