Automated Testability
The Missing Link in Test Automation

John A. Fodeh
fodeh@mail.com

W14 – Dec. 1st 2004
Contents

- Rationale for Test Automation
- What is Automated Testability
- Design for Automated Testability
- Applying a Risk Based Approach
Rationale for Test Automation

- Increasing software size and complexity
- Demanding regulations
- Shorter time-to-market
- Better quality
- Agile development methodologies
Many test automation initiatives fail!

- A key factor for failure is that software is not developed with test/automation in mind
  - Missing management awareness
  - Test/automation needs not included in requirements
  - Automation applied late, taking too much time
  - Immature approaches using Capture/Replay through the Graphical User Interface
What is Automated Testability

“Automated testability is the degree to which the software under test facilitates the implementation, execution and maintenance of automated testing”

Automated testability is about interfaces:

- Between software under test and test software
- Between requirements and implemented features
- Between developers and testers
The Price of Poor Automated Testability

- Higher implementation effort
- Higher maintenance effort
- Buggy and unstable scripts
- Automating what is easy to automate instead of what is important!
- Loss of confidence in test tool
- "Shelfware"
Quality Attributes of Automated Testability

- Visibility
- Control
- Persistence
- Consistency
- Reliability
- Documentation
Visibility

Ability to identify: output, states, properties, system interactions, resource usage, errors

- Completion of actions
- Current state
- Intermediate results
- Error messages
- Disk space, memory and CPU usage

Visibility is essential for synchronization
Control

Ability to exercise system

- Enter input
- Trigger events
- Invoke methods
- Manipulate GUI widgets
- Using interfaces

Custom and dynamic GUI controls are problematic!
Persistence

- The extent and frequency of change in the software under test

  - Change frequency has great impact on maintenance of automated tests
    - Changes must be well considered and carefully planned
    - Impact on test/automation (and side-effects) must be evaluated
    - Changes must be communicated
GUI Changes

Version 1.23

- Window captions
- Control type
- Additions and replacements

Version 1.24

- Default values
- Invisible changes (e.g. internal control name/ID)
Consistency

- The level of coherence in the look, operation and performance of the software under test
- Consistency is essential for developing automation libraries
- Test design patterns
Reliability

- The ability of a system to perform its intended function for a specified period of time
- Tests repeated under identical conditions produce the same results
- A buggy and unstable system can block testing and automation
Documentation

- Well specified system and interface is a prerequisite for automation (and testing)
- Technical documentation should be available and accurate
- Testers should have access to relevant resources and “oracles”
- Changes should be communicated
Benefits

- Robust, cost-effective and efficient test automation

Side benefits:
- Testers gain better understanding of system design and behavior
- Easier way to reproduce bugs
- Better manual testing
- Better debugging facilities
- Improved software maintainability
- Improved learnability and usability of system
Test Automation is Software Development

- Apply software development best practices
- Coding standard
- Design for maintainability, reusability
- Version and source control
- Review
- Design documentation
- Error handling
- Test
Typical Development and Test Organization

- Business Analyst
- Software Analyst
- Software Developer
- Test Analyst
- Test Developer
- Test Case
- System Under Test
- Test System
A Practical Development and Test Organization

Business Analyst

Software Analyst
Software Developer

Test Analyst
Test Automator

Test System
Test Case

System Under Test
Automation Impact

Repeatability
- Regression tests, smoke test of daily build
- Number of supported platforms, hardware configurations

Importance
- High risk functionality
- Tedious but valuable test

Customer Value
- Usage intensity

Effort to run manually
- Complex test, requires specialized skills
Applying a Risk-Based Approach

- Assess automation impact: Repeatability, Importance, Customer Value, Effort to run manually
- Assess automated testability: Visibility, Control, Persistence, Consistency, Reliability and Documentation
- Rank each factor using scale:
  - Low (1)
  - Medium (2)
  - High (3)
- Plot in matrix
Applying a Risk-Based Approach

Example:
- Installation wizard (IW)
- Export facility (EF)
- Report generator (RG)
- Online help (OH)

Automation Impact

Automated Testability

- Don’t automate
- Automate

Example:
- Installation wizard (IW)
- Export facility (EF)
- Report generator (RG)
- Online help (OH)
Promoting Automated Testability

- Early involvement of testers
  - Test and automation requirements
  - Assessment of automated testability on prototypes
  - Security issues must be considered

- Coding standards
  - Naming convention
    - Examples: Check Box chkReadOnly (MSDN)
  - Guidelines for software design (especially GUI)
Promoting Automated Testability

Application Architecture
- Self-test (automated test incorporated in AUT)
- Test interface for special controls
- Alternative interfaces: Application Programming Interface, Command Line Interface, Protocol Interface

Test Team Structure
- Establishing quality gates throughout the process
- Making test automation a project issue
- Communicating impact of poor automated testability
Summary

Test automation requires a collaborative effort from testers, developers and project managers

- Early involvement of testers
- Automation requirements are well-defined and communicated at project start
- Automation is an integrated part of development

Cost-effective test automation calls for automated testability

- Automated testability benefits manual testing
- Automated testability helps building better systems
Further Info

Mark Fewster and Dorothy Graham, Software Test Automation, Addison Wesley, 1999

www.grove.co.uk

Linda Hayes, Automated Testability Tips, StickyMinds.com, Column: Nov 11, 2002

www.stickyminds.com

Software Testing Hotlist

www.testinghotlist.com


www.stqemagazine.com

Ed kit: Integrated, Effective Test Design and Automation, Software Development online, issue February 1999

www.sdmagazine.com
Speaker Details

John A. Fodeh

- Test coordinator at Oticon
- Several years in the field of software testing, test automation and process improvement
- Key person the SPI project “WHEN”
- ISEB foundation certificate in software testing
- Presentations at EuroSPI, EuroSTAR, BCS SIGIST and STARWEST
- Msc. From the Technical University of Denmark