INTERNET & WEB APPLICATION DEVELOPMENT SWE 444

Fall Semester 2008-2009 (081)

Module 6: Web Engineering Fundamentals

Dr. El-Sayed El-Alfy
Computer Science Department
King Fahd University of Petroleum and Minerals
alfy@kfupm.edu.sa

Objectives/Outline

- Objectives
  - Understand the role of web engineering
  - Learn a systematic process for web applications development
- Outline
  - Introduction
  - Requirements Analysis
  - Web Modeling
  - Web Design and Architectures
  - Web Accessibility

Resources

- Books
  - Roger S. Pressman, David Lowe (2009). *Web Engineering: A Practitioner's Approach*, McGraw-Hill. http://highered.mcgraw-hill.com/sites/0073523291/
  - Roger Pressman (2005). *Software Engineering: A Practitioner's Approach*, 6/e, McGraw-Hill Higher Education. Chapters 16-20. http://highered.mcgraw-hill.com/sites/0072853182/information_center_view0/
  - G. Kappel, B. Pröll, S. Reich, and W. Retschitzegger (eds), *Web Engineering - The Discipline of Systematic Development of Web Applications*, John Wiley & Sons, 2006. http://www.web-engineering.at/eng/
- Online material
  - INFSCI 2955: Web Engineering
  - Department of Information Science and Telecommunications, University of Pittsburgh http://www.sis.pitt.edu/~jgrady/

6.1 INTRODUCTION TO WEB ENGINEERING

Web Applications (WebApps)

- A *Web application* is a system that utilizes W3C standards & technologies to deliver Web-specific resources to clients (typically) through a browser.
  - A strict definition that ensures software and UI aspects of the Web are examined carefully
- Web applications encompass:
  - complete Web sites
    - Simple information Web sites
    - Complex e-Commerce of other sites with embedded functionality and data retrieval
    - Complex Web sites that are interoperable with other legacy software and systems
  - specialized functionality within Web sites
  - information processing applications that reside on the Internet or on an Intranet or Extranet.

Web Applications: Attributes

- Network intensiveness:
  - resides on a network and must serve the needs of a diverse community of clients.
- Concurrency:
  - A large number of users may access the WebApp at one time; patterns of usage among end-users will vary greatly.
- Unpredictable load:
  - The number of users of the WebApp may vary by orders of magnitude from day to day.
- Performance:
  - If a WebApp user must wait too long (for access, for server-side processing, for client-side formatting and display), he or she may decide to go elsewhere.

Web Applications: Attributes (cont.)

- Availability:
  - Although expectation of 100 percent availability is unreasonable, users of popular WebApps often demand access on a "24/7/365" basis.
- Data driven:
  - The primary function of many WebApps is to use hypermedia to present text, graphics, audio, and video content to the end-user.
- Content sensitive:
  - The quality and aesthetic nature of content remains an important determinant of the quality of a WebApp.
- Continuous evolution:
  - Unlike conventional application software that evolves over a series of planned, chronologically-spaced releases, Web applications evolve continuously.

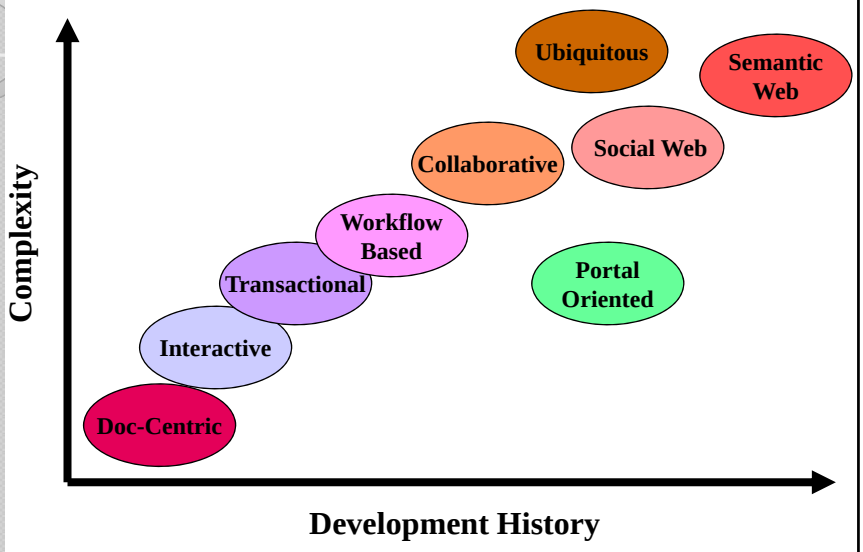
Web Applications: Attributes (cont.)

