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.

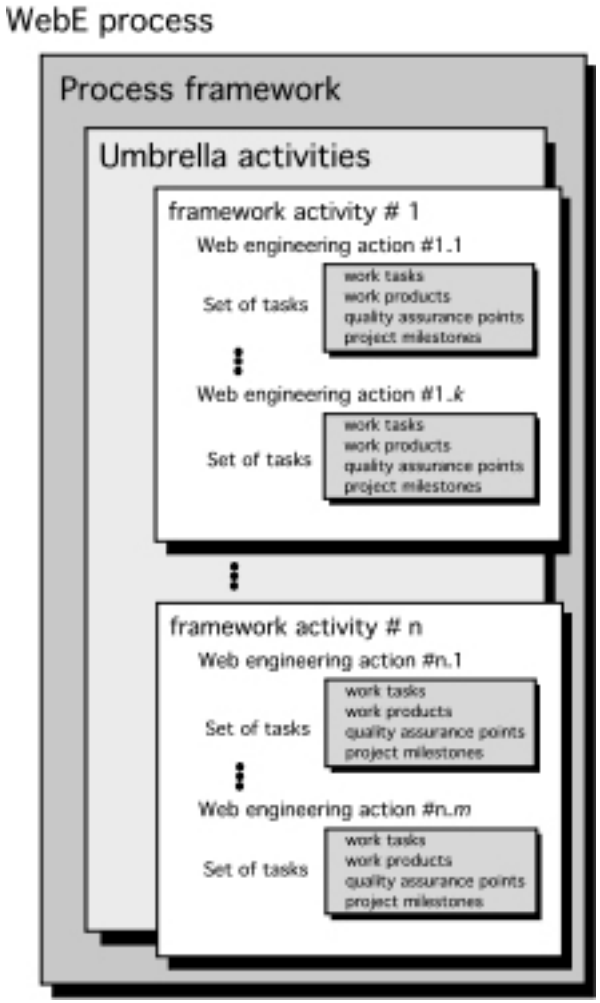
What is an Agile Process?

- 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



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 impacts:
    - 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 d*evelop 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.