

Chapter 1: *Web-Based Systems*

- In the early days of the Web, we built systems using **informality, urgency, intuition, and art**
 - *Informality* leads to an easy work environment—one in which you can do your own thing.
 - *Urgency* leads to action and rapid decision making.
 - *Intuition* is an intangible quality that enables you to “feel” your way through complex situations.
 - *Art* leads to aesthetic form and function—to something that pleases those who encounter it.
- Problem is—**this approach can and often does lead to problems**

But Why?

- As WebApps become larger and more complex,
 - Informality remains, but some degree of requirements gathering and planning are necessary
 - Urgency remains, but it must be tempered by a recognition that decisions may have broad consequences
 - Intuition remains, but it must be augmented by proven management and technical patterns
 - Art remains, but it must be complemented with solid design
- Bottom line—we must adapt the old-school approach to the realities of a Web 2.0 world

And What's the Response?

Web Engineering

The Web

- An indispensable technology
 - In virtually every aspect of modern living
- A transformative technology
 - Changes the way we do things
 - Changes the way we acquire and disseminate information
- An evolving technology
- Bottom line—high impact on everyone in the modern world

WebApps

- The term *Web application (WebApp)* encompasses:
 - everything from a simple Web page that might help a consumer compute an automobile lease payment to a comprehensive website that provides complete travel services for business people and vacationers.
 - Included within this category are complete websites, specialized functionality within websites, and information-processing applications that reside on the Internet or on an Intranet or Extranet.

WebApp Attributes

- Network intensiveness
- Concurrency
- Unpredictable load
- Performance
- Availability
- Data driven
- Content sensitive
- Continuous evolution
- Immediacy
- Security
- Aesthetics

WebApp Types

- Informational
- Download
- Customizable
- Interaction
- User input
- Transaction-oriented
- Service-oriented
- Portals
- Database access
- Data warehousing

(see <http://digitalenterprise.org/models/models.html> for examples)

Chapter 2: *Web Engineering*

- We define it this way:
 - *an agile, yet disciplined framework for building industry-quality WebApps.*
- We must understand the meaning of:
 - Agile
 - Disciplined framework
 - Industry quality

Why Agility?

- Business strategies and rules change rapidly
- Management demands near-instantaneous responsiveness (even when such demands are completely unreasonable)
- Stakeholders often don't understand the consequences of the Web and keep changing their mind even as they demand rapid delivery
- An agile approach helps cope with this fluidity and uncertainty.

What is an Agile Process?

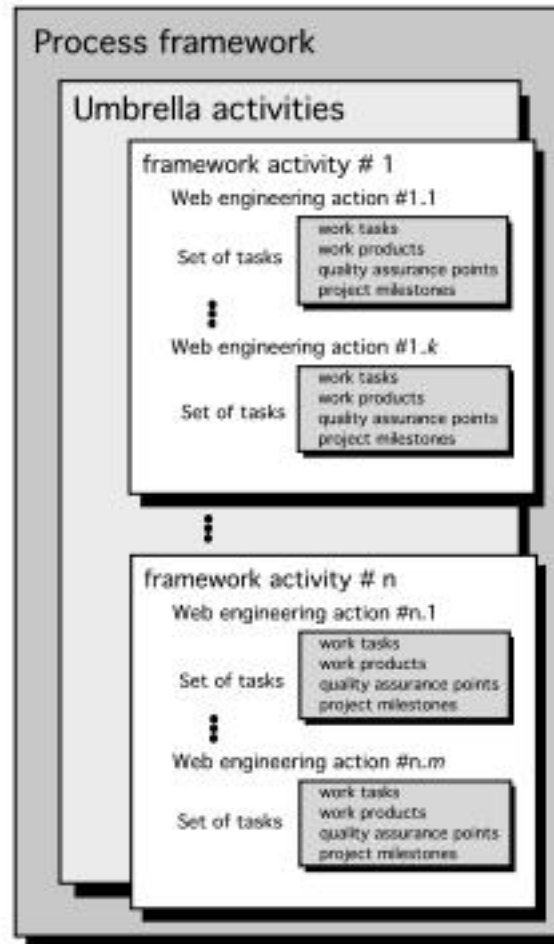
- Agile Web engineering combines a philosophy and a set of development guidelines. The philosophy encourages:
 - customer satisfaction
 - early incremental delivery of the WebApp
 - small, highly motivated project teams
 - informal methods
 - minimal work products
 - overall development simplicity.
- An agile process stresses delivery over analysis and design (although these activities are not discouraged), and active and continuous communication between developers and customers.

