Risk-Based E-Business Testing

Paul Gerrard

Systeme Evolutif Limited
email: paulg@evolutif.co.uk
http://www.evolutif.co.uk/

Paul is the Technical Director and a principal consultant for Systeme Evolutif. He has conducted consultancy and training assignments in all aspects of Software Testing and Quality Assurance. Previously, he has worked as a developer, designer, project manager and consultant for small and large developments using 3 and 4GLs. Paul has engineering degrees from the Universities of Oxford and London, is Special Projects Secretary for the BCS SIG in Software Testing, a member of the BCS Software Component Test Standard Committee and Former Chair of the IS Examination Board (ISEB) Certification Board for a Tester Qualification whose aim is to establish a certification scheme for testing professionals and training organisations. He is a regular speaker at seminars and conferences in Europe and the US, and won the ‘Best Presentation’ award at EuroSTAR ‘95.
Agenda

- E-Business Testing Challenge
- Overview of E-Business Test Strategy
- Tools for E-Business Testing

Some definitions courtesy of www.whatis.com

- E-Commerce: the buying and selling of goods and services on the Internet, especially the web...
- E-Business: the conduct of business on the Internet which includes:
  - buying and selling plus servicing customers (internet)
  - collaborating with partners (extranet)
  - internal work and information flow (intranet)
- E-tailing (direct selling via the Web)
- E-bah-gum - a Northern expression.
Survey of 190 testers on 25/2/2000

- Who's using the web regularly? 100%
- Who's bought online? 50%
- Who's built an E-C or E-B system? <5%
- Who's planning E-C? 25%
- Who's planning E-B? 30%
- Who's experienced problems when using a web site? 100%

What kind of problems have you seen?

- Have you abandoned a site because it was:
  - difficult to use?
  - too slow to respond?
  - you didn't trust it to be secure?
  - the web pages broke?
  - did not process your order correctly?
The E-Business Testing Challenge

Web time

- ‘Web time passes 5-7 times as fast’
- Significant technology shifts every six months
- In many sites, web development done, managed by non-computer experts
- In an immature market, speed to market is first, quality comes second?
- Cultural, practical, technical and schedule obstacles to QA.
**Web site as a retail store**

- A Web site is just like a retail store
- Anyone can come to browse, to buy
- Open to all, including crooks and terrorists
- If your site gives good service, customers will come back again and again
- They won’t come back if
  - if the door is locked (site down)
  - service is slow (poor performance).

**No second chances**

- Customers, prospects won’t come back if:
  - your site isn’t available (24 x 7 x 365)
  - functionality doesn’t work
  - performance is slow
  - it’s difficult to use
- Many of the risks are outside your control
  - unlimited potential users
  - users hardware/software.
  - availability.
No control: client platform

- ‘Traditional’: PCs, Macintosh, Unix
- ‘Coming fast’: WebTV, WAP, PDAs, cars, fridges, anything with a chip in...
- Screen size and resolution
- Local character sets (may be rejected by your database)

No control: client software

- Any OS, any version, any condition...
- Browsers: IE, Netscape, Mosaic, HotJava...
  - 23 Windows compatible browsers (24 Jan 1999)
  - some non-graphic e.g. Lynx (Unix, DOS)
- Obscure browsers:
  - e.g. Cello (http://www.law.cornell.edu/cello/)
  - runs with only 2MB of RAM on a 386SX-16!
No control: client configuration (2)

- User may or may not have required plug-ins
- Cookies:
  - some may allow and ignore them
  - some want to be reminded
  - some will refuse them
- Network connections
  - LAN-based - high speed
  - telephone network and modem - slow
- Firewalls may get in the way.

Anyone can ‘walk in’

- Just like a real shop, open to the public
- Users may be expert, hackers even
- Users are unknown
  - anyone on the web a potential customer
  - we don't see them, they don't see us
  - our site is undocumented, they are untrained
  - if the site is hard to use, they’ll go elsewhere.
Anyone can ‘walk-in’ (2)

- Some may be under-age
  - can you tell if a 10 year old places an order with their parent’s credit card?
  - can you tell whether you are selling alcohol to minors?
- Internationalization e.g.
  - address formats
  - local currencies
  - local tax arrangements.

