University of Windsor

LECTURE 6 – TESTING (PART 3)

Master of Applied Computing

COMP-8117 : Advanced Software Engineering Topics

Dr. Aznam Yacoub – aznam.yacoub@uwindsor.ca
School of Computer Science

SCHEDULE

- Introduction
- Static Tests in Details
- Conclusion

PROGRAM INSPECTIONS

- Myth : « The only way to test a program is to execute it ».
- Generally, software testers don’t read code.

PROGRAM INSPECTIONS

- But software is a set of programs.
- Programs are mathematical descriptions of something.

PROGRAM INSPECTIONS

- Computer Science => Computer Program => Computations => Mathematics
- Consequence : Program code can be studied.
- This is static testing.
- Static testing is white-boxed.
- Static testing is a verification process.

HUMAN TESTING

- Non-computer-based testing process.
- Effective in finding errors.
- Typically step before the computer-based testing.
- Note: programmers will add more errors when trying to fix errors found during computer-based testing.

HUMAN TESTING

- Therefore, human testing involves a specific team reading or visually inspecting a program.
- Preparatory work: « meeting of the minds »
  - Objective : Test (not DEBUG) = find errors (not find the solutions)
  - 3-4 developers perform reviews
  - Only one author
  - Generally find 30-70% of the logic flaws and coding errors

CODE INSPECTIONS

- Set of procedures and error-detection techniques for group code reading
  - Focus on the procedures, forms, etc.
  - 1 Quality Control Engineer
    - Distribute materials, schedules
    - Leading the session
    - Recording all errors found
    - Ensuring that the errors are subsequently corrected
  - Other members are program design, and test specialist

CODE INSPECTIONS

- Well implemented in big companies like Ubisoft, Microsoft, EA, IBM…
- Less use in startup (Developers debug…)
- This team is called Internal Testers, Testing Team, Verification and Validation Team, Test Engineers
- Generally work following a STLC (Software Testing Lifecycle) in parallel of the SDLC

ERROR CHECKLIST FOR INSPECTIONS

- Data Reference Errors
- Date Déclarations Errors
- Computation Errors
- Comparison Errors
- Control-Flow Errors
- Interface Errors
- Input/Output Errors

WALKTHROUGHS

- Same approach than Inspections but:
  - Rather than simply reading the program or using error checklist, the participants « execute » the code as if they were computers.
  - Use of test cases (input/expected output)
  - Mentally execute the program : test data are walked through the logic of the program, state of the program is monitored on paper or whiteboard.

DESK CHECKING

- Inspection / Walkthrough done by one person (generally the programmer testing his/her own code)
- Unproductive:
  - The programmer never executes the program, but what he thinks the program does
  - Undisciplined approach – Not suitable in software engineering

PAIR PROGRAMMING

- Agile technique
- Two programmers work on the same workstation
- One write the code (the driver)
- The other (the navigator) reviews each line of code

PEER RATINGS

- Another human review process but not associated with program testing (the objective is NOT to find errorrs).
- Anonymous evaluations of programs in terms of quality, maitainability, extensibility, usability, and clarity.
- Use a questionnaire with a scale:
  - Is the program easy to understand ?
  - Is the high-level design visible and reasonable ?
  - Is the low-level design visible and reasonable ?
  - …

PEER RATINGS

- Another human review process but not associated with program testing (the objective is NOT to find errorrs).
- Anonymous evaluations of programs in terms of quality, maitainability, extensibility, usability, and clarity.
- Use a questionnaire with a scale:
  - Is the program easy to understand ?
  - Is the high-level design visible and reasonable ?
  - Is the low-level design visible and reasonable ?
  - …

STATIC ANALYSIS

- Evaluation of the quality using tools like theorem prover, model checking, syntax analyzer…
- Detects:
  - Unused or uninitialized variables
  - Dead code
  - Infinite loops
  - Undefined values
  - Wrong syntax
  - Coding standards
  - Code optimization

STATIC ANALYSIS

- Three types:
  - Data flow : stream processing
  - Control flow : execution of statements and instructions
  - Cyclomatic complexity : number of independent paths in the control flow graphs of a program

SUMMARY

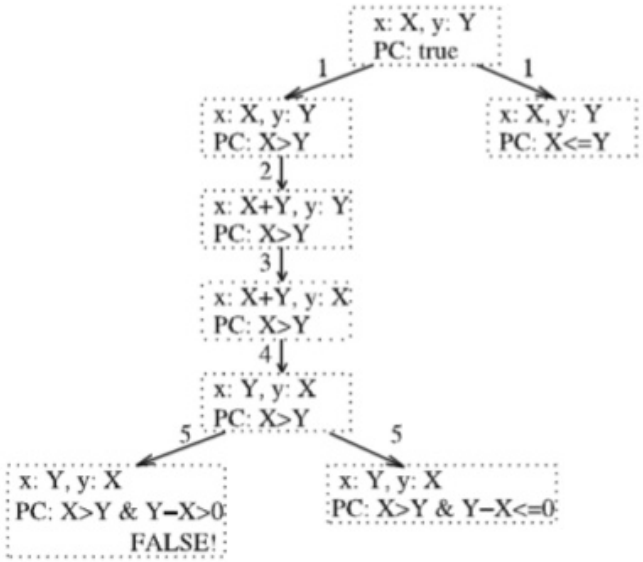
- Main methodologies:
  - Peer-review/Pair-programming and code inspection : two programmers per task, one reads, the other writes
  - Code Analysis : Type checking, interface checking, unused variables, unreachable code
  - Symbolic Evaluation : Simulate the execution using symbolic values

SUMMARY

Symbolic Execution

- An example

```
    int x, y;
1:  if (x > y) {
2:    x = x + y;
3:    y = x - y;
4:    x = x - y;
5:    if (x - y > 0)
6:      assert(false);
    }
```



Code that swaps two integers

Symbolic Execution Tree

Corina S. Păsăreanu, Willem Visser. A survey of new trends in symbolic execution for software testing and analysis. 339-353 2009 11 STTT 4

36/50

REFERENCES

This lecture is based on:

- Introduction to Software Testing – Ziad Kobti – University of Windsor
- Test – Amine Hamri – Aix-Marseille University
- CSCI3060U – Jeremy Bradbury - Ontario Tech University
- CSCI 5828 – Kenneth Anderson
- The Art of Software Testing – Glenford Myers