What is a WebE Framework?

- A framework is a set of activities that will *always* be performed for *every* Web engineering project – though the nature of the activities might vary to suit the project.
- Each framework activity is composed of a set of actions
- Actions encompass
 - work tasks
 - work products
 - quality assurance points, and
 - project milestones
- A framework also has a set of “umbrella activities”

A Generic Framework

WebE process



The WebE Framework: Activities

- **Communication.** Involves heavy interaction and collaboration with the customer (and other stakeholders) and encompasses requirements gathering and other related activities.
- **Planning.** Establishes an incremental plan for the WebE work.
- **Modeling.** Encompasses the creation of models that assist the developer and the customer to better understand WebApp requirements and the design
- **Construction.** Combines both the generation of HTML, XML, Java, and similar code with testing that is required to uncover errors in the code.
- **Deployment.** Delivers a WebApp increment to the customer who evaluates it and provides feedback based on the evaluation.

Adapting the Framework

- Adapt to the problem, to the project, to the team, and to the organizational culture
 - And continue to adapt throughout the project as circumstances change!
- Adaptation leads to:
 - Overall flow of activities, actions, and tasks and the interdependencies among them
 - Degree to which work tasks are defined within each framework activity
 - Degree to which work products are identified and required
 - Manner in which quality assurance activities are applied
 - Manner in which project tracking and control activities are applied
 - Overall degree of detail and rigor with which the process is described
 - Degree to which customers and other stakeholders are involved with the project
 - Level of autonomy given to the software project team
 - Degree to which team organization and roles are prescribed

Underlying Agility Principles - I

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness continuous change for the customer's competitive advantage.
- Deliver working software increments frequently, from as often as every few days to every few months, with a preference to the shorter timescales.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated people. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Underlying Agility Principles - II

- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The Influence of Software Engineering



- Quality: foster a continuous process improvement culture
- Process: the glue that holds the technology layers together
- Methods: provide the technical how-to's
- Tools: support for the process and the methods

WebE Methods

- Communication methods
- Requirements analysis methods
- Design methods
- Construction methods
- Testing methods

What about Tools and Technology?

... tools and technology are very important, but they'll work well only if they're used within the context of an agile framework for Web engineering and in conjunction with proven methods for understanding the problem, designing a solution, and testing it thoroughly.

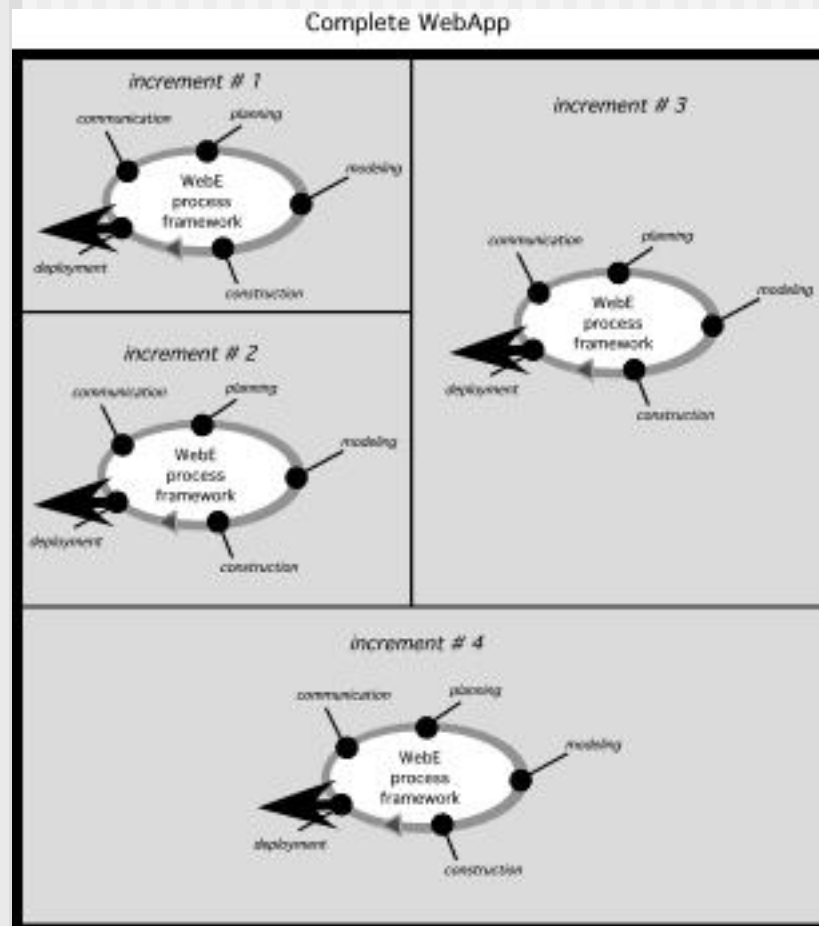
WebE Best Practices

- Take the time to understand 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.
- *Always develop a project plan*, even if it's 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 reinvent when you can reuse.
- Don't rely on early users to debug the WebApp—design and use comprehensive tests before releasing the system.