Our message...

- Risks of poor/missing/wrong functionality aren't necessarily the biggest issue
- Non-functional and Web specific issues are emerging in importance
- There’s a definite shift in product risks

- Why?
  - unfamiliarity breeds uncertainty (and may pass in time)
  - new risks because of the technology and potentially uncontrollable, infinite, unknown user base.
Typical, recent IT project (28 risks)

- Functionality
- Performance
- Security
- Usability
- Availability
- Operability

Functionality is the biggest concern

Recent E-Business project (63 risks)

- Functionality
- Portability
- Security
- Performance
- Highload
- Localisation
- Usability
- Interface
- Post-Deploy Nav

Functionality might not be the biggest concern
Main risk areas

- Functionality
- Reliability/availability
- Usability
- Performance
- Security.

Overview of E-Business
Test Strategy
Defining the test process

- Test stage definition:
  - distinct object under test
  - distinct objectives
  - as early as practical but minimise duplication
  - static test (analysis/review) or dynamic test
- Test strategy:
  - focus on risks, then identify the test types
  - aim at high level of automation.

Essential testing priorities

1. Smoke testing
   - Can the application survive one user session without crashing?
2. Usability
   - If your site is hard to use, the user won't hesitate to leave you
3. Performance
   - The Web server is usually the bottleneck
4. Functionality
   - This isn't the main issue - Web applications are not feature rich
Essential testing priorities (2)

1. Smoke testing ● If you are under severe time pressure, stick to smoke testing
2. Usability ● If you have time, usability testing addresses the next risk
3. Performance ● Performance tests flush out weak points, measures limits
4. Functionality ● Functionality is often simple, so may be the least of your worries

Automation-focused approach

● Our testing approach must therefore be:
  - plan tests once only and reuse
  - automate all tests and reuse
● Need to stop thinking that all tests start as manual tests to be automated
  - plan all tests to be run using a tool
  - developers habitually automate their tests
● Manual tests should be the exception.
Developer testing

- They should test using a variety of browsers
  - not just their favourite
  - not always the latest version
- They should write one automated test script for each transaction and run on each browser
  - these are your ‘essential’ regression tests
- Code+automated scripts checked-in together
- System testers use these tests as a start point.

Testing functionality

- Web applications are relatively simple
- Well designed applications
  - place minimal functionality on the client
  - maximum functionality on the server
- Back-end/legacy systems as complex as ever
- Functionality testing should focus on testing server-based code via the API
- Normally use a test driver and the API.
Test drivers

- Very simple programs which
  - accept data
  - execute software under test
  - store results
  - compare results with a benchmark file
- Proprietary tools exist, but consider writing your own in VB, Perl etc.

Test structure

- Static testing
- Test Browsing
- Functional Testing
- Non-Functional Testing
- Large Scale Integration.
Key to the test process table

- Tests can be static or dynamic
- Tests can be Automated or Manual
- Four broad stages of testing:
  - Desktop development testing (broadly, what the browser executes)
  - Infrastructure testing (what runs on the servers)
  - System testing (of complete system in isolation)
  - Large Scale Integration (with other systems)
- Post-deployment testing (site monitoring).

Test process table

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Test Priorities</th>
<th>Test Type Mapped to Usual Test Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static/Dynamic</td>
<td>Static Testing</td>
<td>Content Checking, HTML testing, Browser syntax compatibility, Visual browser validation, Test Browsing, Link checking, Object load and timing, Transaction verification, Browser page testing, CGI component testing, Transaction Testing, Application System Testing, Internationalisation, Configuration testing, Performance, Soak Testing/reliability, Availability, Usability, Security, Large Scale Integration, External links/legacy system integration, End to end functionality</td>
</tr>
</tbody>
</table>
Test process table (2)

- Not all test types are appropriate or possible
- Test types are grouped by risks
- Usual test stages for structured methodology
- Use ‘Test Priorities’ in a RAD environment
  - automated smoke test types are the high priority
  - functionality test types are low priority
- The table is a guideline to be considered and adapted - tailor it to your environment.

