What is the course about?

- Web Engineering
  - Web application design and development
  - Requirement engineering for Web applications
  - Testing
- Web Technologies
  - HTML, JSP, AJAX, …
- Project Management
  - CVS/SVN, JUnit …
What is this course about?

• The World Wide Web has become a major delivery platform for information resources. Many applications continue to be developed in an ad-hoc way, contributing to problems of usability, maintainability, quality and reliability. This course examines systematic, disciplined and quantifiable approaches to developing high-quality, reliable and usable web applications. The course introduces the methodologies, techniques and tools that support their design, development, evolution, and evaluation.

Course Goals

• The goals of the course are as follows:
  – to be able to analyze and design comprehensive Web application.
  – to learn and use some of the languages currently used to manipulate information on the World Wide Web – i.e. Java and Javascript.
  – to learn techniques and evaluation metrics for ensuring the proper operability, maintenance and security of a web application.
Course Organization

- Course is organized into:
  - 14 lectures (7 blocks of 2 lectures)
  - 1 exam
  - Bi-weekly starting Oct. 5, 2009
- The lecture(r):
  - katharina.siopaes@sti2.at
    - Ask for f2f meeting by email
    - RR 20
  - One or two guest lectures on special topics will be organized... this may cause changes in the schedule
- Web Engineering is a lecture, hence attendance is not required.
- Exam
  - Final exam at the end of the semester
  - Optional: assignments where you can earn points that will be added to the points in the final exam.
  - However, without doing any assignment it is still possible to get grade 1 in the final exam (given good work, of course ☺).

Course Material

- Book
  - Web Engineering, Wiley & Sons.
- Web site
  - http://www.sti-innsbruck.at/teaching
  - Slides will be published online after each lecture
  - News and changes of dates, etc. will be announced here. Please check regularly.
  - Additional reading published on website
- Mailing list
  - https://lists.sti2.at/mailman/listinfo/webengineering20092010
  - Please subscribe.
What is the course structure?

Mo 05.10.2009 14:15-18:00: Rechnerraum 20
Mo 19.10.2009 13:15-17:00: Rechnerraum 20
Mo 16.11.2009 14:15-18:00: Rechnerraum 20
Mo 30.11.2009 14:15-18:00: Rechnerraum 20
Mo 14.12.2009 14:15-18:00: Rechnerraum 20
Mo 18.01.2010 14:15-18:00: Rechnerraum 20
Mo 01.02.2010 14:15-18:00: Rechnerraum 20

14.15 – 15.45
16.00 – 17.30

Assignments

• Some theoretic work
  – Prepare a presentation on some topic
  – Write an essay on some topic
• ... and some practical work
  – Write some code
  – Make a SW review
• Some to be performed by single students
• ... some by group of students
• Grades vary from 1 point to 8 point, according to complexity.
• Points will be added to the score in the final exam.
Assignments – Ethical Aspects

• Homework is “solo” effort
  – Undue cooperation will result in 0 points for all
  – Your original content is what counts
  – It can be supported by external sources
• Reuse of content from external sources
  – Scientific reference rules – always attribute the source
  – Plagiarism will be punished by 0 point result

Assignments – Delivery

• Emailed to me before the start of next class
  – Katharina.siorpaes@sti2.at
  – Late homework accepted for one more week
  – Email subject: Web Engineering Assignment X
  – Differentiate other emails, esp. questions
    • Send them on the mailing list
  – Receipt of assignment will be acknowledged
  – But it does not mean I checked anything
• Naming of attachments (zip them)
  – FirstnameLastname_1.zip
  – For software releases include a README.TXT
Assignments - Grading

• In 1 or 2 week after delivery
  – Feedback emails with scores and comments
  – Scores will be on website

Why do we need Web Engineering?

INTRODUCTION
What is Web Engineering?

• “The application of systematic and quantifiable approaches to cost-effective analysis, design, implementation, testing, operation, and maintenance of high-quality Web applications.” – Kappel et al.
• Extends Software Engineering to Web applications, but with Web-centric approaches.
  – And other relevant contributions from many disciplines

Defining Web Applications

• Unlike traditional software, the Web serves as both development & user platform.
• A Web application is a system that utilizes W3C standards & technologies to deliver Web-specific resources to clients (typically) through a browser.
  – Kind of …
• 2 aspects:
  – Software: static pages are not a Web application
  – User-interface: Web services are not a Web application
The Case for Web Engineering

- Application development on the Web remains largely *ad hoc*.
  - Spontaneous, one-time events
  - Individual experience
  - Re-use does not go beyond copy/paste
  - Little or no documentation for code/design
- Short-term savings lead to long-term problems in operation, maintenance, usability, etc.
- Because Web apps are so interdependent, the problem is compounded.