- Immediacy:
  - WebApps often exhibit a time to market that can be a matter of a few days or weeks.
  - With modern tools, sophisticated Web pages can be produced in only a few hours.
- Security.:
  - In order to protect sensitive content and provide secure modes of data transmission, strong security measures must be implemented throughout the infrastructure that supports a WebApp and within the application itself.
- Aesthetics:
  - When an application has been designed to market or sell products or ideas, aesthetics may have as much to do with success as technical design.

Categories of Web Applications

- Informational
  - read-only content is provided with simple navigation and links
- download
  - a user downloads information from the appropriate server
- customizable
  - the user customizes content to specific needs
- Interaction
  - communication among a community of users occurs via chatroom, bulletin boards, or instant messaging
- user input
  - forms-based input is the primary mechanism for communicating need
- transaction-oriented
  - the user makes a request (e.g., places an order) that is fulfilled by the WebApp
- service-oriented
  - the application provides a service to the user, e.g., assists the user in determining a mortgage payment
- Portal
  - the application channels the user to other Web content or services outside the domain of the portal application
- database access
  - the user queries a large database and extracts information
- data warehousing
  - the user queries a collection of large databases and extracts information

Categories of Web Applications (cont.)



Document-Centric Web sites

- Precursors to Web applications
- Static HTML documents
- Manual updates
- Pros
  - Simple, stable, short response times
- Cons
  - High management costs for frequent updates & large collections
  - More prone to inconsistent/redundant info
- Example: static home pages

Interactive & Transactional

- Come with the introduction of CGI and HTML forms
- Simple interactivity
- Dynamic page creation
  - Web pages and links to other pages generated dynamically based on user input
- Content updates -> Transactions
  - Decentralized
  - Database connectivity
  - Increased complexity
- Examples: news sites, booking systems, online banking

Workflow-Based Applications

- Designed to handle business processes across departments, organizations & enterprises
- Business logic defines the structure
- The role of Web services
  - Interoperability
  - Loosely-coupled
  - Standards-based
- Examples: B2B & e-Government
- High complexity; autonomous entities

Collaborative & Social Web

- Unstructured, cooperative environments
  - Support shared information workspaces to create, edit and manage shared information
- Interpersonal communication is paramount
- Classic example: Wikis
- The Social Web
  - Anonymity traditionally characterized WWW
  - Moving towards *communities of interest*
  - *Examples:* Blogs, collaborative filtering systems, social bookmarking (e.g., del.icio.us)
  - Integration with other forms of web applications (e..g, NetFlix)

Portal-Oriented

- Single points-of-entry to heterogenous information
  - Yahoo!, AOL.com, portal.kfupm.edu.sa
- Specialized portals
  - Business portals (e.g., employee intranet)
  - Marketplace portals (horizontal & vertical)
  - Community portals (targeted groups)

Ubiquitous

- Customized services delivered anywhere via multiple devices
- HCI is critical
  - Limitations of devices (screen size, bandwidth?)
  - Context of use
- Still an emerging field; most devices have single focus:
  - Personalization
  - Location-aware
  - Multi-platform delivery

Semantic Web

- Berners-Lee: Information on the Web should be readable to machines, as well as humans.
- Using metadata and ontologies to facilitate knowledge management across the WWW.
- Content syndication (RSS, Atom) promotes re-use of knowledge
- Is the Semantic Web even possible?

The Case for Web Engineering

- Application development on the Web remains largely *ad hoc*.
  - Spontaneous, one-time events
  - Individual experience
  - Little or no documentation for code/design
- Short-term savings lead to long-term problems in operation, maintenance, usability, etc.
- Because WebApps are so interdependent, the problem is compounded.
- Root causes of poor design:
  - Development as an authoring activity
  - Development is "easy"
  - Techniques that *should not* be used are misapplied.
  - Techniques that *should* be used are *not*.
- Particularly alarming given…
  - Most projects are now Web-based
  - More "mission-critical" apps moving to the Web

The Case for Web Engineering (cont.)

- Top project pitfalls (Cutter, 2000)
  - 84% - Failure to meet business objectives
  - 79% - Project schedule delays
  - 63% - Budget overrun
  - 53% - Lack of functionality
- Web Engineering's solution:
  - Clearly defined goals & objectives
  - Systematic, phased development
  - Careful planning
  - Iterative & continuous auditing of the entire process

What is Web Engineering (WebE)?

