COMP6205: Web Development

Dr A. Rezazadeh, Reza(ra3@ecs.soton.ac.uk)

November 17
Topics that will be covered

• Web Application Testing and Deployment
  – Classification and management of web tests and automation tools
  – Web site hosting

• Other Approaches to Web Development
  – MEAN Stack Approach
  – Serverless Approach
Topics that will be covered

• **Web Information Architecture and Content Management Systems**
  – Content Management Systems such as Sharepoint and Drupal
  – Intranet search techniques

• **Performance modelling and management**
  – Partitioning and parallelism, Amdahl’s law
  – Performance modelling and benchmarking

• **Coursework and ASP.NET related questions should be directed to Andy Gravell.**
COMP6205: Web Development

Testing Web Applications
What is Web Testing?

- “Web testing” is the name given to Software Testing that focuses on testing the web applications.
- This could help to address the issues in web application before exposed to public.
Functional Testing

• In Functional testing we need to check whether each component is functioning as expected or not.
  – so it is also called as “Component Testing”.
  – it is to check the basic functionality mentioned in the functional specification document.
  – Also check whether software application is meeting the user expectations.
  – We can also say that checking the behavior of the software application against test specification.
What Functional Testing includes?

• **Link Testing:**
  – Check that all links are working correctly and there is no broken links.

• This includes:
  – Internal links
  – Outgoing links
  – MailTo Links
  – Anchor Links
What Functional Testing includes?

• **Web form testing**
  
  – In the web application testing the “Form Testing” is the essential part of testing any web site.

  – The main purpose of form testing is to check that the information from user is received and store accurately into the database.
Web form testing

- Data Validations on each form fields. – “Client side” & “Server side” validations.
- Checking default values are being populated
- Check for all Mandatory fields.
- Check if a user not entered a required field showing a mandatory error message.
- Add information using form & update information using form.
- Tab orders.
- Forms are optimally formatted for better readability
What Functional Testing includes?

- **Cookies testing:**
  - Testing the application by disabling the Cookies
  - Testing the application after corrupting the cookies.
  - Check the behavior of application after removing the all the cookies for the website you are testing.
  - Check if cookies for authentication are working or not.
  - Check the behavior of application after deleting the cookies (sessions) by clearing cache or after cookies expired.
What Functional Testing includes?

- **Testing HTML and CSS:**

  - For the purpose of the search engine optimization testing of HTML and CSS are very important.

  - In this testing you should check that different search engines can crawl your site without any error.

  - You should check for all Syntax Errors, check for Compliance with Standards such W3C, ISO, ECMA, IETF, WS-I, OASIS
Automated User Interface Testing

• Web app logic is shifting from the server to the browser side.
  – As your app become more mature product, the interface code will become more elaborate,
  – and manual tests more cumbersome.
  – The answer then may be automated UI tests, which share the same assertion principle as unit tests.
  – In the case of UI tests, the conditions are established through a number of virtual mouse clicks, form interactions and keystrokes that simulate a user’s interaction with the app.
User Interface Testing – Tools

- For small team of developers who are intimately familiar with the interface code may prefer the strictly code-oriented approach of a tool such as Watir:
  - [http://watir.com/](http://watir.com/)
  - For larger teams or apps with particularly dynamic interfaces, the graphical test recording of Selenium may be better suited.
    - [http://seleniumhq.org/](http://seleniumhq.org/)

- Both tools support automated tests on multiple platforms and browsers.

- Create tests for critical workflows like user registration and user login – verify paths through the UI
What Functional Testing includes?

• **Testing business workflow:**
  
• This would include:
  
• End to end testing of business scenarios/workflow which ensure that the completeness of website testing.
  
• Testing of Positive as well as negative scenarios.
Usability Testing

- **Usability testing** is a way to see how easy to use something is by **testing** it with real users.

