

Lecture 2 Notes

Web Engineering

Web Engineering

- **We define it this way:** Web engineering proposes 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 minds 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 philosophy and a set of development guidelines (Process).
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.

Underlying Agility Principles

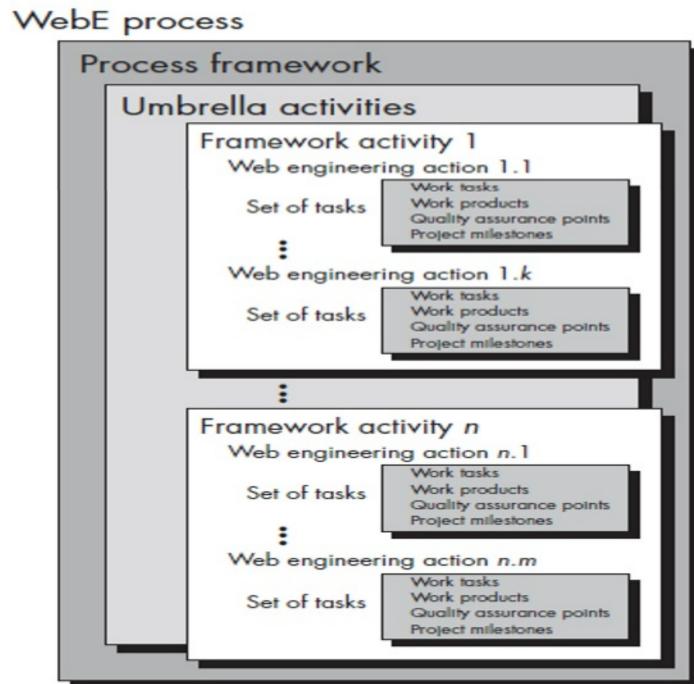
- Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software.
- **Welcome changing requirements**, even late in development. Agile processes harness continuous change for the customer's competitive advantage.
- Delivering working software **increments frequently**, from as often as every few days to every few months, with a preference to the shorter timescales.
- Businesspeople 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**.
- **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 architecture, 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.

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 (e.g., design is a WebE action).
- **Actions encompass:**
 - work tasks: accomplish some part of the work implied by the action.
 - work products
 - quality assurance points, and
 - project milestones
- A framework also has a set of “**umbrella activities**”: that are applicable across the entire WebE process.

A Generic Framework



The WebE Framework: Activities

The following WebE activities are part of a *generic framework* and are applicable to the vast majority of WebApp projects:

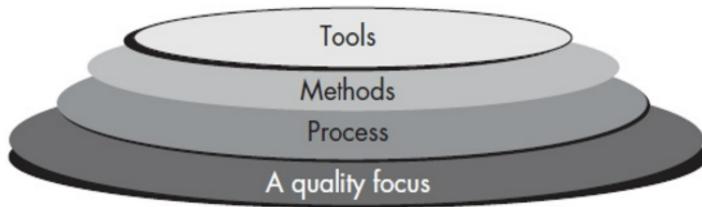
- **Communication:** Involves heavy interaction and collaboration with the customer (and other stakeholders) and encompasses requirements gathering and other related activities.
- **Planning:** Establishes an incremental plan for the WebE work.
- **Modeling:** Encompasses the creation of models that 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

The Influence of Software Engineering on WebApps



Software engineering is a layered technology:

1. **Quality:** foster a continuous process improvement culture.
2. **Process:** the glue that holds the technology layers together and enables rational and timely development of computer software.
3. **Methods:** provide the technical how-to's for building software.
4. **Tools:** provide automated or semiautomated support for the process and the methods.

WebE Methods

WebE methods can be categorized into:

- **Communication methods:** facilitate communication between Web engineers and stakeholders.
- **Requirements analysis methods:** for understanding the content to be delivered by a WebApp.
- **Design methods:** address WebApp content, application and information architecture, interface design and navigation structure.
- **Construction methods:** Apply languages, tools, and related technology to the creation of WebApp content and functionality.
- **Testing methods:** technical reviews of both the content and design model. Testing techniques address component-level and architectural issues, navigation testing, usability testing, security testing, and configuration testing.

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 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.