Web Testing with Performance

Will have what to consider when testing Websites and Some tools to be used.

# Web Testing Considerations

What should be take into account when testing websites

* Visual Testing: Does it look nice? Does it conform to the company/brand. Text boxes within margins etc.
  + Colour Schemes
  + Logical Layout
  + Wireframe
* Cross-Browser/Device Testing
* Functional Testing: Certain functions need user input with acceptance test
  + Form Fields: Do they work as intended/ Validate Emails/Boundary Value Analysis/1000 characters or 1 characters in the address form.
  + Character limits and checks
  + Broken Links
  + Broken Buttons
  + Error Messages and notices/ Help messages
  + Data Integrity
* Accessibility Review: Demographic – Young/Old. Disabled?
  + WAI-ARIA
  + BS8878:2010
  + WACG
  + Company/Industry standards
  + Colour blindness
* Security Testing
  + Know if were using HTTPS (passwords/credit card info)
  + Not storing passwords in plain text
* Code Validity
  + HTML and CSS Validators (pinch of salt)
  + Link Checkers (checks for dead links)
  + **Apply common sense**
    - <http://validator.w3.org/>
    - <http://jigsaw.w3.org/css-validator/>
* Usability: Are user flows optimal?
* Stress/Load Testing
* Performance Testing

# Performance Testing

Relates to the speed of a website upon loading. How quickly can we load the page/elements/ and the interaction of the user with the page

## Web Performance Tools

* Pingdom
* (firefox plugin) YSlow

## Factors affecting performance

* Page size
  + Use page weight to reduce e.g. Minify Code, Compress Images. Gzip Assets etc
* Slow User internet speed
* Connection
* Number of users
* Server Location
* Caching
* Database efficiency

# Stress vs Load Testing

* Load Testing: Modelling the expected usage by simulating multiple users accessing the programs services concurrently.
* Stress Testing: Determine the stability of a given system testing it beyond normal operation capacity
* Reliability Testing: Determines how long the application can sustain optimum performance
* Volume Testing: Testing software application for a certain data volume.

Stress Testing

* What’s the maximum load until a system breaks down?
* How is the system breaking?
* Is the system able to recover?
* When under stress, in how many ways can the system break, and where are the weak points? (knock on effects)

Volume Testing

* Does the system break when handling a large volume of data
* Is its performance affected

Load (or capacity) Testing

* Will the system support the anticipated load?
* Will the system support the expected peak load?
* Is the system capable of handling increasing load over time?
* What additional resources are needed to support additional load?

Reliability & Recovery Testing

* Is the system able to recover from breakages
* How often does the system break (analysis of docs, static testing)

Load Testing

* Expected normal load
* Expected Peak load
* Software KPI’s
* Tools to run test
* (look at own history/competitor data)

**MUST HAVE METRICS FOR IT TO BE VIABLE.**

Example:

* The image loads fast = Wrong
* The image loads under a second under normal load = Correct

**GOT TO BE TESTABLE!!**

# Load Testing Tools

* HP Loadrunner
* Flood.io
* JMeter