Why Web Engineering Is Needed For Web Applications and Services

Bebo White
Stanford Linear Accelerator Center

March 10, 2005
My Basic Premises

- Developing a Web-based system is no longer an event, it is a process
- Web-based systems are growing systems
- Web-based applications are adding a whole new dimension to software development
The Web Started Simple


“It is now technically possible and feasible to put the entire creative works of man online” – Kahle, *The Internet Archives*

E-commerce (Web/Internet) accounted for 2% of US retail sales in 4Q, 2004 - $18.4 billion – US Census Bureau
1st Generation

Client-Side
- Web Application
- Browser / User Agent

Request

Response

Resource

Server-Side
- Web Application
- Web Server

Web System
Application protocol: HTTP

Browser
- Mosaic
- HTML
- Images (GIF)
- HTML-Forms
- Helper
  - Audio, Video etc.

Web System
- HTTP

Web Server
- HTTP
- CGI
  - Database
  - Information Systems
2nd Generation

Client-Side

- Web Application
- Browser / User Agent

Server-Side

- Web Application
- Web Server

Request → Response

Web System
Application protocol: HTTP

Browser
- Mosaic, Netscape
- HTML, Frames
- Images
- HTML-Forms
- Helper
  - Audio, Video
  - etc.

Web System
- HTTP
- Cookies

Web Server
- HTTP
- Server-API & CGI
  - Database
  - Information Systems
  - Media Server
Some Definitions are in Order

- **Web System** – an infrastructure or system enabling the operation of a Web Application
- **Web Application** – a distributed application that accomplishes a certain business need based on technologies of the World Wide Web and that consists of a set of Web-specific resources
Problems with Large Web-based Projects

(Source: Epner, M., Cutter Consortium, 7 November 2000)
Still Ad-hoc instead of a disciplined procedure
- Copy-and-Paste Paradigm
- Lack between design-model and implementation-model
- Design-concepts get lost in the underlying model
- Many short lifecycle of a Web Application: Maintenance and Evolution problems → Reuse Problems

→ Web-Crisis
Lacks rigor, systematic approach

- The completed system is not what the user wants
- System not developed on time, cost overruns
- Lacks scalability and maintainability, hence a limited useful life
- Does not meet performance requirements
- Resources are wasted
Web Systems: Problems

- Problems
  - Inability to maintain
  - Unable to meet evolving needs and grow at the rate needed – scalability
  - Unreliable – crashes
- Web-dependent organizations cannot afford to have
  - Faulty systems – reliability, security issues
  - Frequent downtime – dependability
  - Wrong, inconsistent, or stale content/information
- Web systems problems are not easy to hide
Many developers think that Web application development is just Web page creation using HTML, FrontPage, Dreamweaver, etc. with graphics design and/or simple hyperlinking. They have been taught to think this way! Certain classes of applications do fit this simple generalization – e.g., personal Web pages, event brochures, etc. Many other Web applications go beyond simple content presentation/navigation.
Web Development Issues (2)

- “There is more to Web application development than visual design and user interface”
- Planning, system design, testing, continual maintenance, quality assurance, performance evaluation, scalability,…
Consideration to Quality?

- Lack of consideration to:
  - Navigation
  - Accessibility
  - Scalability
  - Reliability
  - Maintainability
  - Usability
  - Compatibility and interoperability
  - Security
  - Readability
Web Development – Political Challenge

Anyone can be a Web content creator and maintainer!

- Business Unit
- IT Department
- Web Development Team
- Marketing
- Finance/Accounting
- Personnel
- Etc, etc

(Adapted from Susan Dart, 2002)
Web System Development: Summary I

- Less attention is given to development methodologies, testing and evaluation, quality assessment and control
- Largely relies on individuals’ own development practices
- Lack of realization of its lifecycle
  - Analysis of needs, redesign, development (including coding), management, metrics, maintenance
  - Calls for significant system-level and design decisions
- It is an exercise – not an event
- Legitimate concern about the manner in which they are created and their long-term quality and integrity
Web System Development: Summary II

In many cases, Web Development is:
- Chaotic
- Failure-prone
- Unsatisfactory
3rd Generation (Multi-Tier)

Client-Side

- Web Application
- Browser / User Agent

Request

Resource

Response

Server-Side

- Web Application
- Web Server

Web System
Application protocol: HTTP, SOAP, WebDAV, other

User Agent
- Netscape, IE, and PDA-Browser etc.
- Other Types of User Agent
- Plug-Ins, Applets, ActiveX
- Script-Code
- DHTML, More...

