
Automated Testing of Embedded Software

Test Automation 2003

Douglas Hoffman
Software Quality Methods, LLC.
24646 Heather Heights Place
Saratoga, California 95070-9710
Phone 408-741-4830
Fax 408-867-4550
doug.hoffman@acm.org
www.SoftwareQualityMethods.com

Copyright © 2003, Software Quality Methods, LLC. No part of these graphic overhead slides may be reproduced, or used in any form by any electronic or mechanical duplication, or stored in a computer system, without written permission of the author.

Douglas Hoffman

Copyright © 2003, SQM, LLC.

1

Topics

- Background
 - Software testing
 - Test automation
 - Test oracles
- Embedded software
- Automation techniques

Douglas Hoffman

Copyright © 2003, SQM, LLC.

2

Running A Software Test

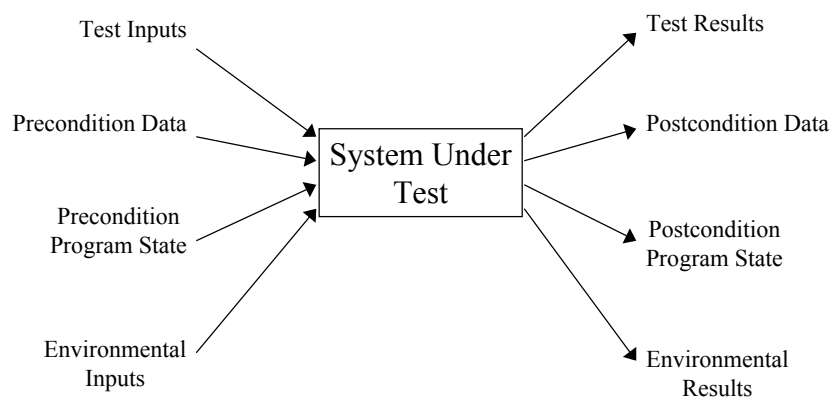
- Test setup
 - SUT program state
 - Data values
 - System environment
- Run test exercise
- Capture/compare actual with expected results

Douglas Hoffman

Copyright © 2003, SQM, LLC.

3

Expanded Testing Model



Douglas Hoffman

Copyright © 2003, SQM, LLC.

4

(Hoffman's) Fully Automated Software Tests

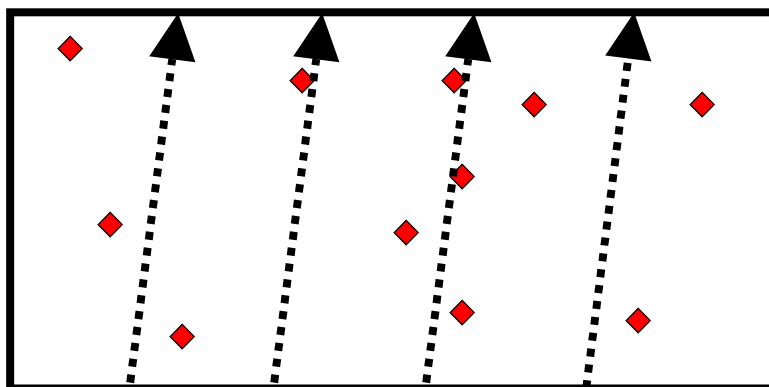
- Able to run two or more specified test cases
- Able to run a subset of the automated test cases
- No intervention needed after launching tests
- Automatically set-up and/or record relevant test environment
- Run test cases
- Capture relevant results
- Compare actual with expected results
- Report analysis of pass/fail

Douglas Hoffman

Copyright © 2003, SQM, LLC.

5

Testing Analogy: Pulling Weeds



◆ weeds

This analogy was first presented by Brian Marick.
These slides are from work by James Bach.

Douglas Hoffman

Copyright © 2003, SQM, LLC.

