Test Automation for Embedded Devices

Originally Created for:

5th International Conference on QA & Testing for Embedded Systems
October 18-20, Bilbao, Spain

Scott Barber
Chief Technologist
PerfTestPlus, Inc.
Introduction

Using Automation to Test Internet-Based Applications on Embedded Devices

Who Am I?

My name is Scott Barber and I’m a test-aholic…

Chief Technologist of PerfTestPlus, Inc.
Executive Director for the Association for Software Testing
Co-Founder of the Workshop On Performance and Reliability (WOPR)
Member of the Context-Driven School of Software Testing
Signatory of the Agile Manifesto for Software Development
Prolific author, speaker and columnist
Internet-based, embedded application testing specialist

…Oh yeah, I almost forgot, I’ve been involved with over 100 separate software testing projects.
Internet-Based, Embedded Applications

Software that requires Internet connectivity to make use of the entire feature-set.

Software that resides on (typically hand held) embedded systems.

Systems cannot (effectively) be instrumented by test by loading or installing test software directly onto the device.

Examples typically include:

- Cell Phones
- PDA's
- Set Top Boxes
- Cameras
Internet-Based, Embedded Applications

- Avionics, car control systems (anti-lock braking)
- Medical instruments
- Weapon systems

Embedded devices

Real-time systems

- PDAs, pagers, cell phones
- WinCE devices

Traditional PCs

- Ground systems for defense, aerospace, etc.
- Process control systems
- Simulation systems

- Digital cameras
- MP3 players
- Stereo equipment

- Web sites
- Client-server applications
- Single-user software (Word, Quicken, etc.)
Internet-Based, Embedded Applications

Significant, non-obvious testing challenges:

Real-time, embedded and PC-based software testing industries each have their own techniques, tools, approaches and terminology... and they rarely overlap.

Real-time and embedded software testers are typically the most senior engineers & developers on the project.

PC-based software testers are predominantly have little to no electrical engineering or development experience.

The majority of Internet-based, embedded applications are being tested by PC-based software testers.
Five Basic Approaches

Unit testing in testing in the IDE (Manual or Automated)

Human testers using actual devices (Manual)

Externally driven test automation via connection to a PC

Testing against simulators or emulators (Manual or Automated)

Back end testing via Internet (Manual or Automated)
Unit Testing in the IDE

Pros:

No special hardware or software required
Relatively easy to accomplish
Can ensure that software units function as designed when accessed independently from one another

Cons:

No indication of how units work in combination
No indication of how software will interact with system hardware
No ability to test realistic usage scenarios
Human Testers on Actual Device

Pros:

No special hardware or software required
Relatively easy to accomplish
Only way to effectively test usability and performance
Typically very important in addition to other forms of testing

Cons:

Can be extremely time consuming
Prone to human error
Typically limits total number of test cases, scenarios and variations tested
External Driven Automation

Pros:

- Can enable test automation
- Can increase test coverage
- Can enable scenario based testing

Cons:

- Often challenging to impossible to implement
- Typically fragile
- Frequently has unexpected effects on device hardware and state
- Rarely enables detailed or accurate test verification
Testing Against Simulators or Emulators

Simulators:

Generally allow users to experience what a thing will look and feel like but which doesn't use the same code base.

PC-based flight simulators are a good example.

Emulators:

Are pieces of software that allow an application written for one platform or operating system to be executed on another platform or operating system.

Unix emulators running on a windows platform to enable the execution of unix-based programs in a windows-based environment are good examples.
Testing Against Simulators or Emulators

Simulator Testing Pros:

Enable testing prior to devices being ready

Very good for early testing of user experience or paradigm testing

Can provide all of the benefits of software prototyping, including generation of test ideas

Simulator Testing Cons:

Not a valid test of actual software to be used in production

Can give a false sense of security

Usability and paradigm testing results may be invalid if executed on a significantly different platform
Testing Against Simulators or Emulators

Emulator Testing Pros:

- **Test actual production code**
- Typically reveal many of the same defects as testing on the actual device
- Frequently enable test automation, increase test coverage and enable scenario based testing

Emulator Testing Cons:

- **Not a valid test for performance**
- Will not reveal defects specific to device hardware or configuration
- Usability and paradigm testing results may be invalid if executed on a significantly different platform
Back-End Testing via Internet

Emulator wireless traffic

Internet

Wireless gateway

Standard TestStudio agents

TestStudio master station

Ethernet

Actual wireless traffic

Radio tower

Internet phone

Wireless PDA

Content or application server
Back-End Testing via Internet

Pros:

Indistinguishable from actual device testing from back-end components (when done correctly)

Generally easy to automate via emulator

Good for performance testing

Does not require cellular/satellite networks to be operational

Cons:

Does not test the actual device or how software will perform on the device

Not good for usability, paradigm or non-internet based features.
First Hand Experiences (Case Studies)

Testing the Inet API on the RIM Blackberry

Testing the ESPN Mobile Phone

Testing Microsoft's IPTV solution
Contact Info

Scott Barber
Chief Technologist
PerfTestPlus, Inc.

E-mail: sbarber@perftestplus.com
Web Site: www.PerfTestPlus.com