Web System
- HTTP, WebDAV, SOAP, other
- Cookies
- UDDI
- Other relevant protocols FTP, SMTP
- More...

Web Server
- HTTP, more
- Server-API & CGI
- XML-Support
- Component-Support
  - Servlets
  - Web-Services
SOA Functionality
- Composition Engine
- Federation, Security
- Transaction, etc.

Configuration/Context
- Components, End Points
- Semantic Web
- Policy, Permissions, etc.
Technology Drivers

- **Computing power**
  - Still doubling every 18 months
  - PC-based data centers

- **Connectivity**
  - Low cost, broad reach Internet
  - Wireless, broadband access

- **Device proliferation**
  - PDAs, cell phones, gas pumps
  - Towards a digital devices decade

- **Internet standards**
  - XML-based integration

- **User Interface**
  - Many (!) possibilities
Range of Complexity

- Static Site
- Data Collector Site
- Dynamic Data Access
- Web Application
- Service-Based Web Application and Federations
- Dynamically Created Site

Complexity Scale:
- Simple
- Complex

Application Centered vs. Document Centered

Characteristics

Today’s focus on large-scale and ubiquitously useable Web Applications

- Many Users – many languages – many cultures
- Different access mechanisms
- Many User Agents

Presents large volume of interrelated information (including different media) and processes

- Appropriate presentation
- Progression through activities – finish one thing before starting another
  Often guides user

Growing and increasing complexity

- Many product iterations/versions/refinements (calls for Reuse)
- Many developers and operators, complex handling of temporal media (e.g. publishing of company news)
- Customization, Personalization, Security issues

and a lot more...

- E.g. “Up-to-date” by following trends
Risks to Web-Based Business

- Long system delivery time
- Low responsiveness to business changes
- High project development and ongoing support costs
- Questionable system quality
Desires from Web-Based Business

- Project Delivery Time - *Shortened*
- System Quality - *Improved*
- Technology Investment - *Optimized*
Need for Process

- Domination of the different requirements calls for a systematic approach
- Producing high-quality Products in a cost-effective way
- Goal – Product should be
  - Maintainable and evolvable
  - Reliable
  - Efficient
  - Appropriate for User Interface (also wrt Hypermedia)
  - Delivered in time with predictable cost
Or simply Software Engineering?

“Fundamental differences [between hypermedia and other disciplines] however, make a pure transposition of techniques both difficult and inadequate. An important part of hypertext design concerns aesthetic and cognitive aspects that software engineering environments do not support.”

(Nanard and Nanard, 1995)
Web Development vs. Software Development

- They are different due to the nature and distinct requirements of Web-based systems.

- Even though Web-based systems often require programming and specialty software, the development of that software is often unique.
Web-based Systems vs Software I

- Web-based systems:
  - Are document-oriented containing static or dynamic content
  - More emphasis on "look and feel"
  - Are "content-driven" – process is driven by the availability of content
  - Need to cater to users with diverse skills and capabilities
  - Are typically constrained to a short development time, making it difficult to apply the same levels of formal planning and testing as used in software development
Web-based systems:
- Differ by means of their access and delivery medium
- Have different lifespans
- Have differing development requirements
  - Developers are vastly varied in terms of their background, skills, knowledge, and system understanding
  - Developers differ in their perception of Web systems
Web-based Systems vs. Software III

- Web-based systems:
  - Should be scaleable
  - Have varying performance requirements – must be able to cope with uncertain, random heavy demands on services
  - Must be secure
  - Are subject to various legal, social, and ethical scrutiny
Why Engineering?

- Engineering is associated with scale-up. New issues arise when scaling by 2+ orders of magnitude.
- Skills of architects and all kinds of engineers are different from those needed for designing and building simple structures and simple systems.
Web Engineering – is the application of systematic, disciplined, and quantifiable approaches to the design, production, deployment, operation, maintenance and evolution of Web-based software products. [Gaedke, 2000]
Key Knowledge Areas

- **Software Engineering**
  - Process
  - Design
  - Implementation
  - Test
  - Operation
  - Maintenance

- **Network Engineering**
  - Physical Layer
  - Internet Layer
  - Transport Layer
  - Performance

- **Hypermedia**
  - Design & Structure Information Space
  - Navigation
  - Visualization
  - Usability
  - Collaboration

- **Information Systems**
  - Data Design, ER, ...
  - RDBMS
  - Query Languages
  - Strg. Devices: FS, ...

