Web Development

Success

DEUCENG
Dokuz Eylül University
Dept of Computer Engineering

CME4414 Advances in Web Technologies

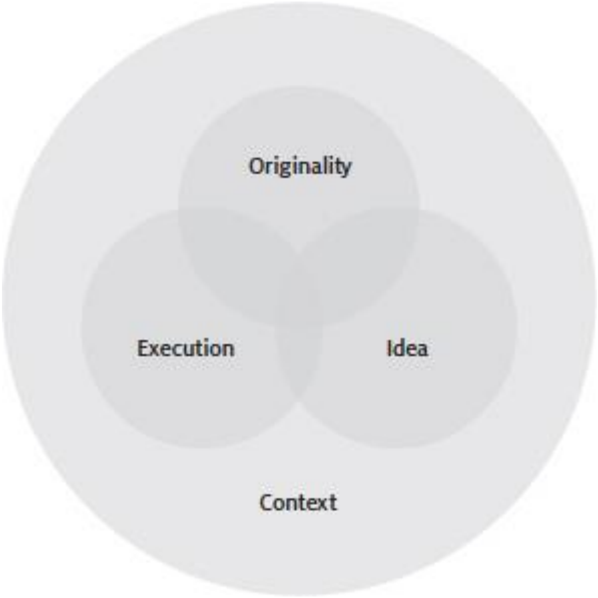
Web App

- Elements of Success
- Project Management
- Market Research - Strategy
- Interface
  - Complexities of designing for the Web
  - Prototype & User Test
- Development
  - Web technology fundamentals
  - Rapid Development
  - Security
  - Performance
  - Test & Deployment
- …..

Elements of Success

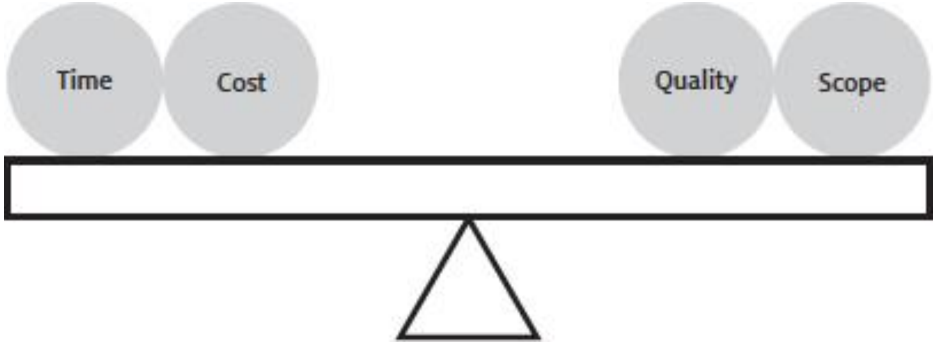
There are four interrelated attributes of a web app to consider

- The idea; what the app does
- The originality of the app; both as an idea and in implementation
- The quality of execution
- How it fits into the wider context of web technologies



Project Management

- Reducing the timescale of a project almost always affects the quality or the scope.
- Spending less money on a project typically results in removing features from scope rather than reducing the time.
- An increase in desired quality almost always demands an increase in timescale or cost, rather than a reduced scope.



Complexities of designing for the Web

Clients/costumer will use assorted devices to access your web app,
&
consequently there are a number of technical factors outside your control that you should take account of throughout your web app design process:

- Connection speed, service provider and reliability (need to take potentially unreliable connections into account)
- Display size
- Pointing device (mouse, trackpad and touchscreens, each with different click capabilities and accuracy)

Complexities of designing for the Web

- Peripherals: speakers, microphones and web cams
- CPU and device performance
- Browser vendors and versions
- Plug-ins and media support
- User preferences: JavaScript support, cookies and window sizes

Prototype & User Test

A prototype test reveals useful insights into the effectiveness and potential of your app.

- List the elements on each page, group & prioritise
- Mock up low fidelity variations of interface elements(with pen and paper)
- Wireframe and prototype the key app interfaces
- Mock up high fidelity prototype interfaces with tool(from pen and paper to specialist mockup software)
- Test prototype before showing it to test participants

Prototype & User Test

- Decide what you want to measure before conducting the tests.
- Use scenario-based tests rather than specific, leading questions.
- Test participants should be relevant to the app
- Reward participants with a small token gift
- Record all test sessions
- As moderator of a test session, you should mostly listen and ask why choices are being made
- Capture notes immediately after a test session, and implement changes to the interface as soon possible after all sessions are complete

Web technology fundamentals

Knowledge of the underlying web technologies enables you to

- Develop workarounds for web browser restrictions
- Optimise performance
- Optimise security