6

Repeating Your Path Won't Clear the Weeds

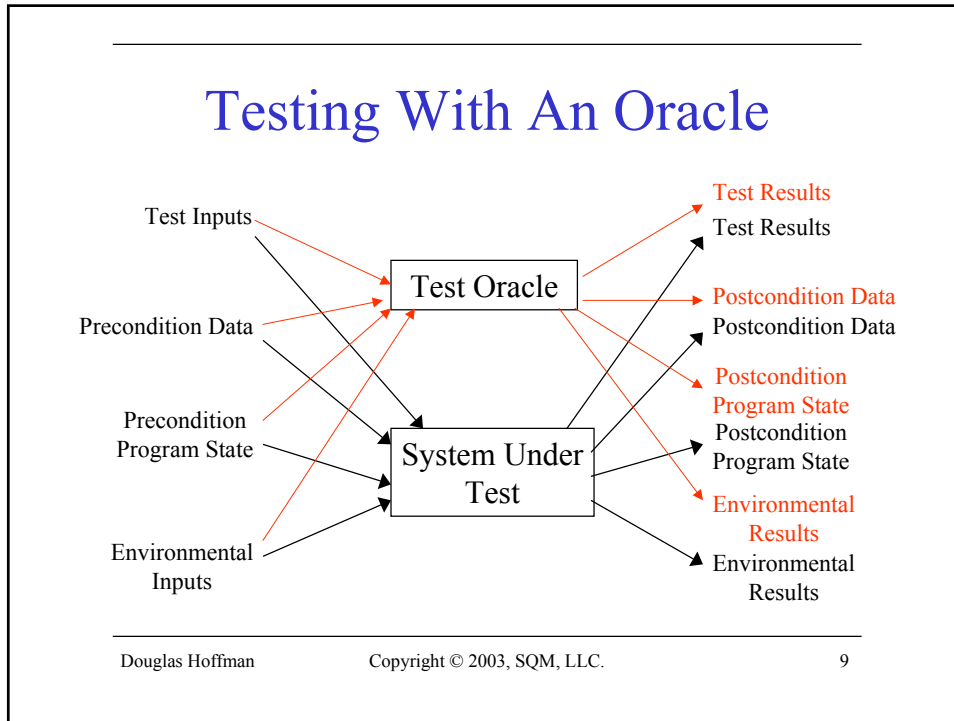
◆ weeds ● fixes

Douglas Hoffman Copyright © 2003, SQM, LLC. 7

Why Automate Tests?

<u>Can help:</u>	<u>Will not help:</u>
<ul style="list-style-type: none">• Repeatability• Streamlining• Consistency• Leverage• Scope	<ul style="list-style-type: none">• Faster test development• Fewer resources• Strong regression tests• New tests• Sufficiency

Douglas Hoffman Copyright © 2003, SQM, LLC. 8



Oracle Strategies for Verification

	No Oracle	True Oracle	Consistency	SelfReferential (SVD)	Heuristic
Definition	- Doesn't check correctness of results	- Independent generation of all expected results	- Verifies current run results with a previous run (Regression Test)	- Embeds answer within data in the messages	- Verifies some characteristics of values
Advantages	- Can run any amount of data (limited only by the time the SUT takes)	- All encountered errors are detected	- Fastest method using an oracle - Verification is straightforward - Can generate and verify large amounts of data	- Allows extensive post-test analysis - Verification is based on message contents - Can generate and verify large amounts of complex data	- Faster and easier than True Oracle - Often much less expensive to create and use
Disadvantages	- Only spectacular failures are noticed.	- Expensive to implement - Complex and often time-consuming when run	- Original run may include undetected errors	- Must define answers and generate messages to contain them	- Can miss errors - Can miss systematic errors

Douglas Hoffman Copyright © 2003, SQM, LLC. 10

What is Embedded Software?

- Software that is part of a larger system and performs some of the requirements of that system (IEEE 610.12-1990)
- Any software system that must be designed on a platform different from the platform on which the system is intended to be deployed (Encontre, Spring 2002, STA)

Environments

- Simulator
 - Software based
 - Debugger capabilities
- Emulator
 - Hardware & software
- Target Environment
 - Cross compilers
 - Weak debuggers

Test Hooks and Scaffolding

- Purpose:
 - Monitor
 - Control
 - Stimulate
- Environment specific mechanisms
 - Interface with SUT
 - Fill in for co- and sub-processes
 - Rationalizing actions

Oracle Strategies for Embedded Software

- No oracle approach is a poor choice
- True oracle may be available
- Different environments are potential consistency oracles
- Self referential approach usually not applicable
- Heuristic approach may be required

Automation Techniques for Embedded Software

- Scripting
- Frameworks
- Individual test programs

Test Scripting

- User language
- Test development independent from tools
- Separates test exercises from mechanisms
- Tailorable to the context
- Table driven tests are simplest examples
- ***Not*** capture/playback

Scripting for Embedded Systems

Pros:

- Natural language
- Can operate across environments
- Tailorable to contexts
- Can evolve as needed
- Separates test specifications from mechanisms

Cons:

- Resource sink
- Environment dependencies
- Limits what is possible or interesting
- Can limit to least common denominator

Frameworks

- Subroutine library
- Common actions
- Complex, repeated actions
- Test development requires programming

Framework for Embedded Systems

Pros:

- Standardization
- Efficiency
- Enforces discipline
- Can evolve as needed

Cons:

- Limits what is possible or interesting
- May be one person's ideal
- Can limit to least common denominator
- Resource sink
- Maintenance costs

Individual Test Programs

- Tests are explicit programs
- Seeking widely different errors
- Independent tests
- Separable test activities

Individual Test Programs

Pros:

- Can craft each test
- Test specifications is the mechanism

Cons:

- Each test is crafted
- High cost of development
- Strong environment dependencies
- High maintenance costs

Choosing A Strategy

- Understand the development and platform environments
- Evaluate the automation options
- Identify the oracle strategy
- Consider costs and benefits between manual and automated testing

Closing Thoughts

- Testing embedded software can be challenging
- Test automation is hard work
- The combination can be *interesting*:
 - platforms / environments
 - test creation and maintenance
 - getting useful information
 - oracles and validation



Douglas Hoffman

Copyright © 2003, SQM, LLC.

23

References

- Encontre, Vincent; "Test, Observe, and Assess Embedded Applications During Development," International Conference on Software Test Automation, Spring 2002.
- Hoffman, Douglas; "Using Test Oracles in Automation," Pacific Northwest Software Quality Conference (PNSQC), 2001
- IEEE Std **610.12-1990**, IEEE Standard Glossary of Software Engineering Terminology

Douglas Hoffman

Copyright © 2003, SQM, LLC.

24