- Users are asked to complete tasks, typically while they are being observed by a researcher, to see where they encounter problems and experience confusion.
Usability Testing

- Correct navigation should be there between web pages.
- Site map should be there.
- User friendliness to all types of users, from novice to expert.
- Condition hold for physically challenged people.
- Web pages navigation should be Easy to use.
- Help instruction content should be clear & should satisfy the purpose.
Accessibility Tests

• Do images, videos and audio files have accurate text alternatives?
• Do form controls have relevant labels or titles?
• Is the content marked up with the most appropriate semantic HTML, including form and table elements?
• Does the app reset the focus to the new content after an Ajax content update?
• Do colour and contrast choices allow the text to be read easily?
• If colour is used to convey information (in required form fields, for instance), is the same information also available in text format?
• Can all of the app functionality be accessed using the keyboard alone?
Accessibility Tests– Cont.

- Accessibility tests are offered by a number of free online web services,
  - for example WAVE: http://wave.webaim.org/
  - for a faster test-fix-test workflow, you may prefer to use a browser plug-in, like the Firefox Accessibility Extension:
    - http://www.accessfirefox.org/Firefox_Accessibility_Extension.php
Compatibility Testing

• In software testing, compatibility testing is the non-functional part of testing.

• It is to ensure that application is working properly in the supported environments.

• Customers are using different web application on different operating systems, browser compatibility, hardware capacity, databases and bandwidth capacity of the network.

• Is web application renders correctly across different devices?”
Compatibility Testing – Tools

1. BrowserStack
2. Browsershots
3. Browserling
4. Litmus
5. IE NetRenderer
6. Browser Sandbox
7. TestingBot
8. BrowseEmAll
9. Multibrowser
10. CrossBrowserTesting
11. equafy
12. Sauce Labs
Web Security Testing

- Security testing is one of the most important types of software testing that make sure that there are no major vulnerabilities or weaknesses in the system.

- The main objective of security testing is to determine that its data and resources are protected from possible intruder and confidential data stays confidential.
Major Aspects of Web Security Testing

• Penetration Testing
  – Is is an authorized simulated attack on a system that looks for security weaknesses, potentially gaining access to the system's features and data.

• Password cracking

• URL manipulation

• SQL injection

• Cross Site Scripting (XSS)
  – XSS enables attackers to inject client-side scripts into web pages viewed by other users.
Security Testing – Tools

- Skipfish

- ratprox

- Burp Intruder
  - [http://portswigger.net/burp/intruder.html](http://portswigger.net/burp/intruder.html)

- These tools should be used in conjunction with data from the attack pattern database fuzzdb:

- The Open Web Application Security Project (OWASP) provides a comprehensive Testing Guide
Performance Testing

- Performance testing is to check the web application is working under the heavy load and response time stay within the accepted timing boundaries.

- Performance testing is categorized into two parts:
  - **Web Load Testing**: a planned test to check the functionality of the system under specific levels of simultaneous requests.
    - The goal of a load test is to prove that a system can handle the expected volume with minimal degradation of acceptable performance.
  - **Web Stress Testing**: simultaneous requests on a system increased to the point where performance is degraded, possibly even to the point of complete failure.
Performance Testing

- Performance tests measure typical response times for the app:
  - how long do the key pages and actions take to load for a single user?

- You’ll first need to configure your database server and web application server with profilers to capture timing information.
  - for example with Microsoft SQL Server Profiler (SQL Server), MySQL Slow Query Log (MySQL), dotTrace for (.NET), or XDebug with Webgrind for (PHP).
  - The Yahoo! YSlow profiler highlights similar problems in front-end code.
Load Tests

• Load tests simulate the expected load on the app by automatically creating virtual users with concurrent requests to the app.

• Load is normally incremented up to the maximum expected value to identify the point at which the application becomes unresponsive.

  – For example, if an app is expected to serve 100 users simultaneously, the load test might begin at 10 users, each of whom make 500 requests to the app. The performance will be measured and recorded before increasing to 20 users, who each make 500 requests, and so forth.
Load Tests - Benefits

- New bottlenecks may appear in your web app profiling hat didn’t surface in the simpler single-user performance test.

- These issues may highlight a need for caching, better use of file locking, and so on.

- Additionally, load tests can identify hardware limits or problems with server resources, like memory, disk space and so on.

