Web Application Engineering

Jeff Offutt
http://www.cs.gmu.edu/~offutt/

SWE 642
Software Engineering for the World Wide Web

“Where is the wisdom we lost in knowledge?
Where is the knowledge we lost in information?”
– T.S. Eliot

Motivation – Overview

• Modern web applications are:
  – Distributed (world-wide)
  – Heterogeneous (hardware and software)
  – Highly user interactive
  – Built on new technologies

• The software is:
  – Very loosely coupled
  – Written in multiple languages
  – Often generated dynamically

Diverse: In terms of software, communication, and people
Motivation – Overview (2)

• Web application software has to be better than most shrink-wrap or contract software
• The combination of higher quality requirements and unique technologies make for a very interesting situation
  (Academics think “interesting” means fun, managers think “interesting” is scary …)

This talk discusses why and in what ways web software must be better

Software Deployment

• Bundled: On your computer when you buy it
• Shrink-wrapped: Bought at a store on a CD
  – Downloaded from company’s website or OSS site
• Contract: Single customer
• Embedded: Installed on an electronic device
• Web application: On the web through a URL
  – Component-based
  – Concurrent / distributed
  – One copy on the server
  – Can be updated at any time (fast update cycle)
  – User interactive
Formation of the WWW

The World Wide Web is a result of the *confluence* of three developments.

**Fast Computers** → **Internet** → **Hypertext Theory** → **World Wide Web**

Historical Highlights
Hypertext, Internet, and the Web

1945: Vannevar Bush proposes hypertext
1965: Ted Nelson coins the term “Hypertext”
   - Douglas Engelbart invents mouse
     - (The ACM Douglas Engelbart award is given yearly at the HyperText conference)
1968: Engelbart’s hypertext system “Augment/NLS”
   - Brown university’s HES
1969: Arpanet
1970: Unix
1972: Tomlinson invents email
1977: UUCP
1978: First bulletin board system – Christensen and Suess
Historical Highlights (cont.)
1979: Truscott and Bellovin’s usenet news
1982: < 100 sites on “internet” (netnews)
1983: TCP / IP defines the internet
1986: NSFNET
   SGML
1987: HyperCard (Apple)
   First hypertext workshop
1989: WWW proposed by Berners-Lee
1990: HTML defined
1992: CERN (Switzerland) releases WWW
1993: NCSA Mosaic

Historical Highlights (cont.)
1994: WWW surpasses gopher
1994: Netscape
1995: Lycos search engine
   Java
   DEC’s Alta Vista search engine
   15 million pages
1996: US Telecommunications Bill
1997: 31,000,000 pages, 476,000 servers, 14,000 news groups
2000: More than 100,000,000 hosts
   Amount of back-end programming exceeds the front-end hypertext
2004: 3,307,998,701 pages (google), 500,000,000 hosts,
   1,200,000,000 users
2013: 3.65 billion pages (www.worldwidewebsize.com)
The World Wide Web

The world wide web was invented by Tim Berners-Lee, an Englishman at CERN, the physics research lab in Switzerland.

Tim Berners-Lee’s vision of the world wide web was to make all information available to all people at all times.

General Hypertext Terminology

- **Node**: Atomic amount of text – smallest unit of information known to system (pages)
- **Link**: Connects one node to another. Depicted by buttons or highlighted region.
  - **Link Region (anchor)**: Location in a node where link starts.
- **Traverse (visit or view)**: Moving from one node to another
  - **Backtrack**: Following links in reverse
- **Landmark**: Semantically important nodes that are linked to many nodes
- **Hypermedia**: Incorporates sound, video, pictures, or other non-textual information
General Web Terminology

- **Web Page**: Data that fits in one browser screen
  - Static: HTML exists as a file on a computer
  - Dynamic: Created as needed

- **Web Site**: A collection of connected web pages

- **Web Application**: Program deployed on the web
  - UI is in HTML
  - User interacts through HTTP’s request / response cycle

Web Software Terms

- **HTTP**, HyperText Transfer Protocol: The network protocol used to connect to web applications
  - It is fast, lightweight, and connectionless