The Case for Web Engineering II

- Root Causes of poor design
  - Development is regarded as an authoring activity.
  - Development is “easy”.
  - Techniques that should not be used are misapplied.
  - Software engineering techniques are applied.
  - Techniques that should be used are not.
  - HCI is often not considered ...
- Particularly alarming given…
  - Most projects are now Web-based
  - More “mission-critical” apps moving to the Web
The Case for Web Engineering III

- Top project pitfalls (Cutter, 2000)
  - 84% - Failure to meet business objectives
  - 79% - Project schedule delays
  - 63% - Budget overrun
  - 53% - Lack of functionality

- Web Engineering’s solution:
  - Clearly defined goals & objectives
  - Systematic, phased development
  - Careful planning
  - Iterative & continuous auditing of the entire process

Quelle: Cutter Consortium, Poor Project Management
Number-one Problem of Outsourced E-projects, Cutter Research Briefs, November, 2000,

The Case for Web Engineering IV

- Web Technologies pose new restriction to software development
  - HTTP
  - URL
  - HTML

- HTTP is stateless
  - unless you use tricks

- Web is based on the pull mechanism
  - unless you use tricks

- Why can not change this? Tricks are not good!
  - It's a trade off…
Principles

1. Clearly defined goals and requirements
2. Systematic development of a Web application in phases
3. Careful definition of these phases
4. Continuous monitoring of the development process

Categories of Web Applications
Categories of Web applications

- The development can be started in any category.
- Newer categories show a higher complexity.
  - Older categories are not necessarily replaced by newer ones.
  - Each category has its major application area.
- Complex applications often fit into several categories.
  Example: online shopping mall
  - Integration of providers of various services.
  - Various search possibilities.
  - Monitoring of transactions.
  - Online auctions.
- Many traditional areas are covered (e.g. Online banking).
- Creation of novel areas (e.g. Location-based services).
Document-Centric Web sites

• Precursors to Web applications
• Websites are provided in the form of static HTML that are sent as a reply to a request from the client.
• Manual updates
• Pros
  – Simple, stable, short response times
• Cons
  – High management costs for frequent updates & large collections
  – More prone to inconsistent/redundant info
• Examples
  – Static homepages
  – Company homepages

Interactive Web applications

• The Common Gateway Interface
  – http://hoohoo.ncsa.uiuc.edu/cgi/interface.html
• “The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers. A plain HTML document that the Web daemon retrieves is static, which means it exists in a constant state: a text file that doesn’t change. A CGI program, on the other hand, is executed in real-time, so that it can output dynamic information.”
## Interactive Web applications

- CGI ...
- Simple interactivity
- Dynamic page creation
- Examples
  - Virtual exhibitions
  - News sites
  - Online train schedule systems

## Transactional Web applications

- More interactivity
  - Z.B. Modifications by user are allowed
- Database systems allow
  - An efficient and consistent data management
  - Structured queries
- Examples:
  - Online banking
  - E-Shopping
  - Reservation systems
**Workflow-Based Applications**

- Designed to handle business processes across departments, organizations & enterprises
- Web services provide interoperability.
- Requirement:
  - Processes must be structured!
- Challenges:
  - Complexity of all services
  - Autonomy of participating parties
  - Robustness and flexibility of business processes
- Business logic defines the structure
- The role of Web services
  - Interoperability
  - Loosely-coupled
  - Standards-based
- Examples: B2B & e-Government

---

**Collaborative Web applications**

- Unstructured, cooperative environments with a high communication need.
- Interpersonal communication is paramount.
- Examples:
  - Shared workspaces (Wikis, Office Live, Google Docs)
  - Support of meetings and decision making, e.g. chat rooms
  - Shared schedule management
  - E-learning platforms
  - Collaborative tagging systems
  - ....
Portal-Oriented

- Single points-of-entry to heterogeneous information
- General portals for access to Web
  - Browser, such as Firefox or IE
  - Search services, such as Google or Yahoo!
  - Onlineservices such as AOL
  - Media groups
- Specialized portals
  - Business portals
    - employee intranet to share information
  - Marketplace portals
    - horizontal: B2C and B2B
    - Vertical: unifying companies in a sector
  - Community portals
    - Target groups
    - Interactivity and foster customer loyalty
    - One-to-one marketing

Ubiquitous

- Customized services delivered anywhere via multiple devices. Access everywhere at any time.
- Example
  - Menus for users that enter a restaurant between 11am and 2pm.
- Requirement: extraction of context information.
- HCI is critical
  - Limitations of devices (screen size, bandwidth?)
  - Context of use
- Still an emerging field; most devices have single focus:
  - Personalization
  - Location-aware
  - Multi-platform delivery
Semantic Web

• Berners-Lee: Information on the Web should be readable to machines, as well as humans.
• Using metadata and ontologies to facilitate knowledge management across the WWW.
• Content syndication (RSS, Atom) promotes re-use of knowledge
• Is the Semantic Web even possible?
• Authors devote a chapter to the Semantic Web, but we will not focus on it in this course (see lecture Semantic Web!).

