Seven Principles of *(highly effective)* Software Testing

by

Bertrand Meyer

Presented by

Jerry Sun
Peter Guy
Who is Bertrand Meyer?

Long-time Computer Scientist

Author of *Object-Oriented Software Construction*

Created the programming Language Eiffel
- OO
  - Design-by-Contract

Teaches at ETH Zurich
Software Testing

What is the scope of software testing?

   Quality assessment?

   Finding bugs?

   Employing interns?
On-demand computer scan

- Smart scan
  - Local disk scan
- Custom scan...
  - Selection of scan profile and targets to scan

Scan setup...

Date and time of the last scan: 2009/11/21 下午 12:05:36
Last scan result: 0 infected files
Virus signature database used in last scan: 4625 (20091120)
Principle 1: Definition

To test a program is to try to make it fail

- keeps the process focused - single goal
- don't correct faults, just find them
Principle 2: Tests versus specs

Tests are no substitute for specifications

- Tests are only instances - points - that can be interpolated into a curve, but the area between the points is uncovered.

- Specifications can produce tests (model-based testing, for example), but tests cannot generate specifications.
Principle 3: Regression testing

Any failed execution must yield a test case, to remain a permanent part of the project’s test suite.

• Once you have uncovered a fault it must remain part of your life forever.

• AV Software performs regression testing by rescanning all files and by keeping a virus definition in the dictionary.
Principle 4: Applying Oracles

Determining success or failure of tests must be an automatic process.

• As the number of test cases increases, manually examining the results becomes less feasible.

• Anti-Virus software interprets the results of scanning files and only tells you when it finds malware.

Date and time of the last scan: 2009/11/21 下午 12:05:36
Last scan result: 0 infected files
Virus signature database used in last scan: 4625 (20091120)
Principle 5: Manual and automatic test cases

An effective testing process must include both manually and automatically produced test cases.

- Manual tests take advantage of the tester's domain knowledge to do more in-depth testing.

- Automatic tests take advantage of tireless machines to do more testing than humans can.
Principle 6: Empirical assessment of testing strategies

Evaluate any testing strategy, however attractive in principle through objective assessment using explicit criteria in a reproducible testing process.

- Don't rely on intuition - testing is tricky.

- There is no substitute for empirical assessment.
Principle 7: Assessment criteria

A testing strategy's most important property is the number of faults it uncovers as a function of time.

- What matters is how fast a strategy can produce failures revealing faults.
- The relevant function is fault count against time.
  - Criteria for principle 6.
  - Fault density.
  - When to stop testing.
"Testing a program tells us little about its quality, since 10 or even 10 million test runs are a drop in the ocean of possible cases."

Therefore:
• Don't try to do too much - find bugs, don't fix them!
• Automate!

"Testing is tricky"

Therefore:
• Try it out - your intuition may not be right.