Test scope - browser support

- Which browsers will your software support?
  - what versions?
  - what about frames v no-frame browsers?
  - what about non-graphic browsers?
- Propose a scope for the testing to be performed (and the potential cost)
- This will help to force a decision on what will actually be supported.
Test scope - Web conventions

- Adherence to conventions is not mandatory, but can make a big difference to the user
- E.g. will your application support:
  - users turning off graphics?
  - users rejecting cookies?
  - cookies timing out?
  - user not having required plug-ins
- Need to establish what conventions apply.

Web accessibility initiative (WAI) guidelines (a work in progress)

A. Transform Gracefully
   - A.1 Provide alternative text for all images, applets, and image maps.
   - A.2 Provide descriptions for important graphics, images, or elements if they are not fully described through alternative text or in the document's content.
   - A.3 Provide instructions for transforming text and images for visually impaired users.
   - A.4 Ensure that text and images are accessible to visually impaired users through voice browser.
   - A.5 Ensure that text and images are accessible to visually impaired users through screen readers.
   - A.6 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.7 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.8 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.9 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.10 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.11 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.12 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.13 Ensure that text and images are accessible to visually impaired users through screen reader software.
   - A.14 Ensure that text and images are accessible to visually impaired users through screen reader software.

B. Orientation, Navigation, and Comprehension
   - B.1 Provide context and orientation information for complex pages or elements.
   - B.2 Provide mechanisms that facilitate navigation within your site.
   - B.3 Use language and formats that facilitate comprehension of information.

C. Appendix - Testing
   - C.1 Validate your pages and assess the accessibility with automated tools, manual tests, and other services.

D. Appendix - Definitions
   - D.1 Define terms and concepts that facilitate comprehension of information.
Tools for E-Business Testing

Web testing tools

- HTML validators
- Link checkers
- Test drivers for server based code
- Java test tools
- Browser test execution tools
- Load/performance test tools
- Security checking and auditing
- External site monitors

Comprehensive list of tools can be seen at: http://www.softwareqatest.com/qatweb1.html
Test process table

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Test Priorities</th>
<th>Test Types Mapped to Usual Test Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functionality Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static/Dynamic Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desktop Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infrastructure Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-Deployment Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitoring</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Test Priorities</th>
<th>Test Types Mapped to Usual Test Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content Checking</td>
<td>Y   S  A/M</td>
<td></td>
</tr>
<tr>
<td>HTML testing</td>
<td>Y   S  A/M</td>
<td></td>
</tr>
<tr>
<td>Browser syntax compatibility</td>
<td>Y   D   A  A</td>
<td></td>
</tr>
<tr>
<td>Visual browser validation</td>
<td>Y   D   M  M  M</td>
<td></td>
</tr>
<tr>
<td>Test Browsing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link checking</td>
<td>Y   D   A  A</td>
<td></td>
</tr>
<tr>
<td>Object load and timing</td>
<td>Y   Y  D   A  A</td>
<td></td>
</tr>
<tr>
<td>Transaction verification</td>
<td>Y   S  A/M  A/M</td>
<td></td>
</tr>
<tr>
<td>Functional Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browser page testing</td>
<td>Y   D  A/M</td>
<td></td>
</tr>
<tr>
<td>CGI component testing</td>
<td>Y   D  A/M</td>
<td></td>
</tr>
<tr>
<td>Transaction Testing</td>
<td>Y   D  A/M</td>
<td></td>
</tr>
<tr>
<td>Application System Testing</td>
<td>Y   D  A/M</td>
<td></td>
</tr>
<tr>
<td>Internationalisation</td>
<td>Y   D A/M A/M</td>
<td></td>
</tr>
<tr>
<td>Non-Functional Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration testing</td>
<td>Y   D  M  A/M</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Y   D  A A A A</td>
<td></td>
</tr>
<tr>
<td>Soak Testing/reliability</td>
<td>Y   D  A A A A</td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td>Y   S/D M</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>Y   D A/M A/M A/M A</td>
<td></td>
</tr>
<tr>
<td>Large Scale Integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External links/legacy system integration</td>
<td>Y   D  A/M A/M</td>
<td></td>
</tr>
<tr>
<td>End to end functionality</td>
<td>Y   D  A/M A</td>
<td></td>
</tr>
</tbody>
</table>

