Web Design

Week 6

MSDN Account

All the accounts are created. If students did not get an email then they already had an account.

All you need to do is to give your students this URL
http://msdn06.e-academy.com/elms/Storefront/Home.aspx?campus=csun_e_ceng

and have them click on the login and forgotten my password link. Then put in their CSUN gmail account and it will send them their account info.

Or you could have them email me (Mark Siegmund [msiegmund@gmail.com]).

Announcement

- Midterm I
  - Monday March, 7th
- Scope
  - Ch. 1, 2, 3, 4 and Ch. 6 of the text book
  - Ch. 1, 2 and 3 of the lab book
  - Bring a scantron

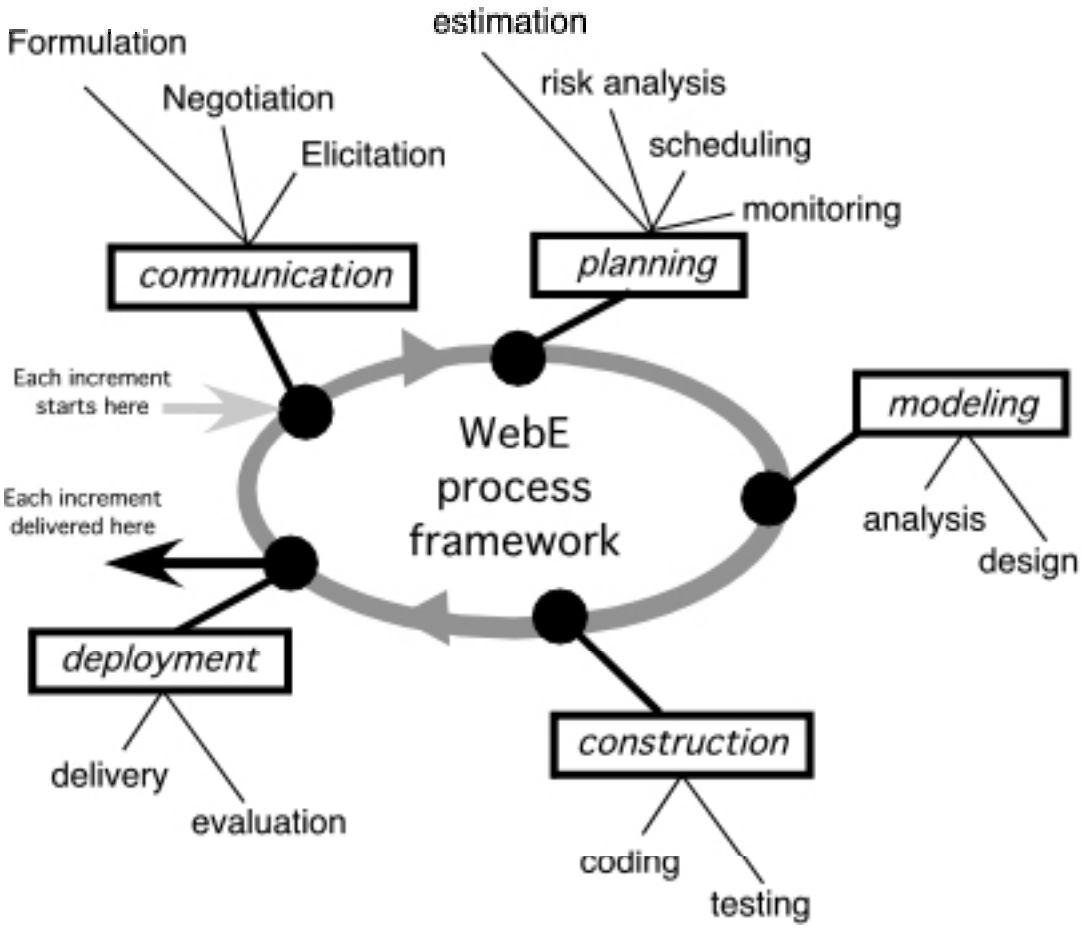
Agenda (Lecture)

- Overview of Web Design

Agenda (Lab)

- Weekly progress report
- Homework/Lab assignments

WebE Process Activities & Actions



WebApp Design – I

- Jakob Nielsen states: “There are essentially two basic approaches to design: the artistic ideal of expressing yourself and the engineering ideal of solving a problem for a customer.”