Web technology fundamentals

- DNS converts domain names to computer-usable identification numbers.
- HTTP messages govern the requests and responses between web browsers and web servers
- HTTP is stateless, but cookies can be used to remember a computer from one request to another
- Content-type HTTP header fields tell the browser what type of content is being sent

Web technology fundamentals

- Character encoding headers tell the browser how to understand text files
- UTF-8 is the most practical character encoding for the web
- Web browsers convert HTML into a document object model (DOM) tree in memory
- A second render tree is created in browser memory from the DOM, to represent the visual page layout
- Use a DOCTYPE to tell the browser which layout mode to use
- JavaScript can modify the DOM, and Ajax techniques can request additional data from the server and make partial updates to the DOM

Rapid Development

- Many developers have a deep-rooted need to solve problems & they invest time in the minutiae of artful custom code
- This can be a desirable trait for the stability and longevity of established products, the unpredictability of **early-stage apps** demands a less fastidious mindset

- Create code in the easiest, fastest way possible
- Consider that all code might change or be thrown away
- Stick to core technologies that you already know
- Break down the app features into individual tasks
- Agree on nomenclature - terminology

Rapid Development

- Use libraries, frameworks and web services to speed up early development and neutralise browser inconsistencies

- Check the licence of any third-party code that you use

- The MVC architecture is suitable for many web apps

- Create code that is modularised, abstracted and loosely coupled (where possible)

Security

- The development process must also consider non-functional aspects of the application, the most important of which are **security and performance**

- Feature development is a relatively short-term investment that is iterated to retain customers or capture additional market share.

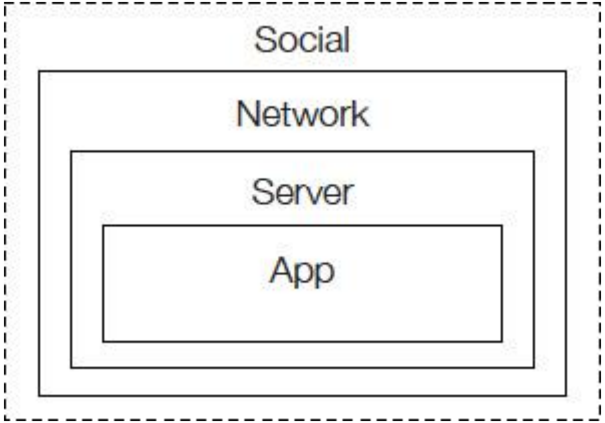
- Non-functional development is a long-term investment to enable growth, protect the business and reputation, and alleviate legal issues.

- It too requires frequent attention
- Not simply a checklist to mark off once at the start of a project

Web App Security

Attacks can be targeted at the app software, the server software, the network software, the hardware or even at the people who work on the app.

- Even the smallest web app faces attack from thousands of indiscriminate hacking tools
- Put basic security measures in place to protect your app and your users’ data



Security

- Everyone working on the app should be aware of social hacks and how they work
- Ensure that your hosting provider is secure and that it keeps its systems updated
- Keep all software, including third-party libraries, updated
- Configure your web and database servers to minimise risk
- Never trust user input. Assume that all input can be malicious
- Reject invalid input rather than trying to clean or convert it
- Never display raw user data to the screen: always sanitise it
- Use HTTPS for sensitive credentials, including login, registration and financial data
- Don’t let your users create weak passwords, and encrypt passwords

Performance

- A 2010 study by Jakob Nielsen found that
- “[s]lowness (or speed) makes such an impact that it can **become one of the brand values** customers associate with a site”
- If a user is forced to wait more than a few seconds for a page, they no longer feel in control of the experience and are likely to investigate alternatives

- On a more positive note, Nielsen also reveals that a **speed increase of as little as 0.1 seconds** can produce a noticeable lift in conversion rates
- 0.2 second delay decreased the number of searches by up to 0.36%, a significant number of customers for mass-market web apps (confirmed by a Google study, 2009)

Performance

Performance speed is a crucial element of a user’s experience of an app, (because of the uptake of web-enabled mobile devices that have bandwidth, memory and processor constraints)

- Optimise image files
- Remove unused code
- Minify text files
- Gzip text files
- Load larger files on demand
- Implement client-side validation
- Use UTF-8 characters rather than HTML entities
- Place CSS and JavaScript in external files
- Set cache HTTP headers
- Make Ajax cacheable

Performance

- Use a content delivery network
- Reduce cookie sizes
- Host static content on a cookie-less domain
- Minimise DNS lookups
- Enable Keep-Alives
- Combine files where possible
- Optimise database tables and indexes
- Optimise database queries
- Tune database settings
- Store frequently used data in an in-memory datastore
- Minimise the DOM size , Use the DOM efficiently
- Include style sheets in the <head> and JavaScript just before the </body>
- …….
- ….

