CHAPTER 16 WEB ENGINEERING

Overview

The chapter describes Web engineering (WebE) as the process used to create high quality Web-based applications (WebApps). As WebApps become increasingly integrated in business strategies (e.g., e-commerce) the need to build reliable, usable, and adaptable systems grows in importance. Web engineering is not a perfect clone of software engineering, but it draws heavily on many of software engineering's principles and management activities. The Web engineering process begins with the formulation of the problem to be solved by the WebApp. The project is planned and the WebApp requirements are analyzed. Architectural, navigational, and interface design are conducted. The system is implemented using specialized languages and tools associated with the Web. WebApps tend to be highly evolutionary, so mechanisms for configuration management, quality control, and maintenance must be established early. Web engineering relies on formal technical reviews to assess the quality of the analysis and design models. Specialized reviews are conducted to assess the usability of the WebApp. Testing is applied to uncover errors in content, functionality, and compatibility.

WebApp Attributes

- Network intensive
- High concurrency
- Unpredictable user loads
- Performance (fast delivery)
- High availability
- Data driven
- Content sensitive
- Continuous evolution
- Immediacy
- Security
- Aesthetics

WebE Application Categories

- Informational (read only content provided with simple navigation)
- Downloads (user downloads information from server)
- Customizable (user customizes content to specific needs)
- Interaction (community of users communicate using chat rooms, bulletin boards, or instant messaging)
- User input (users complete on-line forms to communicate need)
- Transaction-oriented (user makes request fulfilled by WebApp - places an order)
- Service-oriented (application provides service to user, e.g., helps user determine mortgage payment)
- Portal (application directs users to other web content or services)
- Database access (user queries a large database and extracts information)
• Data warehousing (user queries large collection of databases and extracts information)

WebApp Enabling Technologies

• Component-based development (CORBA, COM/DCOM, JavaBeans)
• Security (encryption, firewalls, etc.)
• Internet standards (HTML, XML, SGML)

WebApp Development Process Characteristics

• WebApps are often delivered incrementally
• Changes occur frequently
• Timelines are short

WebApp Process Framework

• Customer communication
  ▪ Business analysis - defines business and organizational context for WebApp
  ▪ Formulation - involves stakeholders in gathering information needed to describe WebApp requirements
• Planning
  ▪ Plan consisting of task definition and timeline for the WebApp increment being developed
• Modeling
  ▪ Rapid analysis and design modeling tasks are adapted to WebApp
  ▪ Models define the requirements and define the WebApp needed to satisfy them
• Construction
  ▪ WebE tools used to construct WebApp from model
  ▪ Constructed increment is subjected to rapid tests to ensure errors (i.e., content, architecture, interface, navigation)
• Delivery and evaluation
  ▪ WebApp configured for its operational environment and delivered to end-users
  ▪ Users evaluate WebApp in their environment and provide feedback to the development team
  ▪ Increment is modified as required

WebE Analysis and Design Questions

• How important is a Web site homepage?
• What is the most effective page layout?
• What media applications have the most impact?
• How much work can we expect the user to do when searching for information?
- How important are navigational aids in a complex WebApp?
- How complex can input forms be without irritating the user?
- How important are search and browse capabilities to the users?
- Will the WebApp be designed to be accessible to users with disabilities?

WebE Task Set

- Customer Communication (Analysis/Formulation)
  - Identify business stakeholders
  - Formulate business context
  - Define key business goals and objectives
  - Define information and applicative goals
  - Identify the problem
  - Gather requirements
- WebE Planning
  - Define development strategy
  - Assess risks
  - Define development schedule
  - Establish metrics for content management and change control
- WebE Modeling
  - Refine content classes
  - Identify content relationships
  - Refine user tasks
  - Identify computational functions
  - Identify database requirements
  - Refine interface requirements
  - Design the WebApp architecture
  - Design the interface
  - Design the navigational scheme
  - Design appropriate security and privacy mechanisms
  - Review the design
- WebE Construction
  - Build or acquire the content and integrate it into WebAPP architecture
  - Establish navigational capabilities
  - Implement computational functions
  - Address configuration issues
  - Test all WebApp components
  - Address configuration issues
  - Test all WebApp components (content and function)
  - Test navigation
  - Test usability
  - Test security and performance
  - Test increment for different configurations
- WebE Delivery and evaluation
  - Deliver WebApp increment to representative end-users
  - Evaluate end-user interaction
- Assess lessons-learned and consider all end-user feedback
- Make modifications to WebApp increment as required

Web Engineering Best Practices

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