WebApp Design – II

- Even today, some proponents of agile software development use WebApps as poster children for the development of applications based on “limited design.”
- However --
  - When content and function are complex
  - when the size of the WebApp encompasses hundreds of content objects, functions, and analysis classes
  - when multiple people become involved in the design
  - when the success of the WebApp will have a direct impact on the success of the business,
- Design cannot and should not be taken lightly.

WebApp Design

- The design model encompasses content, aesthetics, architecture, interface, navigation, and component-level design issues.
- The design model provides sufficient information for the Web engineering team to construct the final WebApp
- alternative solutions are considered, and the degree to which the current design model will lead to an effective implementation is also assessed

Design Goals - I

- **Simplicity.** Although it may seem old-fashioned, the aphorism "all things in moderation" applies to WebApps. Rather than *feature-bloat*, it is better to strive for moderation and simplicity.

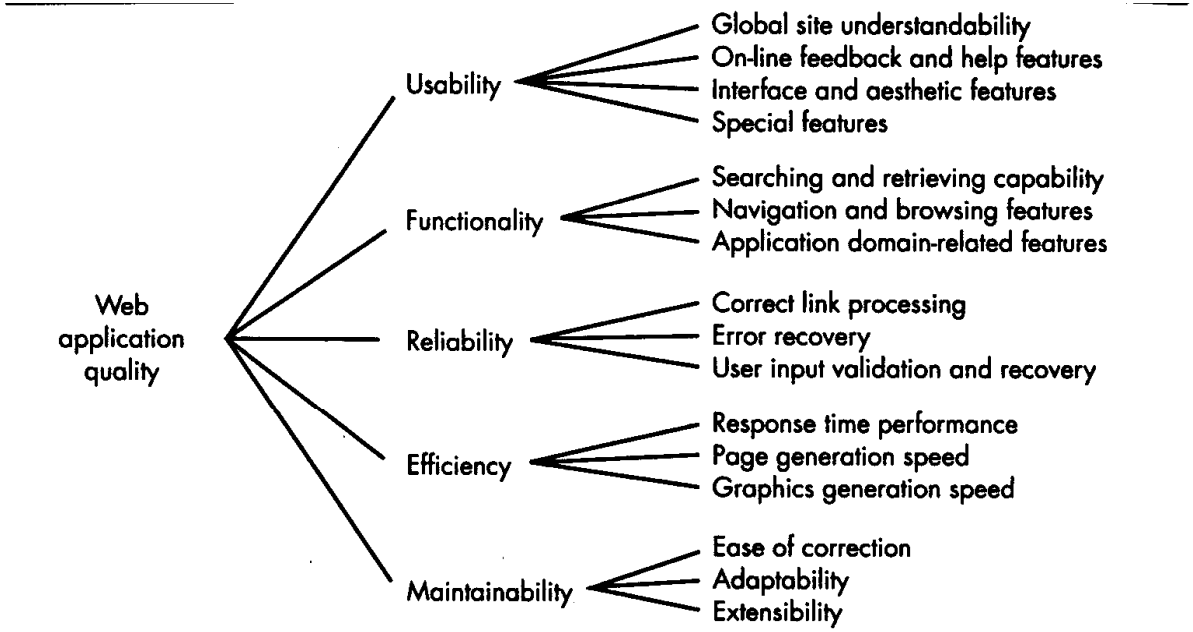
- **Consistency.**
  - Content should be constructed consistently
  - Graphic design (aesthetics) should present a consistent look
  - Architectural design should establish templates that lead to a consistent hypermedia navigation
  - Navigation mechanisms should be used consistently

- **Identity.** The aesthetic, interface, and navigational design of a WebApp must be consistent with the application domain for which it is to be built.

