Automated Testability

The Missing Link in Test Automation

John A. Fodeh
fodeh@mail.com

STAR*WEST*
Oct. 30, 2003

Contents

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

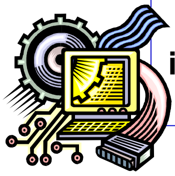
Rationale for Test Automation

- Increasing software size and complexity
- Demanding regulations
- Shorter time-to-market
- Better quality
- New and iterative development models

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
  - Software incompatible with automation tool
  - Automation applied late, taking too much time
  - Automated tests very vulnerable
  - Immature approaches using Capture/Replay through the graphical user interface (GUI)

What is Automated Testability



"Automated testability is the degree to which the application under test facilitates the implementation, execution and maintenance of automated testing"

Automated testability is about interfaces:

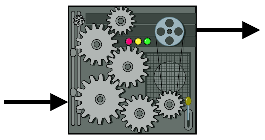
- Between software under test and test software
- Between requirements and implemented features
- Between developers and testers

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!
- Ineffective and inefficient tests
- Loss of confidence in test tool
- "Shelfware"

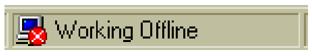
Quality Attributes of Automated Testability

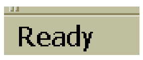
- Visibility
  - Applying a glass-box approach
- Control
  - Ability to exercise system parts
- Persistence
  - Frequency of change
- Consistency
  - Similar parts behave in a similar manner
- Reliability
  - Probability that system will perform its intended function
- Documentation
  - Information on how system should function

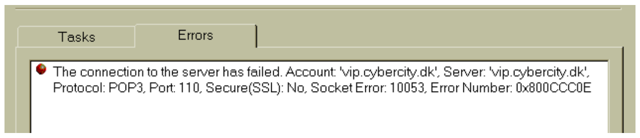


Visibility

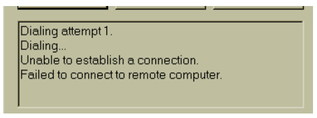
- Ability to identify: output, states, properties, system interactions, resource usage, errors
  - Reporting completion of actions
  - Communication status
  - Unexpected events, warnings and error messages
- Visibility is essential for synchronization
- Security issues must be considered







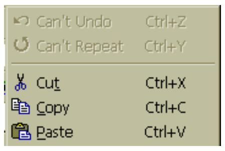
The connection to the server has failed. Account: 'vip.cybercity.dk', Server: 'vip.cybercity.dk', Protocol: POP3, Port: 110, Secure(SSL): No, Socket Error: 10053, Error Number: 0x800CCC0E


Dialing attempt 1.
Dialing...
Unable to establish a connection.
Failed to connect to remote computer.

B-K Medical

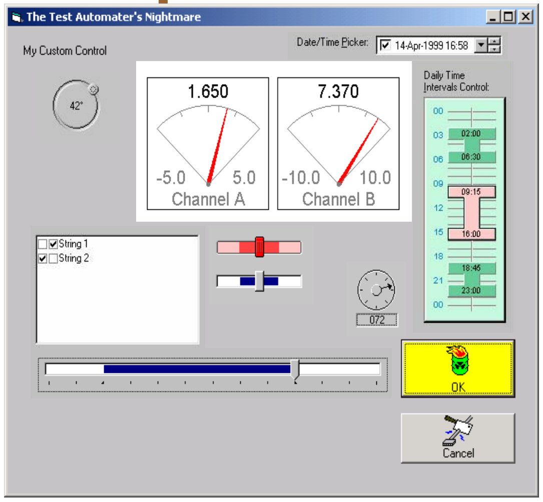
Control

- Ability to enter input, trigger events, Invoke methods, manipulate GUI widgets
  - Using standard GUI elements
  - Avoiding custom and dynamic controls
  - Keyboard access
  - Enabling/disabling of controls
  - Application Programming Interface (API)
  - Dedicated test interface



B-K Medical

Custom GUI Controls



The Test Automater's Nightmare
My Custom Control
Date/Time Picker: 14-Apr-1999 16:58
42°
1.650
-5.0 5.0
Channel A
7.370
-10.0 10.0
Channel B
Daily Time Intervals Control:
String 1
String 2
072
OK
Cancel

