

# E-Business Testing

Paul Gerrard

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



Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 1

## Agenda I

- ◆ Internet, E-Commerce, and E-Business Overview
- ◆ Web Time
- ◆ Web risks
- ◆ The E-Business Testing Challenge
- ◆ Internet Test Strategy
- ◆ Tools for E-Business Testing.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 3

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 4

## Survey of 190 testers on 25/2/2000

	Percentage
◆ Who's using the web regularly?	<input type="text" value="100"/>
◆ Who's bought online?	<input type="text" value="50"/>
◆ Who's built an E-C or E-B system?	<input type="text" value="&lt;5"/>
◆ Who's planning E-C?	<input type="text" value="25"/>
◆ Who's planning E-B?	<input type="text" value="30"/>
◆ Who's experienced problems when using a web site?	<input type="text" value="100"/>

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 5

## 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?

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 6

‘Web Time’

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 7

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 8

## Web time (2)

- ◆ Software development is out, Web publishing is in
- ◆ Where does content end and code begin?
- ◆ ‘Hold the front page!’
  - hourly release of content
  - daily releases of minor fixes
  - weekly releases of upgrades.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 9

## Web Risks

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 10

## 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).

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 11

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 12

## No control: client platform

- ◆ 'Traditional': PCs, Macintosh, Unix
- ◆ 'Coming fast': WebTV, Mobile Phones, PalmPilots...
- ◆ Screen size and resolution
- ◆ Local character sets (may be rejected by your database)

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 13

## No control: client software

- ◆ Any OS, any version, any condition...
- ◆ Browsers: IE, Netscape, Mosaic, HotJava...
  - <http://browserwatch.internet.com/> lists 35 browsers!
  - 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!

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 14

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 15

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 16

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 17



## Main risk areas

---

- ◆ Functionality and integration
- ◆ Reliability/availability
- ◆ Usability
- ◆ Performance
- ◆ Security.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 18

## The E-Business Testing Challenge

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 19

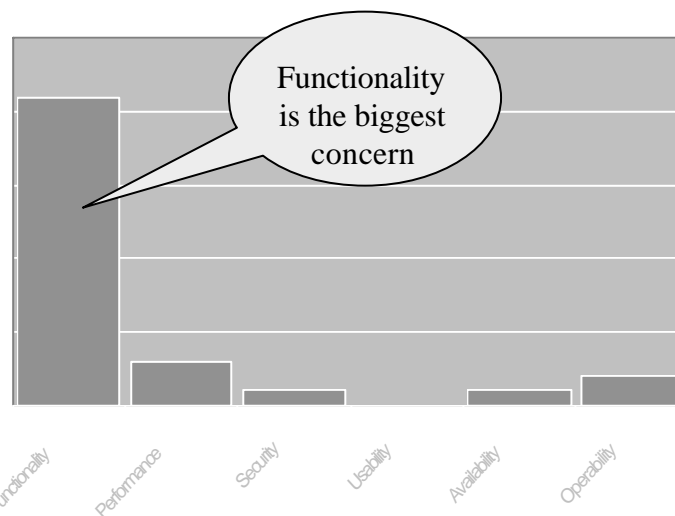
## Our message...

- ◆ 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 20

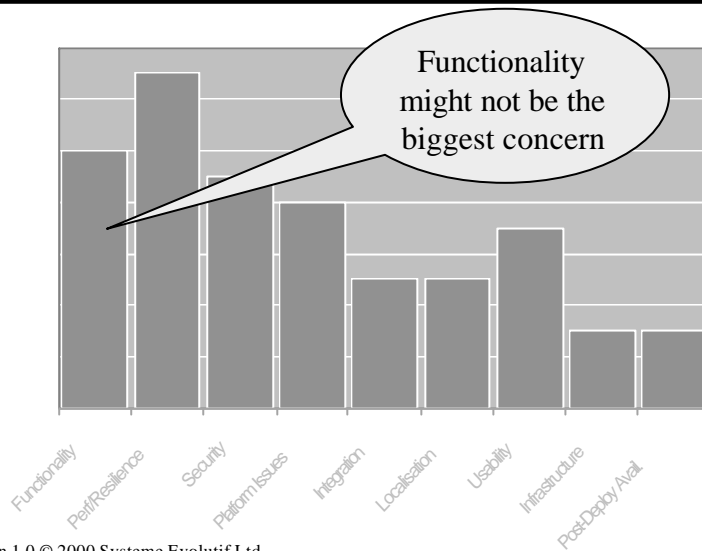
## Typical, recent IT project (28 risks)



Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 21

## Recent E-Business project (63 risks)



Slide 22

## Internet Test Strategy

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 23

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 24

## 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

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 25

## 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

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 26

## Testing under impossible deadlines

- ◆ If testing is to keep pace with development
- ◆ Need to squeeze the testing into the develop/release cycle
- ◆ Tests designed and automated up-front
- ◆ Testing tied into developer check-in procedures.
- ◆ Too many tests to run in time, so we must automate.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 27

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 28

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 29

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 30

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 31

## Test structure

- ◆ Static testing
- ◆ Test Browsing
- ◆ Functional Testing
- ◆ Non-Functional Testing
- ◆ Large Scale Integration.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 32

## Key to the test process table

- ◆ Tests can be static or dynamic
- ◆ Test can be Automated or Manual
- ◆ Four broad stages of testing:
  - Desktop development testing (broadly, what 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).