Characteristics of Web Apps

• Web applications have several characteristics, that
  – Are not evident in traditional applications (e.g. Non-linear navigation).
  – Are strongly evident in Web applications (e.g. Frequency of changes).
• Example
  – Transactional applications like e-commerce systems have to focus more on consistency than information providers of digital libraries.
• Often times, these characteristics are the reason why concepts, methods, techniques, and tools of traditional software engineering
  – Only work in an adapted way
  – Are not suitable at all
Characteristics of Web Apps

Characteristics - Product

• The “building blocks” of a Web application
  – Content
  – Hypertext
  – Presentation
• Content
  – Document-centric character and multimedia (# of dimensions?)
    • Presentation in the form of tables, text, graphics, animations, video, audio
    • Communication of contents
    • Didactic presentation of contents
    • Usability challenge!
  – Quality demands
    • Up-to-dateness, consistency, reliability, volume
    • Price or availability of products
    • Critical factor for acceptance!
Characteristics - Product

- Navigation Structure (Hypertext)
  - Non-linearity
    - Difference to traditional software: possibility of systematic reading (e.g. Guided tour)
    - Moving in information space depending on various factors, such as interests.
    - Challenge for authors!
  - Disorientation and cognitive overload
    - Disorientation: Loss of sense of direction and location in a non-linear document
    - Cognitive overload: Higher effort and concentration to maintain overview of various pathes

Characteristics - Product

- User interface (Presentation)
  - Aesthetics
    - Look and feel
    - Trends
  - Self-explanatory design
    - Usage without documentation
    - Consistent user logic
Characteristics - Usage

- Much greater diversity compared to traditional applications
- Social Context (Users)
  - Spontaneity of users
    - Unknown number of users
    - Scalability
  - Multi-cultural
    - Development for various user groups
    - Challenge: requirements analysis (finding representative group of users)
- Technical Context (network and devices)
  - Quality-of-Service
  - Multi-platform delivery
- Natural Context (place and time)
  - Globality
    - Internationalisation regarding regional, cultural, and linguistic differences
    - Location-based services
    - Increased security demands
  - Availability
    - 24/7
    - Time-dependent services

Characteristics - Development

- Project members
  - Multiple disciplines
    - Print publishing, software development, marketing and computer science, art and technology
    - IT experts, hypertext, designer and domain experts
  - Developers are often young and inexperienced without interest in old and proven methods
  - Community development (open source)
- Technical Infrastructure
  - Inhomogeneity
    - Web server (configurable) vs. Client (no influence)
  - Immaturity
    - Time to market pressure increases
    - Malfunctioning components
- Process
  - Flexibility
    - No fixed schemata
  - Parallelism
    - Sub-applications / phases
- Integration
  - Internal with existing legacy systems
  - External with contents and services of other Web applications
    - High number of changing sources
    - High autonomy
    - Little detail information
    - Heterogeneity on various levels
Characteristics - Evolution

- All the above mentioned dimension are governed by the evolution principle
  - Continuous change
  - Competitive pressure
  - Fast pace
- Software Engineering: evolution is planned in a constant number of release version
- Web Engineering: evolution is continuous
  - Nowadays this is becoming true also for SE… it’s a loop, when a discipline overlaps its ancestor, the ancestor learn something back!

Key Knowledge Areas

- Software Engineering
  - Process
  - Design
  - Implementation
  - Test
  - Operation
  - Maintenance
- Network Engineering
  - Physical Layer
  - Internet Layer
  - Transport Layer
  - Performance
- Hypermedia
  - Design & Structure
  - Information Space
  - Navigation
  - Visualization
  - Usability
  - Collaboration
- Information Systems
  - Data Design, ER...
  - RDBMS
  - Query Languages
  - Storage Devices: FS...
  - Others...
Standardization I

W3C
- Produces W3C Recommendations on Web protocols
- Managed approach to developments
- Protocols initially developed by W3C members
- Decisions made by W3C, influenced by member and public review

ISO
- Produces ISO Standards
- Can be slow moving and bureaucratic
- Produce robust standards

IETF
- Produces Internet Drafts on Internet protocols
- Bottom-up approach to developments
- Protocols developed by interested individuals
- "Rough consensus and working code"

Proprietary
- De facto standards
- Often initially appealing (cf PowerPoint, PDF)
- May emerge as standards

PNG
HTML
HTTP

Standardization II

- Standards are important, especially for national initiatives and other large-scale services
  - More easy to integrate different projects if they adopt standards
- Proprietary solutions are often tempting because:
  - They are available
  - They are often well-marketed and well-supported
  - They may become standardized
  - Solutions based on standards may not be properly supported by applications
That’s almost all for day…

**WRAP-UP**

**Things to keep in mind**
(or summary)

- Web Engineering is not about HTML and JavaScript
  - Like Software Engineering is not about C or Java!
- It aims at systematic development of Web applications according to a specific methodology
- Web Engineering is not just Software Engineering for the Web
- Web Engineering ask for multidisciplinary approach
- Standards are important in Web like in all the other Engineering fields
Bibliography

• Mandatory reading
  – Web Engineering
    • Chapter 1

• Suggested
  – Google "Web Engineering"

Assignment

• Pick up one of the categories introduced today and provide a small essay on it
  – 2 points
  – 1 page Word, Times 11pt

  – Figures do not count for the space
Questions?