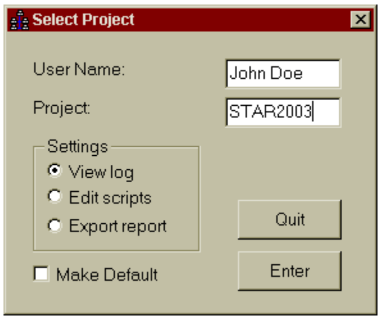
- GUI requirements have low priority
- GUI not specified in detail
- Custom controls not recognized
- Controls not identified <u>uniquely</u>
- Control properties or contents not visible
- Cannot operate on control
- Dynamic windows and controls

Persistence

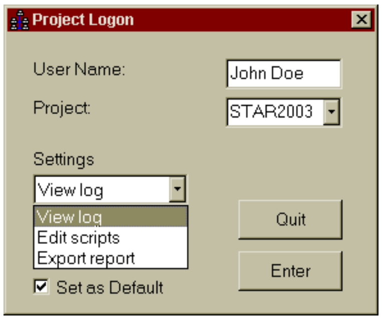
- The extent and frequency of change in the software under test
- Change frequency has great impact on maintenance of automated tests
  - Changes must be well considered and carefully planned
  - Impact on test/automation (and side effects) is evaluated
  - Changes are communicated

GUI Changes

Version 1.23



Version 1.24



- Window captions
- Control type
- Additions and replacements

- 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
- Applying standards for (GUI) programming
- Design (and test) patterns
- Naming convention
  - Examples: Check Box chkReadOnly
  - http://msdn.microsoft.com/

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
  - Tests (and defects) are reproducible
- System is stable and has a limited number of bugs
  - A buggy and unstable system can block testing and automation

Documentation

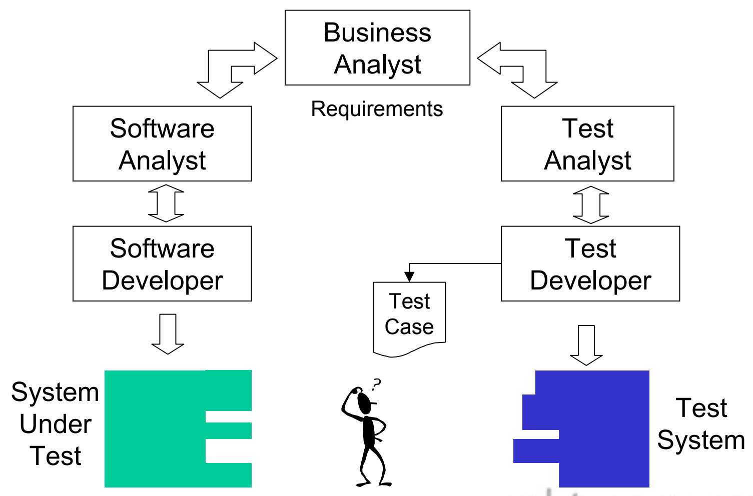
- A well specified system and interface is a prerequisite for automation (and testing)
  - Technical documentation/information must be available and accurate
  - When changes occur, documentation must be updated
  - Changes must be communicated

Benefits

- Robust, cost-effective and efficient test automation
- Side benefits:
  - Testers gain understanding of system design, behavior and vulnerabilities
  - Easier way to reproduce bugs
  - Better manual testing
  - Better debugging facilities
  - Improved software maintainability
  - Improved learnability and usability of system
  - Higher quality software

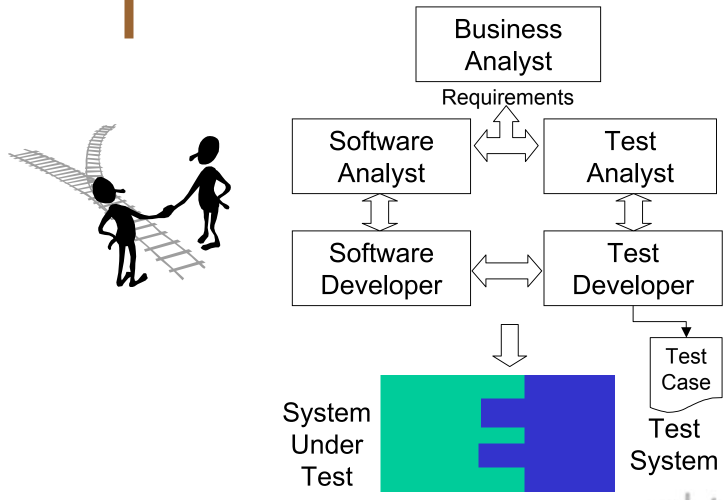
B-K Medical

Typical Development and Test Organization



B-K Medical

A Practical Development and Test Organization



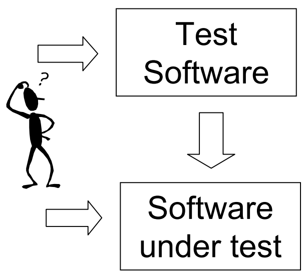
Large Scale Test Automation is Software Development

