An Overview of Verification and Validation

Software Requirements & Project Management
CITS 3220
"An effective way to test code is to exercise it at its natural boundaries" Brian Kernighan

"Testing is the process of comparing the invisible to the ambiguous, so as to avoid the unthinkable happening to the anonymous." James Bach

"Testing is organised skepticism." James Bach

"Program testing can be used to show the presence of bugs, but never to show their absence!" Dijkstra

"Beware of bugs in the above code; I have only proved it correct, not tried it." Knuth

"Software Testers: Depraved minds, usefully employed." Rex Black

http://en.wikipedia.org/wiki/Software_testing
Lecture Overview

- V&V definitions
- Different Test Purposes
- Classification of Test Methods
- Planning and Organisation
- When to stop testing
Examination Question

when you have studied the material in the next 2 lectures you will be able to answer this

Suppose CSSE has decided to replace end of semester written exams with an online examination system to be run in CSSE labs.

Identify 4 goals for this system. Explain the importance of each goal for the stakeholders of the new system.

For each goal, propose tests to be used to validate that the final system meets the goal and describe an appropriate technique for performing the test.
Verification and Validation

Assuring that a software system meets a user's needs
Verification vs validation

- Verification: to establish the truth of correspondence between a software product and its specification (from the Latin *veritas*, “truth”)
- Validation: To establish the fitness or worth of a software product for its operational mission (from the Latin *valere*, “to be worth”)

Ref: Boehm, SE Economics 1981
Verification vs validation

- **Verification:**
  "Are we building the product right?"
  The software should conform to its specification

- **Validation:**
  "Are we building the right product?"
  The software should do what the user really requires
The V & V process

- Is a whole life-cycle process - V & V must be applied at each stage in the software process.

- Has two principal objectives
  - The discovery of defects in a system
  - The assessment of whether or not the system is usable in an operational situation.
Different Test Purposes
Defect vs Reliability Testing

- **Defect testing**
  - Tests designed to discover system defects.
  - A successful defect test is one which reveals the presence of defects in a system.

- **Reliability testing (aka Statistical testing)**
  - Tests designed to reflect the frequency of user inputs. Used for reliability estimation.
Testing vs debugging

- Defect testing and debugging are distinct processes.
- Verification and validation is concerned with establishing the existence of defects in a program.
- Debugging is concerned with locating and repairing these errors.
- Debugging involves formulating a hypothesis about program behaviour then testing these hypotheses to find the system error.
The debugging process

1. Test results
2. Locate error
3. Design error repair
4. Repair error
5. Re-test program
6. Test cases
7. Specification

Steps in the debugging process:
- Test results
- Locate error
- Design error repair
- Repair error
- Re-test program
- Test cases
- Specification
What can we do about faults?

- Three approaches to reducing the number of faults in computer systems:
  - avoidance
  - detection & removal
  - tolerance
1. Fault Avoidance

- Aim: to ensure faults are not introduced into a system in the first place

- Techniques:
  - rigorous SW development process e.g. clean room
  - defensive programming
  - reviews of requirements, design, and implementations

- Comments:
  - this approach could be called fault *intolerance*
2a. Discovery and Removal

- **Aim:** to discover and remove faults
- **Techniques:**
  - software testing
  - corrective maintenance
2b. Run Time Detection

- **Aim:** to identify errors at run time, before they lead to system failures

- **Techniques:**
  - functionality and consistency checks
  - checksums, cyclic redundancy codes, error correcting codes
  - watchdog timers
3. Fault Tolerance

- **Aim:** to use redundancy to provide increased safety for computer systems

- **Techniques:**
  - time redundancy: recovery/timeout blocks, active replication
  - SW redundancy: information redundancy, multiple computations
  - HW redundancy: passive or active replication
Fault Handling Summary
Classification of Test Methods
Static vs Dynamic

- **Software inspections** Concerned with analysis of the static system representation to discover problems (static verification)
  - May be supplement by tool-based document and code analysis
- **Software testing** Concerned with exercising and observing product behaviour (dynamic verification)
  - The system is executed with test data and its operational behaviour is observed
Static and dynamic V&V

Static verification

- Requirements specification
- High-level design
- Formal specification
- Detailed design
- Program

Dynamic validation

Prototype
White box vs Black box

- **White box tests** focus on the internal structure of a component
- **Black box tests** focus on the input/output behaviour of a component
Planning and Organisation
V& V goals

- Verification and validation should establish confidence that the software is fit for purpose
- This does NOT mean completely free of defects
- Rather, it must be good enough for its intended use and the type of use will determine the degree of confidence that is needed
V & V confidence

- Depends on system’s purpose, user expectations and marketing environment
  - Software function
    - The level of confidence depends on how critical the software is to an organisation
  - User expectations
    - Users may have low expectations of certain kinds of software
  - Marketing environment
    - Getting a product to market early may be more important than finding defects in the program
V & V planning

- Careful planning is required to get the most out of testing and inspection processes.
- Planning should start early in the development process.
- The plan should identify the balance between static verification and testing.
- Test planning is about defining standards for the testing process rather than describing product tests.
Testing has its own Life Cycle

- Establish the test objectives
- Design the test cases
- Write the test cases
- Test the test cases
- Execute the tests
- Evaluate the test results
- Change the system
- Do regression testing
Testing is often viewed as dirty work.

To develop an effective test, one must have:
- Detailed understanding of the system
- Knowledge of the testing techniques
- Skill to apply these techniques in an effective and efficient manner

Testing is done best by independent testers
- We often develop a certain mental attitude that the program should in a certain way when in fact it does not.

Programmer often stick to the data set that makes the program work
- "Don’t mess up my code!"

A program often does not work when tried by somebody else
- Don't let this be the end-user.
Test Team

- Professional Tester
- Analyst
- User
- System Designer
- Configuration Management Specialist
- Programmer

(too familiar with code)
Acknowledgements

The material in this lecture is adapted from lecture notes of

- Sommerville (6th ed) chapter 19
- Bruegge & Dutoit chapter 9