  – You can use tools such as like **Top** (Linux)
  – or Performance **Monitor** (Windows).
Load Tests – Other Tools

• Free load testing software packages include:
  • ApacheBench
    – http://httpd.apache.org/docs/2.0/programs/ab.html
  • Siege
    – http://www.joedog.org/index/siege-home
  • Httperf
  • and the more graphical Jmeter
    – http://jakarta.apache.org/jmeter/
  • and The Grinder
    – http://grinder.sourceforge.net/
Stress Tests

• A stress test evaluates the graceful recovery of an app when placed under abnormal conditions.

• To apply a stress test, deliberately remove resources from the environment or overwhelm the application while it is in use:
  – Use a load test tool to simulate an unsupportable volume of traffic.
  – Create a large temporary file that fills the available disk space.
  – Restart the database server.
  – Run a processor-intensive application on the web server.
Stress Tests

- When the resources are reinstated the application should recover and serve visitors normally.

- More importantly, the forced fail should not cause any detrimental data corruption or data loss, which may include:
  
  - Incomplete cache files that are mistakenly processed or displayed when the app recovers.
  
  - An incomplete financial transaction, where payment is taken from the customer but their order is not recorded.
Database Testing

• Data reliability and integrity are the key part in the Database testing. So for web application should be thoroughly tested. Testing should include:

  – If queries are executed without any errors.
  – Creating, updating or deleting data in database should maintain the data integrity.
  – time needed to execute the queries, if required tune the queries for better performance.
  – load on database while executing heavier queries & check the result.
  – Querying data from database is represented correctly on the web pages.
Components’ Interface Testing

• In the Interface testing mainly three areas should be covered:
  – Web Server, Application Server and Database Server.

• Ensure that all the communications between these all servers should be carried out correctly.
  – Verify that if connection between any servers is reset or lost then what is happening.
  – Check if any request interrupts in-between then how application is responding.
  – On returns of any error from web server or database server to application server they should be handled properly & display such errors to the user.
Crowd Testing

• You will select a large number of people (crowd) to execute tests which otherwise would have been executed a select group of people in the company.

• Crowdsourced testing is an interesting and upcoming concept and helps unravel many a unnoticed defects.
Testing Strategy

• If your app is used in a hospital to prescribe medication doses, or it is a critical financial component in a large enterprise, don’t skimp on the tests.

• If you building a new web app that doesn’t impact human safety or hundreds of jobs, then follow the minimum viable product (MVP) approach.
  – you will do the bare minimum necessary to get something out the door as quickly as possible to test the waters.
Some types of test require greater investment than others before they pay off.
Test-Driven Development

• Under the *test-driven development* approach, each test is written prior to the functional code.
  – This to define the expectations of the function. When it is first run, the test should fail.
  – The developer then writes the minimum amount of code necessary to satisfy the test.

• Confidence in the successful test enables the developer to iteratively re-factor improvements to the code.
  – You can make it the initial scaffolding from which the app is developed.

• Hundreds of frameworks are available.
  – The Wikipedia list is a great place to start.
Summary

• Tests and deployment options come in many shapes and sizes; start with critical checks to your core features and gradually expand your test infrastructure as your app features stabilise and your user base grows.

• Create a test plan of primary and secondary features and paths.

• Run through the test plan with your team.

• Implement unit tests for critical functions and complex business logic.

• Implement automated interface tests to test paths through the interface.
Summary

• Build a profile of your target market’s browser usage.
• Use local virtualised browsers for compatibility testing.
• Test for accessibility issues, including alternative text, appropriate semantic mark-up, use of colour and keyboard controls.
• Profile your database and web server during normal use.
• Load test to identify resource issues and bottlenecks.
• Stress test to assess graceful recoverability.
• Perform manual code reviews.
• Run automated penetration tests.
• Develop an automated build and deployment process.
Next Stage – Deployment

- Deployment is moving a website from a local environment to live servers.

- What seems like a simple thing can actually be quite complex.

- There are absolutely loads of ways to go about it. They range from user friendly software and services, to more complex command line tools, to full blown systems with lots of moving parts.