Design Goals - II

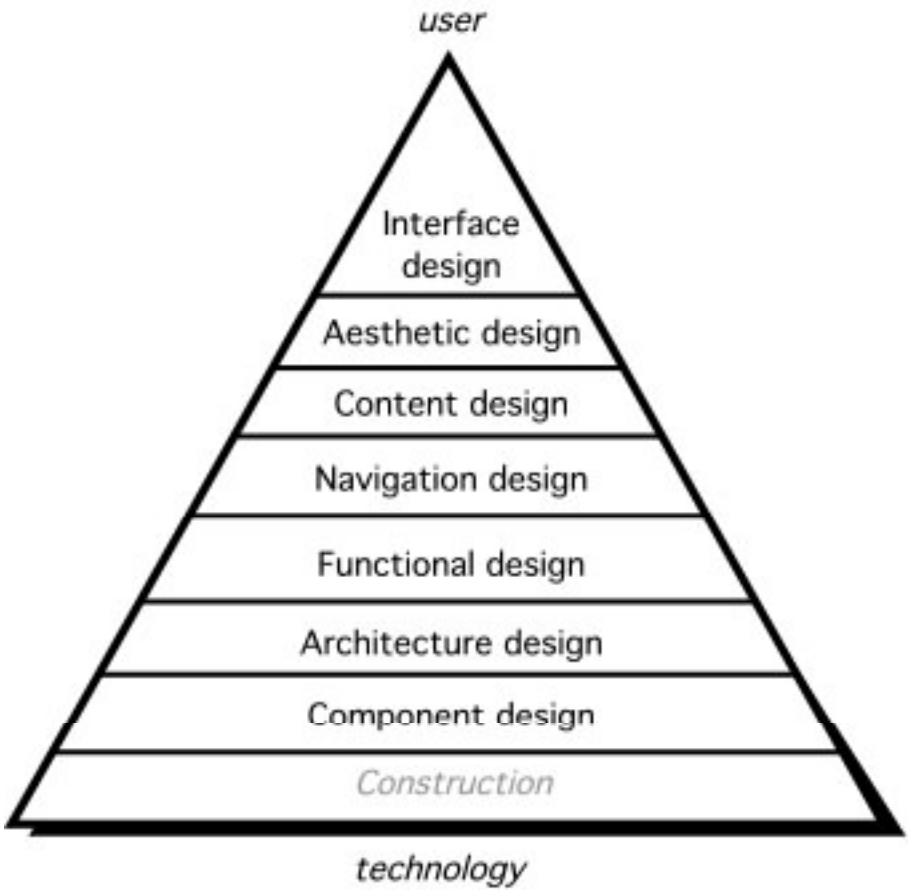
- **Robustness.** The user expects robust content and functions that are relevant to the user's needs.
- **Navigability.** users should be able to understand how to move about the WebApp without having to search for navigation links or instructions.
- **Visual appeal.** design characteristics (e.g., the look and feel of content, interface layout, color coordination, the balance of text, graphics and other media, and navigation mechanisms) contribute to visual appeal.
- **Compatibility.** Most WebApps will be used in a variety of environments (e.g., different hardware, Internet connection types, operating systems, and browsers) and must be designed to be compatible with each