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 33



# Test process table

Test Type	Test Priorities					Test Types Mapped to Usual Test Stages				
	Smoke	Usability	Performance	Functionality	Static/ Dynamic	Desktop Development	Infrastructure Testing	System Testing	Integration Testing	Post- Deployment Monitoring
<b>Static Testing</b>										
HTML testing	Y			S		A/M				
Browser syntax compatibility	Y			S		A				
Visual browser validation	Y			D		M		M		M
<b>Test Browsing</b>										
Link checking	Y			D				A		A
Object load and timing	Y	Y	Y	D				A		A
Transaction verification	Y			S		A/M		A/M		
<b>Functional Testing</b>										
Browser page testing	Y			D		A/M				
CGI component testing	Y			D			AM			
Transaction Testing			Y	D				A/M		
Application testing			Y	D				A/M		
Internationalisation		Y		D		A/M		A/M		
<b>Non-Functional Testing</b>										
Configuration testing	Y			D		M		A/M	M	
Performance			Y	D			A	A		A
Soak Testing/reliability	Y			D		A	A	A	A	A
Availability				D						A
Usability		Y		S/D				M		
Security			Y	D			AM	AM	AM	A
<b>Large Scale Integration</b>										
External links/legacy system integration			Y	D			AM		AM	A
End to end functionality	Y			D				AM	AM	A

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 34

# 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 35

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 36

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 37

# 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, scripts, or applets if they are not fully described through alternative text or in the document's content.
- A.3 Provide textual equivalents (captions) for all audio information.
- A.4 Provide verbal descriptions of moving visual information in both auditory and text form (for movies, animations, etc.).
- A.5 Ensure that text and graphics are perceivable and understandable when viewed without color.
- A.6 Indicate structure with structural elements, and control presentation with presentation elements and style sheets.
- A.7 Provide supplemental information needed to pronounce or interpret abbreviated or foreign text.
- A.8 Ensure that tables have necessary markup to be properly restructured or presented by accessible browsers and other user agents.
- A.9 Ensure that pages using newer technologies will transform gracefully into an accessible form if the technology is not supported or is turned off.
- A.10 Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or frozen.
- A.11 Elements that contain their own user interface should have accessibility built in.
- A.12 Use features that enable activation of page elements via input devices other than a pointing device (e.g., via keyboard, voice, etc.).
- A.13 Use interim accessibility solutions so that assistive technologies and older browsers will operate correctly.
- A.14 Wherever possible use a W3C technology in accordance with guidelines on its proper use. Where this is either not possible, or results in material that does not transform gracefully you must provide an alternative version of the content that is accessible.

## 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

[Wwww.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth.html](http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth.html)

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 38

# Tools for E-Business Testing

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 39

## Test process table

Test Type	Test Priorities					Test Types Mapped to Usual Test Stages				
	Smoke	Usability	Performance	Functionality	Static/ Dynamic	Desktop Development	Infrastructure Testing	System Testing	Integration Testing	Post- Deployment Monitoring
<b>Static Testing</b>										
HTML testing	Y			S		A/M				
Browser syntax compatibility	Y			S		A				
Visual browser validation	Y			D		M		M		M
<b>Test Browsing</b>										
Link checking	Y			D				A		A
Object load and timing	Y	Y	Y	D				A		A
Transaction verification	Y			S		A/M		A/M		
<b>Functional Testing</b>										
Browser page testing	Y			D		A/M				
CGI component testing	Y			D			AM			
Transaction Testing			Y	D				A/M		
Application testing			Y	D				A/M		
Internationalisation		Y		D		A/M		A/M		
<b>Non-Functional Testing</b>										
Configuration testing	Y			D		M		A/M	M	
Performance			Y	D			A	A		A
Soak Testing/reliability	Y			D		A	A	A	A	A
Availability				D						A
Usability		Y		S/D				M		
Security			Y	D			AM	AM	AM	A
<b>Large Scale Integration</b>										
External links/legacy system integration			Y	D			AM		AM	A
End to end functionality	Y			D				AM	AM	A

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 40

## 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>

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 41

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 42

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 43

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 44

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 45

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 46

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 47

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 48

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 49



## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 50

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 51

## 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.

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 52

## E-Business Testing

End

Version 1.0 © 2000 Systeme Evolutif Ltd

Slide 53