- **Web Engineering**
### Categories of Web-based Systems

<table>
<thead>
<tr>
<th>Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informational</td>
<td>Online newspapers, manuals</td>
</tr>
<tr>
<td>Interactive</td>
<td>Registration forms</td>
</tr>
<tr>
<td>Transactional</td>
<td>Electronic shopping</td>
</tr>
<tr>
<td>Workflow-oriented</td>
<td>Status monitoring</td>
</tr>
<tr>
<td>Collaborative work</td>
<td>Distributed authoring</td>
</tr>
<tr>
<td>Online communities</td>
<td>Discussion groups</td>
</tr>
<tr>
<td>Web portals</td>
<td>Shopping malls</td>
</tr>
<tr>
<td>Web services</td>
<td>Enterprise applications</td>
</tr>
</tbody>
</table>
Do engineers design? Can designers engineer? Looking back at great projects throughout history, it seems these kinds of questions never needed to be asked. There was a philosophy that surfaced in many great works that to do anything well required more than one skill set or discipline. On the contrary, unchecked specialization breeds fragile and shallow ideas. As technology has progressed, I think we’ve lost our connections with the great works of the past and the philosophies and attitudes that enabled their creation. The design and engineering of modern technology, software and the web has bred a hubris that anything older than a few years can’t possibly be relevant, and I think it’s a mistake. To argue this point, there is no better place to start as a basis of comparison and learning than the story of the Brooklyn Bridge.
<table>
<thead>
<tr>
<th>Modern web developer</th>
<th>Washington Roebling’s team</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 week / month release cycle</td>
<td>14 year release cycle</td>
</tr>
<tr>
<td>Electricity</td>
<td>Horses</td>
</tr>
<tr>
<td>Coffee, doughnuts and air conditioning</td>
<td>Water and the elements (think muggy NYC summers)</td>
</tr>
<tr>
<td>Carpal tunnel syndrome</td>
<td>The bends</td>
</tr>
<tr>
<td>Layoffs</td>
<td>27 Deaths</td>
</tr>
</tbody>
</table>
Web Engineering I

- A holistic and pro-active approach to Web systems development
- Offers systematic approaches and disciplined processes for development
- Deals with the management of complexity and diversity of Web development
- Brings to Web-based system development
  - Control
  - Risk minimization
  - Enhanced maintainability and quality
Other factors

- Document orientation
- Navigational design
- Changing technology
- Budget and time constraints
- People and internal politics
- Division between theory and practice
- Lack of understanding…
Develop (high quality) Web Applications
- Effective
- Efficient
- Achieve desired application

Maintain and Evolve
- Plan for change (Solution may change the problem!!!)
  ...using systematic, disciplined and quantifiable Approaches: Process Models
Web Engineering Activities I

- Requirements specification and analysis
- Web-based system analysis and design
- Web development methodologies and techniques
- Migration of legacy systems to Web environments
- Web-based real-time applications development
- Web-based multimedia application development
- Testing, verification and validation techniques and tools
- Quality assessment, control and assurance
Management of access to applications and privileges
Configuration and project management
“Web metrics” – metrics for estimation of development effort
Performance specification and evaluation
Update and maintenance
Development models, teams, and staffing
Human and cultural aspects
User-centric development
Graphics, animation, and streaming
Copyright, legal and social aspects
Web Engineering is Multidisciplinary

- Interface Engineering
- Evaluation And Testing
- Performance Engineering
- Project Management
- Modeling And Simulation
- System Analysis And Design
- Software Engineering
- Multimedia
- Hypertext/Hypermedia
- Information Engineering
- Requirements Engineering
- Interface Engineering
Yesterday’s Webmaster has become tomorrow’s Web Engineer
Some Further Resources

- The Web Engineering Community Portal (http://www.webengineering.org)
- Journal of Web Engineering (http://www.rintonpress.com)
- International Conference on Web Engineering (ICWE) (http://www.icwe2005.org)
Thank You

Questions?

bebo@slac.stanford.edu