HTML Validators

- Validation of HTML against standards
- HTML browser compatibility checking
- Web-based tools can visit your site and check one page or the whole site
- Developer tools check pages individually or in batch
- Most HTML editors offer some degree of HTML validation.
Link checkers

- Tracing of in-page links to their targets
  - does the target exist?
  - Does the target load quickly?
- Web-based tools can visit your site and check one page or the whole site
- Developer tools check pages individually or in batch
- Most HTML editors offer some degree of HTML validation.

Test drivers for server based code

- Developers tend to write server based code before the user interface
- Need test drivers to exercise server based objects, servlets, CGI scripts
- A dummy page with hard coded buttons works, but is limited
- Simple drivers to send/receive HTTP requests may work fine.
Java test tools

- Mainly developer tools covering:
  - static analysis - complexity, flowgraphs
  - coverage analyser - statement, branch, method, classes, files, packages
  - dynamic analysis - debugger-like facilities to examine and monitor executing code
  - profilers - to monitor memory usage, threads, synchronisation problems, CPU and other resource usage
  - test drivers for applets/servlets
- Many sold as suites of tools.

Browser test execution tools

- Functional/regression test tools which execute applications through the browser interface
- Most are extensions to existing GUI capture-replay tools
- Most are sold as a component of a larger test suite
- Most can be used with performance test tools to capture response times seen by users.
Load/performance test tools

- Simulation of many simultaneous users
- Workloads represents different mixes of transactions, volumes of users
- Proprietary tools have most sophisticated:
  - script languages
  - load regulation
  - analysis facilities
- Freeware tools can simulate loads, but are less controllable/flexible.

Security checking and auditing

- Network scanner/audit of firewalls, web servers, routers, intranets
  - problem identification, potential impact, proposed counter measures, vendor patch installation
- Server O/S and web software audit against databases of known vulnerabilities
  - application checks, CGI, development product checks
- Password security checking
- User management and account security checking.
External site monitors

- External services scheduled to check out your site
  - site availability
  - transaction verification
  - page availability
  - and transaction response time
  - security audit
- Monitoring can be scheduled to run daily or every 5 minutes
- Alarms dispatched by emails, fax, pager
- Periodic reporting can highlight trends.

Proprietary tools

- Tool types requiring large and complex functionality
  - all functional/GUI test execution tools
  - some load/performance tools
- What you usually get with proprietary tools:
  - comprehensive functionality
  - better levels of usability
  - slick, helpdesk support
  - big prices.
Shareware tools

- Characteristics of shareware vendors
  - an individual or small development team
  - no direct sales/marketing
  - technically advanced, but lacking a ‘history’
- Characteristics of shareware tools
  - technically advanced, limited functionality
  - standalone products/novel products
  - limited support by possibly the developer
  - low prices.

Freeware tools

- Freely downloadable, at your own risk
- Freeware status not necessarily an indicator of usefulness or quality
- Some products are cut down versions of shareware/proprietary products
- Some free to use for low volumes of use:
  - e.g. HTML validators that have a page limit
- Potentially tiny user base, but personal support by the programmer.
Risk-Based E-Business Testing

www.evolutif.co.uk
Please register

Copies of this presentation are available on the resources page.