Topic 3

The Web Engineering (WebE) Process Models

Web Engineering Process:
Framework (main) Activities & Actions
(software development life cycle – SDLC – for Web systems)
5 WebE framework activities (for large Web systems, or in large-size companies).

But … in what order? returns/iterations/loops? Feedback from users?

**Main Types of SE Process Flow**
(Various types will have significant impact on project time, cost, human resources, quality, etc.)

a) **Linear process flow** (no feedback, no iterations/loops)

b) **Iterative process flow** (a circular manner, with a feedback, with iterations)

c) **Evolutionary process flow** (with various versions or increments released)

d) **Parallel process flow** (no feedback, no iterations/loops, with parallel activities)

Web system/software development team (project manager, system analyst, software engineers, etc.) MUST identify the best (optimal) process flow or, possibly, a combination of them for a specific software project.
Question: Why do we need to know all those types of SE Process Flow?
Answer: It is about Money, Time, People, and Goal (Top Quality Software/Web system at the end)
(Various types will have significant impact on project time, cost, human resources, quality, etc.)

<table>
<thead>
<tr>
<th>% of project budget</th>
<th>Waterfall model</th>
<th>Prototyping/iteration model</th>
<th>Spiral model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>5-10%</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Planning</td>
<td>25-30%</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Modeling</td>
<td>25-40%</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Construction/development</td>
<td>15-25%</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Deployment</td>
<td>10-15%</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

Notes:
The same budget but for 3..5..7... prototypes or iterations is unknown at the beginning.
The same budget but for 3..5..7... evolutionary spirals of spiral is unknown at the beginning and each spiral is more expensive than the previous one.

Classic SE: Waterfall Model

1. Requirements
2. Design
3. Construction and unit testing
4. Integration and system testing
5. Operation
The 1st increment = a core product.
Following increments are aimed to better meet customer requirements and deliver additional functionality.
Classic Evolutionary Process Flow: A Prototyping Model

Prototyping Model (good first step when customer has a legitimate need, but is clueless about the details, developer needs to resist pressure to extend a rough prototype into a production product)

Web Engineering:

Use Incremental and Prototyping Models
Prototyping Models

- **Throwaway prototyping** – (also called close Ended Prototyping, or Rapid Prototyping) refers to the creation of a model that will eventually be discarded rather than becoming part of the final delivered software.

- **Evolutionary Prototyping** (also known as Breadboard Prototyping) is quite different from Throwaway Prototyping. The main goal when using Evolutionary Prototyping is to build a very robust prototype in a structured manner and constantly refine it.

- **Incremental prototyping** – In this case, the final product is built as separate prototypes. At the end, the separate prototypes are merged in an overall design.

- **Extreme prototyping** – Extreme Prototyping as a development process is used for developing especially web applications. Basically, it breaks down web development into three phases, each one based on the preceding one.
  1. **STATIC.** The first phase is a static prototype that consists mainly of HTML pages.
  2. **FUNCTIONALITY ADDED TO EACH PAGE/SCREEN.** In the second phase, the screens are programmed and fully functional using a simulated services layer.
  3. **SERVICES ADDED.** In the third phase, the services are implemented.

The process is called Extreme Prototyping to draw attention to the second phase of the process, where a fully-functional UI is developed with very little regard to the actual services to be used.

Source: Software Engineering, 7th Ed., by Roger Pressman

Web Engineering Process:
Framework (main) Activities & Actions
(development cycle based on incremental models)

- **Why incremental?**
  - Requirements evolve over project time (duration)
  - Changes will occur frequently (and always at inconvenient times for engineers and developers)
  - Time lines are short … and very short …

- **The answer:** Incremental delivery allows you to manage those changes!
Incremental Delivery

Repeat the ENTIRE system development life cycle (SDLC) for each quick increment (however, with most of required framework activities):

- Increment 1: small amount of time, just a few functions, GUI.
- Increment # 2: small amount of time, just a few functions, GUI.
- Increment # 3: Time is doubled + new functions
- Increment # 4: Time is doubled again + almost all other functions.

The system grows by adding new and enhanced functionality with each increment created.

Each increment tackles a relatively small set of requirements and proceeds until the entire scope of the project is completed.

Repeating a process phase until ultimately meeting the project requirements (iterating the phases) and developing and delivering a system in stages (increments).

