DANSE Software Testing

Tom Swain
Software Quality Research Laboratory
Department of Electrical Engineering and Computer Science
University of Tennessee
Software Certification

Objective evidence of software fitness for intended uses
Certification: The User’s Point of View

- Documented Specifications
- Specifications Peer-reviewed
- Code Inspected
- Verification Testing
- Beta Test

Confidence in Results
Verification Testing

Applications and Other Deliverable Packages

- Test Plan
  - Test boundary
  - Usage model
  - Test Cases
  - Oracle definition

- Automation (optional)
  - Test case generation & analysis (via JUMBL)
  - Run time automation (via PyUnit & dogtail?)

- Test Execution
  - Run test cases
  - Record pass/fail
  - Write Trac tickets for failures
  - Fix faults & close tickets
  - Iterate until completion criteria met
Prerequisites

- Python 2.4.2, or later
- Java 1.4, or later
- JUMBL 4.5
JUMBL Functions Needed

- Check usage model structural consistency
- Analyze model for validation and test planning
- Generate test cases as Python scripts
- Maintain test record
- Perform reliability analysis
Python unittest Module (aka PyUnit)

- Stone soup: They provide the broth; we provide the meat & veggies
- Not just for unit test
**Test Framework**

### Common

- **TestLoader**
- **MBSTLoader**
  - testDir : string(idl)
  - loadTestsForMBST()
- **MBSTRunner**
  - model : string(idl)
  - testDir : string(idl)
  - results : TestResult
  - debug : boolean(idl)
  - usage()
  - run()
  - recordResults()
  - printResults()
  - main()

### Specific to SUT

- **TestCase**
- **ModelTestTemplate**
  - testCaseName : string(idl)
  - setup()
  - execute()
- **<model>_n_.py**
  - **ModelTest**
    - runTest()
Test Automation: Static Applications
Test Sequence: Dynamic Applications
GUI Test Automation:
Free Options Look Messy

- **dogtail**
  - Looks very good conceptually
  - GNOME (Unix) desktop only
  - Poor documentation

- **Windows options**
  - pywinauto – may be something to build on
  - Win32::GUI::Test – Perl-based
  - AutoIT – BASIC-like scripting

- **Mac options**
  - No free tools found
Some GUI Alternatives

- Test all GUIs on Linux/GNOME
- Design to MVC pattern and emulate view
- Pay the ransom for a cross-platform commercial tool
- Pick a free Windows tool and make it work
TBD

- Semiautomated usage model generation from specification info
- Simple GUI interface to framework
- Absorb gentest into autombst.py
- Integrate dogtail for GUI testing
- Stateful SUT demo