- **CGI**, Common Gateway Interface: The protocol used to transfer data from web browsers (client-side) to executable programs (server-side)

- **RDBMS**, Relational Database Management System: A type of database that uses tables of rows and columns based on SQL

- **DBA**, Data Base Administrator: Someone who manages the data, schema, and connections to a DB

- **SQL**, Structured Query Language: The primary language used to communicate with a databases
Modern Applications

Web applications have continuously evolved in the last two decades

We have moved from:
- ftp/email …
- to gopher …
- to simple html pages …
- to web sites …
- to dynamic html …
- to web commerce …

with amazing speed!

Modern Web Applications

- Web applications are too complicated for individuals to manage
- They need to be engineered by teams of people with diverse talents:
  - Programming skills
  - Graphics design
  - Usability
  - Information layout and engineering
  - Data communications
  - Data base

Web Application Engineering
Important Quality Attributes for Software in the 1990s

**Traditional**

1. Efficiency of process (time-to-market)
2. Efficiency of execution (performance)

50. Reliability
51. Safety
52. Maintainability
53. Security

Important Quality Attributes for Web Software

1. Reliability
2. Usability
3. Security

Customers have little “site loyalty” and will switch quickly, thus time to market is much less important than in other application areas.

4. Availability
5. Scalability
6. Maintainability
7. Performance & Time to Market

Based on an informal survey of around a dozen software development managers, 2000
Common N-Tier Architecture

Client-server ... 3-tier ... N-tier ...

Problems Can Occur Anywhere

- 1995: Web sites were 100% interface
- 1998: Web sites were about 90% interface
- 2001: Web sites are less than 50% interface
- 2005: Web applications about 25% interface
- 2013: Web application development dominates the software industry

There is a huge shortage of knowledgeable, skilled web programmers and software engineers
Issues in Engineering Web Apps

- Software development process for the web?
- Design modeling for web applications?
- Testing web applications?
- How to maintain and evolve web applications?
- Which design patterns and frameworks work, and when?
- Programming technologies:
  - 1998: HTML 3.2, Java 1.1, CGI, Perl, JavaScript, AWT, Active Server Pages, ...
  - 2001: HTML 4.0, Java 1.2, Java Servlets, EJB, XML, JSP, ...
    ⊗ More centered (around Java)
  - 2002: .NET
  - 2004-2013: Lots of frameworks and APIs

Summary
Changes in Web Technologies

1997
- Static web pages
- “Soft brochures”
- Webmasters
- HTML, CGI, JavaScript

1998-1999
- Dynamic HTML
- Programs (poorly written)
- Confused webmasters
- ASP, CSS, ...

2000-2004
- Web applications
- ECommerce +
- Web services
- WebManager + programmers, DB, network, UI, graphics designers, ...
- Java (JSP, Servlets, beans), {HT,U,X}ML, Component-based
- .NET

2004-2013
- Web application frameworks
- Web 2.0
- AJAX
- Widespread enterprise applications
## Summary
Concerns of Software

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Web Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Efficiency of process</td>
<td>1. Reliability</td>
</tr>
<tr>
<td>(time to market)</td>
<td>2. Usability</td>
</tr>
<tr>
<td>2. Efficiency of execution</td>
<td>3. Scalability</td>
</tr>
<tr>
<td></td>
<td>4. Security</td>
</tr>
<tr>
<td>50. Reliability</td>
<td>5. Availability</td>
</tr>
<tr>
<td>51. Safety</td>
<td>6. Maintainability</td>
</tr>
<tr>
<td>52. Maintainability</td>
<td>7. Performance &amp; Time to Market</td>
</tr>
<tr>
<td>53. Security</td>
<td></td>
</tr>
</tbody>
</table>

### Technology Predictions for 2020

- 1990: Command-based email, FTP
- 2000: Widespread use of web applications
- 2013: A web-based world

I am not foolish enough to make 10 year predictions about the web!
Summary

The concept of Web Application Engineering is just beginning …