The Incremental Model: Real World Example

- By September 1991, Linux version 0.01 was released. It had 10,239 lines of code.
- In October 1991, Linux version 0.02 was released.
- In December 1991, Linux 0.11 was released. This version was the first to be self-hosted - Linux 0.11 could be compiled by a computer running Linux 0.11.
- When he released version 0.12 in February 1992, Torvalds adopted the GNU General Public License (GPL) over his previous self-drafted license, which had not permitted commercial redistribution.
- In March 1992, Linux version 0.95 was the first to be capable of running X. This large version number jump (from 0.1x to 0.9x) was due to a feeling that a version 1.0 with no major missing pieces was imminent. However, this proved to be somewhat overoptimistic, and from 1993 to early 1994, 15 development versions of version 0.99 appeared.
- On 14 March 1994, Linux 1.0.0 was released, with 176,250 lines of code.
- In March 1995, Linux 1.2.0 was released (310,950 lines of code).
- Version 2 of Linux, released on 9 June 1996, was followed by additional major versions under the version 2 header, including the following ones:
  - 25 January 1999 - Linux 2.2.0 was released (1,800,847 lines of code).
  - 18 December 1999 - IBM mainframe patches for 2.2.13 were published, allowing Linux to be used on enterprise-class machines.
  - 4 January 2001 - Linux 2.4.0 was released (3,377,902 lines of code).
  - 17 December 2003 - Linux 2.6.0 was released (5,929,913 lines of code).
  - 9 June 2009 - Linux 2.6.30 was released (11,637,173 lines of code).
- Jan 2011: Linux Linux 2.6.37

In July 2009 Microsoft contributed 20,000 lines of code to the Linux kernel. The contribution consisted of Hyper-V drivers, which improve the performance of virtual Linux guest systems in a Windows hosted environment. Microsoft licensed its Linux Hyper-V drivers under the GPL.
Multiple Possible Questions

- Should we throw away a previous increment?
- Will we use a previous increment and refine it?
- Should we use primitive tools (like HTML) for first increments and sophisticated tools (like Dreamweaver) for following increments?
- How can we increase SW (increment’s) development speed?
- …
Incremental Model: Strengths and Weaknesses

- **Strengths**
  - Generates working software relatively quickly and early
  - **Flexibility** (requirements and limits can be changed)
  - **Ease of risk management** (we know in advance that we can change, improve, update increments)

- **Weaknesses**
  - How many iterations are needed to be sponsored? Budget breakdown for unpredictable number of iterations is an issue
  - Not easy to manage due to multiple possible changes (too many changes may happen and very quickly)
  - Hard to determine total cost and time estimates early in the WebE process

Tools (IDEs) for Quick Developments of SW Increments: Desktop Applications (examples)

- Apple Xcode
- C++Builder
- Clarion is a data-centric Advanced Rapid Application Development tool
- Code::Blocks
- Delphi
- Delphi for PHP
- Gambas Basic, Open source, Linux
- Gupta Team Developer / SQLWindows
- Habanero
- Microsoft Visual Basic
- Lazarus Pascal, Open Source, Multi-platform
- Panther
- RADvolution Designer
- Runtime Revolution
- REAL software REALbasic
- Softwell Maker is a desktop IDE with a cross-platform deployment component allowing publish application in almost any Java enable system.
- Thoroughbred OPENworkshop is a RAD for Windows, UNIX, Linux, and OpenVMS
- The Virtual Enterprise is an Interactive Voice Response (IVR) toolkit developed specifically for telephony and speech inside Microsoft Visual Studio.NET.
- wxDev-C++
- Microsoft Visual Foxpro
- WinDev
- XVT
- MX-Frame - Business Application Framework
- GNavi
Tools for Quick Increments: Databases (examples)

- **Base One Foundation Component Library (BFCL)** is a RAD framework for building .NET applications using SQL Server, Oracle, Unix, Sybase, and MySQL databases.
- **Clio** is a database-independent RAD framework to build traditional GUI applications, as well as web servers. It supports application Web pages ASPX, user interface code and data access logic (C#, VB.NET Basic, .NET, and SQL queries) without hand-coding.
- **FileMaker** is a cross-platform database application from FileMaker Inc. (a subsidiary of Apple Inc.)
- **Sybase PowerBuilder** is a data-driven development tool for creating client/server, distributed, Web and Smart Clients applications for J2EE, Win32, and .NET platforms.
- **Kexi** is an open source database-driven RAD development environment for building desktop applications. It is considered an alternative to OpenOffice Base in the Open Source environment and provides similar features to commercially available RAD development environments such as FileMaker, Alpha Five and Microsoft Access.
- **Oracle Forms**
- **Oracle Application Express** (Oracle APEX) is a software development environment based on the Oracle database. It allows for a very fast development cycle to be achieved to create web-based applications.
- **Panther** (and its open source version POSH) is a cross-platform (Windows, Unix, Linux, TUX, GUI, Web) cross-database RAD toolset for development of C/S and B/S database oriented applications.
- **NConstruct** is Windows and Web rapid enterprise application development tool and environment for .NET framework. It supports Oracle database, Microsoft SQL Server databases and Microsoft Access.
- **EASYProcess** is a .NET based RAD for the creation of web portal interfaces, work flows, data integration, reporting and web services largely focused on the JD Edwards community.
- **Softwell Maker** is a cross-platform deployment component allowing publication into almost any Java enabled system.
- **WinDev**

Tools for Quick Increments: Web-Based Tools (examples)