Testing

Web app tests fall into five main categories:

- Functional tests: does the app work?
- Compatibility tests: does the app work consistently for everyone?
- Performance tests: does the app respond quickly and how does traffic affect performance?
- Security tests: is the app secure against attacks?
- Usability tests: is the app easy to use and does it respond to interaction as expected?

Tests and deployment

- Tests and deployment options come in many shapes and sizes
  - Start with critical checks to your core features
  - 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
- Build a profile of your target market's browser usage
- Use local virtualised browsers for compatibility testing

Tests and deployment

- Test for accessibility issues, including alternative text, appropriate semantic markup, 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
- Develop an automated build and deployment process
- …

Performance Testing

- Web app responsiveness can be evaluated through performance tests,
    - load tests
    - stress tests
- Practicality, starting with simple performance tests & hold off on the more exhaustive load and stress tests until you’ve gained some customers
- scalable cloud hosting platforms - performance optimisation

Load Test & Stress Test

- 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

- 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 (Restart the database server)

Scalability

- Scalability can be assessed through the ratio of the increase in system performance to the increase in resources used.

- Also, scalability means the ability to add extra resources while keeping the structure of the central node intact.

- The website scalability definition is the ability of a system, network, or process to cope with the increase in workload when adding resources (usually hardware).

Characteristics Affect

What characteristics affect the web application scalability?

- **Architecture :** This is one of the most important parts of app scalability
- **Framework load:** Sometimes scalability is limited to the framework, so its choice affects the performance of an app with the increase of features.
- **Sustainable design :** The quality of code also greatly affects scalability.
- **Sustainable load testing:** With proper load and performance testing, you’ll be able to find and eliminate the bottlenecks of your app and ensure its smooth growth
- **Hardware limitations:** not only software can affect scalability
- **Third-party component integration:** This is the most widespread cause of bottlenecks and failures in the operation

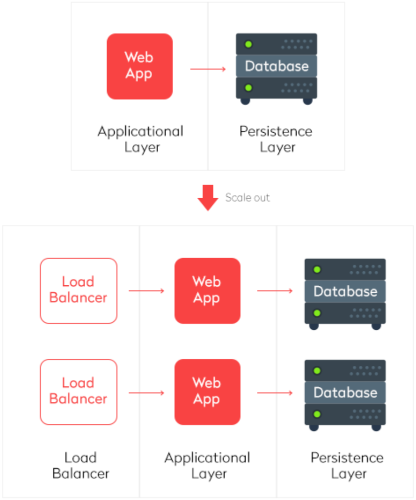
Frontend & Backend

- **A properly designed frontend** also helps avoid problems with scalability in the future
- Backend stands for everything that happens on the server
- It interprets everything the user does on the frontend, communicates it to the database if necessary and sends a response back to the browser.
- This part takes up to 80% of the entire code and affects the performance of the entire application the most.

Multi tier

A multi-tier model is a software model that should contain three components:

- the client
- the application server the client application connects to
- the database server to run the application



Horizantal Scaling

Horizontal scaling is when you separate the system into

- smaller structural components
- space it across individual physical machines
- increase the number of servers that **execute the same function simultaneously**

Vertical Scaling

- Vertical scaling implies a performance **increase of each system component** to improve overall performance

- Scalability in this context means the **ability to replace the components of an existing computing system with more powerful and faster ones** as requirements grow and technology develops

- This is the easiest scaling approach since it doesn’t require any changes to the application programs running on such systems

Scalable web app principles

- Smooth performance
- Constant availability
- Quick data retrieval
- Malfunction traceability
- Real-time response

Popular scalable web app frameworks

- Node.js
- Django
- Ruby on Rails

Search engine optimisation

SEO aims to optimise organic search rankings and click-through rates from search engines results pages.

- Choose target keywords based on search volume, trend, competition, relevance and commercial intent
- Optimise your link structure based on key pages
- Use single URLs for pages and domains
- Only indexable content counts towards SEO(include alt text and transcripts for media files)
- Tweak your URLs for keywords and neatness
- Choose page titles carefully (include target keywords)
- Don’t target the same keywords on multiple pages
- Create <meta> descriptions with target keywords for key pages to improve click-throughs
- Include a few variations of target keywords in the page content

References

- A Practical Guide to Web App Success,2011, Five Simple Steps , *by Dan Zambonini*
- https://justcoded.com/blog
- http://www.useit.com/alertbox/response-times.html
- http://googleresearch.blogspot.com/2009/06/speed-matters.html