Chapter 3: *The WebE Process*

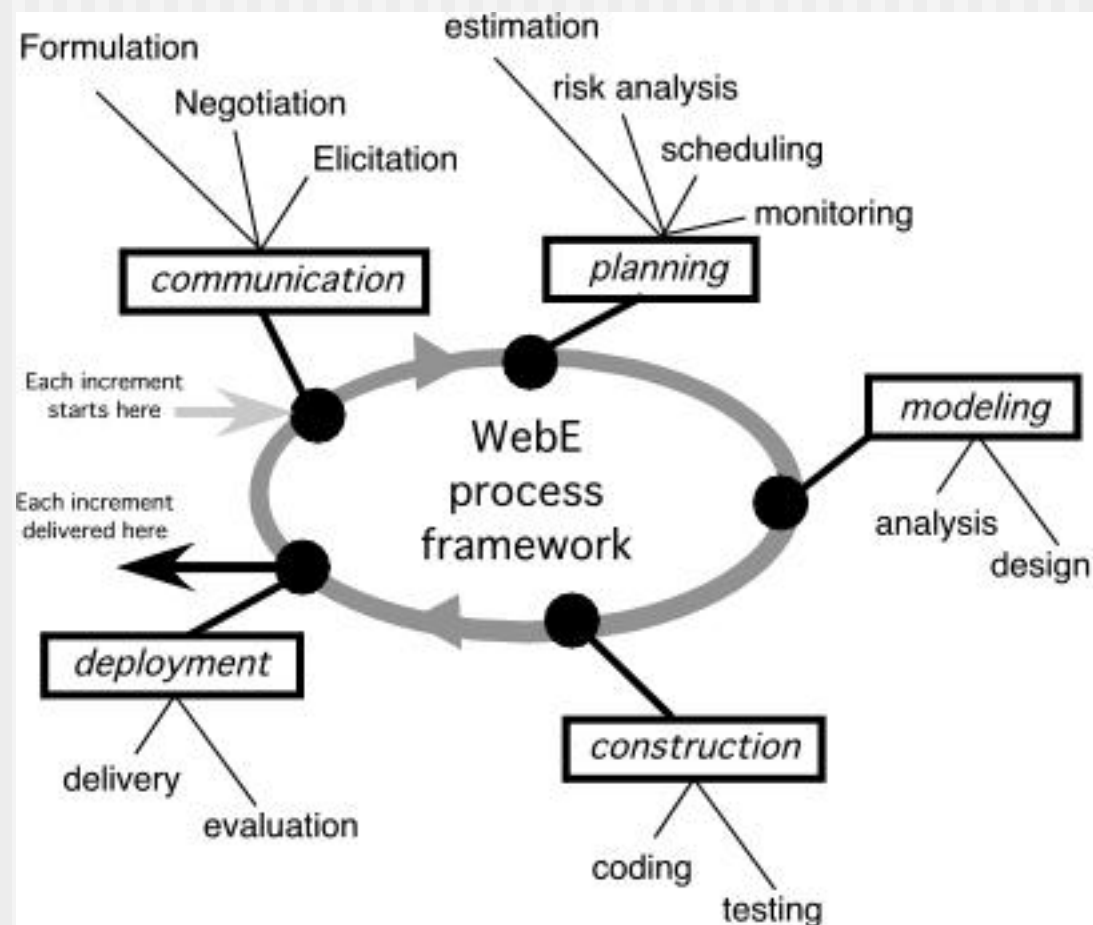
- The process must be agile and adaptable, but it must also be *incremental*
- Why incremental?
 - Requirements evolve over time
 - Changes will occur frequently (and always at inconvenient times)
 - Time lines are short
- Incremental delivery allows you to manage this change!

Incremental Delivery



Repeat the development cycle for each increment!