- "The application of systematic and quantifiable approaches to cost-effective analysis, design, implementation, testing, operation, and maintenance of high-quality Web applications." – Kappel *et al.*
- "Web development is an adolescent … Like most adolescents, it wants to be accepted as an adult as it tries to pull away from its parents. If it is going to reach its full potential, it must take a few lessons from the more seasoned world of software development." -- Doug Wallace et al
- Extends *Software Engineering* to Web applications, but with Web-centric approaches.

Characteristics of Web Apps

- How do WebApps differ from traditional applications?
- Or, another way, what Software Engineering methods & techniques can be adapted to Web Engineering?
- 3 dimensions of the ISO/IEC 9126-1 standard
  - Product
  - Usage
  - Development

Characteristics - Product

- Product-related characteristics constitute the "building blocks" of a Web application
- Content
  - Document character & multimedia (# of dimensions?)
  - Quality demands: current, exact, consistent, reliable
- Navigation Structure (Hypertext)
  - Non-linearity
  - Potential problems: Disorientation & cognitive overload
- User interface (Presentation)
  - Aesthetics
  - Self-explanation

Characteristics - Usage

- Much greater diversity compared to traditional non-Web applications
  - Users vary in numbers, cultural background, devices, h/w, s/w, location etc
- Social Context (Users)
  - Spontaneity - scalability
  - Heterogeneous groups
- Technical Context (Network & Devices)
  - Quality-of-Service
  - Multi-platform delivery
- Natural Context (Place & Time)
  - Globality
  - Availability

Characteristics - Development

- The Development Team
  - Multidisciplinary – print publishing, s/w devt, marketing & computing, art & technology
  - Community (including Open Source)
- Technical Infrastructure
  - Lack of control on the client side
  - Immaturity
- Process
  - Flexibility
  - Parallelism
- Integration
  - Internal – with existing legacy systems
  - External – with Web services
  - Integration issues: correct interaction, guaranteed QoS

The WebE Process

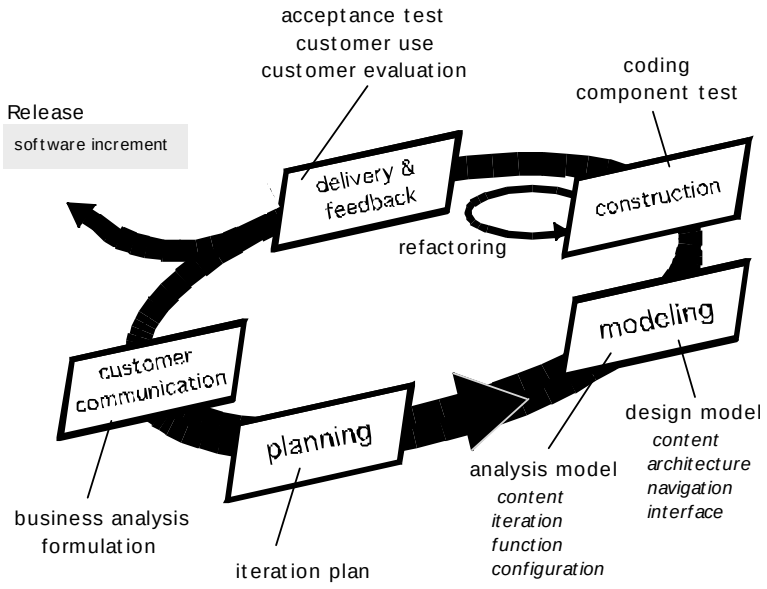
Must accommodate:

- Incremental delivery
- Frequent changes
- Short timeline

Therefore,

- An incremental process model should be used in virtually all situations
- An agile process model is appropriate in many situations

The WebE Process (cont.)



WebE: Basic Questions

- How important is a Web site home page?
- What is the most effective page layout (e.g., menu on top, on the right or left?) and does it vary depending upon the type of WebApp being developed?
- Which media options have the most impact?
- How much work can we expect a user to do when he or she is looking for information?
- How important are navigational aids when WebApps are complex?
- How complex can form input be before it becomes irritating for the user? How can form input be expedited?
- How important are search capabilities?
- Will the WebApp be designed in a manner that makes it accessible to those who have physical or other disabilities?

-- *Susan Weinshenk*

WebE: Best Practices

- Take the time to understand the business needs and product objectives, even if the details of the WebApp are vague.
- Describe how users will interact with the WebApp using a scenario-based approach
- Develop a project plan, even if it is very brief.
- Spend some time modeling what it is that you're going to build.
- Review the models for consistency and quality.
- Use tools and technology that enable you to construct the system with as many reusable components as possible.
- Don't rely on early users to debug the WebApp—design comprehensive tests and execute them before releasing the system.

Q & A