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

Test Team Structure

Possible team structures

- Test automation handled by testers
- Test automation handled by developers
- Test automation handled by separate team
  - Data-driven/action words approach separates test design and automation development
  - Testers determine the test design
  - Test "automators" implement test software



Automation Impact

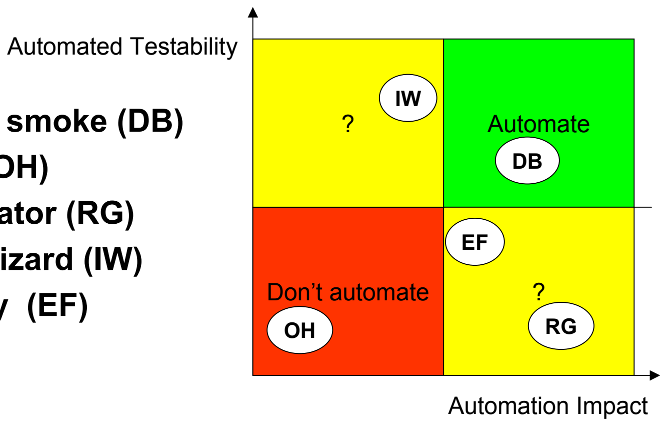
- Repeatability
  - Regression tests, daily build of smoke
- Portability
  - Number of supported platforms, hardware configurations
- Importance
  - Tedious but valuable test. Usage intensity. High risk tests.
- Effort to run manually
  - Complex test, requires specialized skills
- Simplicity
  - Technical challenge. Effort to implement

Applying a Risk-Based Approach (1)

- Assess automation impact: Repeatability, Portability, Importance, Effort to run manually, Simplicity
- 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 (2)

Example:
- **Daily build of smoke (DB)**
- **Online help (OH)**
- **Report generator (RG)**
- **Installation wizard (IW)**
- **Export facility (EF)**



Handling Lack of Automated Testability (1)

- **Apply workarounds**
  - **Keyboard access and shortcuts**
  - **Copy/paste to clipboard**
  - **Optical Character Recognition**
  - **Windows messages**
- **Bypass GUI**
  - **Direct access to database, registry, files, etc.**
  - **Use alternative interfaces: API, Command line**

Handling Lack of Automated Testability (2)

- Change scope of test automation
  - Don’t automate!
- <u>Change application under test</u>
  - Change problematic GUI elements
  - Build in test facilities: Event logging, state monitoring, dumping information in tabular text form
  - Add specialized test interface
- Communicate impact of poor automated testability

Promoting Automated Testability

- Early involvement of testers in requirement phase
- Test and automation requirements are considered
  - Naming convention for GUI elements
  - Predefined and unique control names/Id
  - Guidelines for GUI design and style
  - Error reporting convention
- Test interface for special controls
- Application Programming Interface
- Self-test
  - Incorporate automated test in software under test

Summary

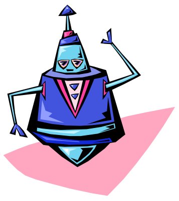
- Test automation requires a collaborative effort from testers, developers and project managers
  - Early involvement of tester in requirements
  - Automation requirements are well defined and communicated at project start
  - Automation is an integrated part of the software delivery
- Cost-effective test automation calls for automated testability
  - Automated testability benefits manual testing
  - Automated testability helps build 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
- Hans Buwalda and Maartje Kasdorp, Getting Automated Testing under Control, Software Testing and Quality Engineering, issue nov/dec 1999, Software Quality Engineering
  - www.stqemagazine.com
- Ed kit: Integrated, Effective Test Design and Automation, Software Development online, issue February 1999
  - www.sdmagazine.com

B-K Medical
A Subsidiary of Analogic Corporation

Speaker Details



John A. Fodeh

- Test manager at B-K Medical A/S
- 6 years in the field of software testing, test automation and process improvement
- Key person in the SPI project “WHEN”
- ISEB foundation certificate in software testing
- Presentations at EuroSPI2000, EuroSTAR 2001, BCS SIGIST and EuroSTAR 2002
- M.Sc. From the Technical University of Denmark

ultrasound we specialize because you do