Overview

1. What is Web Application?
2. Web Application Engineering
3. The Web Engineering Process
4. Best Practices

Web Applications

- WebApps 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.
WebApp Attributes—I
- Network Intensiveness（密集型）
- Concurrency
- Unpredictable Load（不可预测的负载）
- Performance
- Availability（可用性）
- Data driven
- Content Sensitive（敏感性）
- Continuous Evolution

WebApp Attributes—II
- Immediacy（即时性）
- Security
- Aesthetics（美学）

WebApp Categories I
- 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 chat room, bulletin boards, or instant messaging
- User input—forms-based input is the primary mechanism for communicating need
### WebApp Categories II

- **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

### WebApp Engineering

Web Engineering deals with disciplined and systematic approaches to development, deployment, and maintenance of Web-based system and applications.

- Yogesh Deshpande

### WebApp Engineering Layers I

- **Process**
  - It is often agile and is almost always incremental:
    - (1) embraces change;
    - (2) encourages the creativity and independence of development staff and strong interaction with WebApp stakeholders;
    - (3) builds systems using small development teams;
    - (4) emphasizes evolutionary or incremental development using short development cycles.
WebApp Engineering Layers II

- **Methods**
  - Communication Methods;
  - Requirements analysis methods;
  - Design methods;
  - Testing methods;

- **Tools and technology**

Web Development

“Web development is an adolescent (青春期)...
... 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 (潜力), it must take a few lessons from the more seasoned (经验丰富的) world of software development.”

Doug Wallace et al

Defining the framework

- Must accommodate: (适应)
  - Incremental delivery
  - Continuous change
  - Short timelines
- Therefore,
  - An incremental process model should be used in virtually all situations
  - An agile process model is appropriate in many situations
The Web Engineering Process I

- Customer communication
- Planning
- Modeling
- Construction
- Deployment

The Web Engineering Process II

- Software increment
- Release
- Refactoring
- Business analysis
- Formulation
- Iteration plan
- Analysis model
- Interface
- Navigation
- Content
- Configuration
- Design
- Test
- Acceptance test
- Customer use
- Customer evaluation

The Process Framework—I

- Customer communication
- Business analysis
- Formulation
- Planning
- a task definition
- a timeline schedule
The Process Framework—II

- **Modeling**
  - Analysis model—establishes a basis for design
  - Content Analysis
  - Interaction Analysis
  - Functional Analysis
  - Configuration Analysis
- **Design model**—represents key WebApp elements
  - Content design
  - Aesthetic design
  - Architectural design
  - Interface design
  - Navigation design
  - Component design

The Process Framework—III

- **Construction**
  - WebE tools and technology are applied to construct the WebApp that has been modeled
  - Testing of all design elements
- **Delivery and Evaluation (Deployment)**
  - configure for its operational environment
  - deliver to end-users
  - evaluation feedback

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 forms input be before it becomes irritating for the user? How can forms 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
Best Practices I

- 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's very brief.
- Spend some time modeling what it is that you're going to build.

Best Practices II

- 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.