WebE Process Activities & Actions



Conducting Framework Activities-I

- The first iteration
 - define business context
 - establish overall requirements
 - create a set of usage scenarios
 - negotiate conflicting needs among stakeholders, and
 - from this information derive the set of WebApp increments that is to be delivered.
- Develop a broad outline of all components, recognizing that it will change

Conducting Framework Activities-II

- The second iteration
 - You've learned that the first increment is an informational WebApp and it must be delivered in one week!
 - You meet with stakeholders and later review your notes:
 - Logo and graphics—need aesthetic design.
 - One- or two-paragraph introduction.
 - CPI mission statement (file exists)
 - A word to visitors (someone will write this tomorrow)
 - Basic navigation bar will look like ...
 - About the company
 - Our offerings
 - Home security products (hierarchical at next level)
 - Monitoring services (a list)
 - Our Technology (the new sensor)
 - Contact us
 - Other issues:
 - Informational content will change over time.
 - This "home page" will be the navigation starting point for content and functions required for subsequent increments.

Conducting Framework Activities-III

■ The second iteration

■ You spend a few minutes developing a plan

- Day 1: Create a prototype layout (a model) of the WebApp.
- Collect and review all existing CPI content and graphics.
- Get stakeholder feedback on prototype, if possible.
- Day 2: Using the prototype as a guide, begin construction of the increment.
- Build navigation bar.
- Lay out content areas.
- Integrate graphics, links, etc.
- Test all links for validity.
- Review all content for completeness and correctness.
- Day 3: FTP all files to (an existing) domain.
- Perform navigation tests.
- Deployment: Inform selected stakeholders that the increment is available.
- Day 4: Poll stakeholders for feedback.
- Make modifications based on stakeholder feedback.

Conducting Framework Activities-IV

- The next iteration
 - You've deployed the informational WebApp
- the communication activity during this second iteration will identify the requirements (including content and functionality)
 - assume that the second increment delivers the capability to select and download product specifications and related information
- the process flow is restarted at the beginning, performing the communication activity for this increment.
- The tasks you select to populate each framework activity for the increment may differ from the tasks performed for the preceding increment, but the overall process flow remains the same

Revisiting the Framework Activities

- WEPA pp. 32 - 42 presents a breakdown of the generic actions and tasks for each of the five framework activities
- Recognize that a WebE team must refine and adapt these generic tasks to the problem at hand
 - And continue to refine them throughout the project

Communication

- Identify business stakeholders
- Identify user categories
- Formulate the business context
- Define key business goals and objectives
- Identify the problem
- Define informational and applicative goals
- Gather requirements
- Develop usage scenarios

Planning

- Refine description of the increment to be delivered.
- Select the increment to be delivered now.
- Estimate the effort and time required to develop the increment.
- Assess risks associated with the delivery of the increment.
- Define the development schedule for the increment.
- Establish work products to be produced as a consequence of each framework activity.
- Define approach to change control.
- Establish quality assurance approach.

Analysis Modeling

- Decide whether a requirements model is needed.
- Classify and represent web application content
- Identify content relationships
- Refine and extend user scenarios
- Review usage scenarios
- Create an interaction model for complex scenarios
- Refine interface requirements
- Identify functions
- Define constraints and performance requirements
- Identify database requirements

Aspects of Web Application Design

- Interface design
- Aesthetic design
- Content design
- Navigation design
- Architecture design
- Component design

Design Modeling

- Design the interface
- Design the aesthetic
- Design the navigation scheme
- Design the architecture
- Design the content and the structure that supports it
- Design functional components
- Select appropriate design patterns
- Design appropriate security and privacy mechanisms
- Review the design

Construction

- Build and/or acquire all content, and integrate the content into the web application architecture
- Select the appropriate tool for generation of HTML code
- Implement each page layout, function, form, and navigation capability
- Implement all computation functions
- Address configuration issues.

Testing

- Test all web application components (content and function)
- Test navigation
- Test usability
- Test security
- Test the web application increment for different configurations.

Deployment

- Deliver the web application increment to a server as a predefined domain.
- Establish an online feedback mechanism for end users.
- Evaluate end-user interaction
- Assess lessons learned and consider all end-user feedback.
- Make modifications to the web application increment as required.

Umbrella Activities

- Background activities which occur in parallel with the main development activities
- Equally important to the success of a project
 - And so should be considered explicitly.
- Many umbrella activities can be defined
 - But only four are crucial for a successful Web engineering project:

Umbrella Activities

- **Change management.** Manages the effects of change as each increment is engineered, integrating tools that assist in the management of all WebApp content
- **Quality assurance.** Defines and conducts those tasks that help ensure that each work product and the deployed increment exhibits quality
- **Risk management.** Considers project and technical risks as an increment is engineered
- **Project management.** Tracks and monitors progress as an increment is engineered