- **Active Agenda**'s code generator is a RAD development framework using XML specification files and the PHP development language.
- **Alpha Five** is a commercial RAD development environment for both client and web-server based database driven applications. This tool is typically distributed with commercial packages such as Microsoft Access and FileMaker.
- **Axiom CMS** is an open source web application framework designed to foster rapid development through the use of XML/Axis (Java, Python) and Java. Tools such as the Axiom CMS and Inspector are written to aid in application development.
- **Base** is a client-server framework for both client and server-side development in the .NET environment.
- **CakePHP** is a RAD development framework using PHP development language.
- **Clio** is an open source rapid application development environment for web-based database driven application development. It places emphasis on code generation technology to provide ASP.NET, PHP, JSP, Hibernate, CoreFusion and Perl language support.
- **J2SD Framework** is an open source, protocol-oriented web application framework licensed under the New BSD License.
- **J2SP** is an open source web application framework, written in Python, which loosely follows the model-view-controller design pattern.
- **IBM Dynamic Business Developer Extension** is a cross-platform, rapid application development IDE for creating enterprise web applications and services for Windows, Unix, Macintosh, Linux, and Sun Solaris.
- **NConstruct** is Windows and Web rapid enterprise application development tool and environment for .NET framework. It supports Oracle database, Microsoft SQL Server databases and Microsoft Access.
- **EASYProcess** is a .NET based RAD for the creation of web portal interfaces, work flows, data integration, reporting and web services largely focused on the JD Edwards community.
- **Softwell Maker** is a cross-platform deployment component allowing publication into almost any Java enabled system.
Tools for Quick Increments: Cross Platform Tools (examples)

- **Boa Constructor** is a cross-platform, wxPython based Python RAD IDE
- **Code::Blocks** is a cross-platform C/C++ RAD IDE using wxWidgets, the latest developmental builds have a built-in form designer wxSmith, so it serves as a replacement for Borland C++ Builder and Microsoft Visual C++/MFC now.
- **HyenaNext** is a freeware cross-platform software development system for Macintosh OS X & OS 9, and Microsoft Windows XP & Vista. It has many similarities with Boa Constructor and can compile to both stand alone applications and stacks for the cross-platform HyperNext Player.
- **IBM Rational Business Developer: Extension** is a cross-platform, Rapid Application Development IDE for creating enterprise and web applications and services for Windows, Linux, Unix (Solaris, HPUX, AIX), System z and System i.
- **IBM Rational Application Developer** is a cross-platform, Rapid Application Development IDE for creating enterprise and web applications and services for multiple Linux and Unix (Solaris, HPUX, AIX).
- **LANSa** is a development environment for generating applications on multiple computer systems. The main feature of the LANSa environment is the LANSa RDML language. It is classified as a 4GL (4th generation computing language), it runs on many systems including MS Windows, Unix, and Linux. In its first release in 1987, the RDML language was known as Lambda.
- **Language** is a cross-platform IDE similar to Borland Delphi.
- **i-Motor** is a Software Development tool which automates application development and rapidly creates enterprise-class Web applications over any database or platform.
- **JDeveloper** is a cross-platform, RAD IDE for creating visual desktop, mobile, web, and SOA applications for Linux, Windows and Mac OS X. The IDE currently supports Java, Ruby, PHP, JavaScript and C/C++ programming languages.
- **Omni Studio** is a cross-platform, Rapid Application Development tool or IDE for creating enterprise and web applications for Windows, Linux, Solaris, and Mac OS X.
- **OpenERP** is a RAD framework in python.
- **OpenSUSE** is a cross-platform IDE for Windows/Linux. Windows with embedded SQL support.
- **Paradigm (or an open-source version iParadigm) is a cross-platform (Windows, Unix, Linux, GUI, Web) cross-database RAD toolkit for development of web and interndatabase ETL applications.
- **Pfaffian** is a cross-platform IDE for creating desktop applications for Windows, Linux and Mac OS X.
- **Runtime Revolution** is a cross-platform RAD which creates desktop applications for Mac Classic, Mac OS X, Windows 98/Me/XP/Vista, and various flavors of Linux.
- **Web Dynpro** is SAP's RAD to create web applications connected to function modules in mySAP ERP.
- **Swift** is a cross-platform IDE for creating Ruby on Rails web applications.
- **Servoy** is a cross-platform application development and deployment environment. Servoy consists of a GUI designer, event-driven and server driven scripts through JavaScript. Servoy allows applications to be deployed to both a native Smart client / Rich client and to a pure HTML Web client from the same codebase and user interface.
- **WideStudio** is an open source integrated development environment for desktop applications purely.
- **J2EE** is a cross-platform, Rapid Application Development IDE for creating enterprise and desktop applications in C/C++ on Windows, Linux, and Mac.
- **CA Plex** is a software development tool that combines the techniques of model-based development, patterns and code generation to accelerate the delivery and maintenance of multi-platform, distributed business applications.

The Web Engineering (WebE) Process

Additional Information
Web Engineering: Umbrella Activities

- Background activities which occur in parallel with the main development activities.

- Equally important to the success of a project

- Many umbrella activities can be defined. But only four are crucial for a successful Web engineering project:
  1. Project management. Tracks and monitors progress as an increment is engineered.
  2. Quality assurance. Defines and conducts those tasks that help ensure that each work product and the deployed increment exhibits quality.
  3. Change management. Manages the effects of change as each increment is engineered, integrating tools that assist in the management of all WebApp content.
  4. Risk management. Considers project and technical risks as an increment is engineered.