Design & WebApp Quality

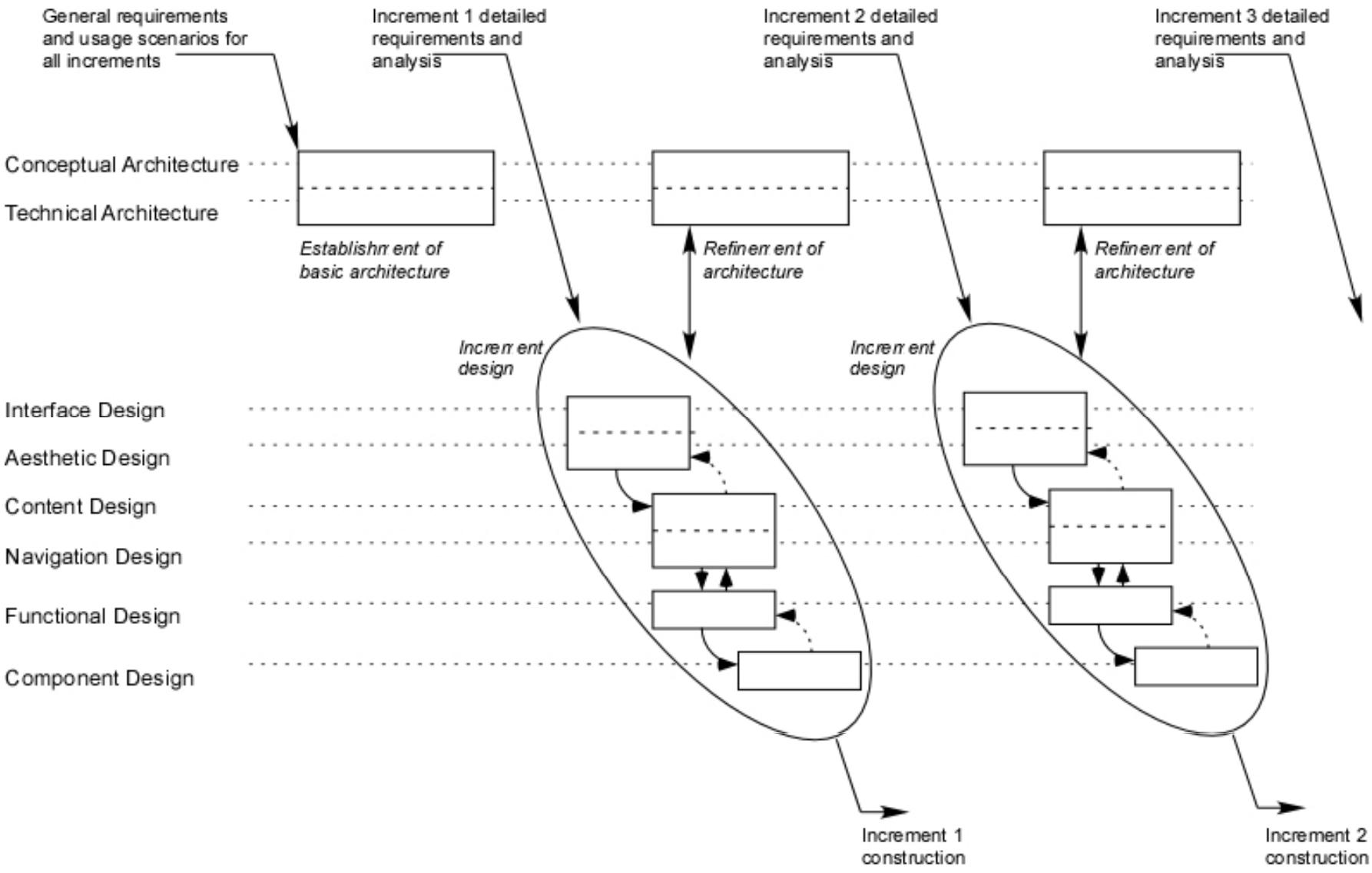


- Try using the design IQ checklist!

Design Actions



Design Process



Conceptual Architecture

- Provides an overall structure for the WebApp design
  - Affects all later increments – so important for it to developed in the context of the full set of *likely* increments
- Represents the major functional and information components for the WebApp and describes how these will fit together
  - Depends on the nature of the WebApp, but in every case, it should ensure a sound integration between the WebApp information and the WebApp functionality.

Developing the architecture-I

- How do we achieve an effective balance between information and functionality in the conceptual architecture?
- A good place to start is with workflows or functional scenarios (which are an expression of the system functionality) and information flows

Developing the architecture-II

- As a simple example, consider the following set of key functionalities for **SafeHomeAssured.com**
  - Provide product quotation
  - Process security system order
  - Process user data
  - Create user profile
  - Draw user space layout
  - Recommend security system for layout
  - Process monitoring order
  - Get and display account info
  - Get and display monitoring info
  - Customer service functions (to be defined later)
  - Tech support functions (to be defined later)

Developing the architecture-III

- From these key functionalities we can identify the following partial list of *functional subsystems:*
  - **UserManagement.** Manages all user functions, including user registration, authentication and profiling, user-specific content, and interface adaptation and customization.
  - **ProductManagement**. Handles all product information, including pricing models and content management.
  - **OrderHandling**. Supports the management of customers' orders.
  - **AccountAdministration**. Manages customers' accounts, including invoicing and payment.
  - **SecuritySystemSupport**. Manages users' space layout models and recommends security layouts.
  - **SecuritySystemMonitoring**. Monitors customers' security systems and handles security events.

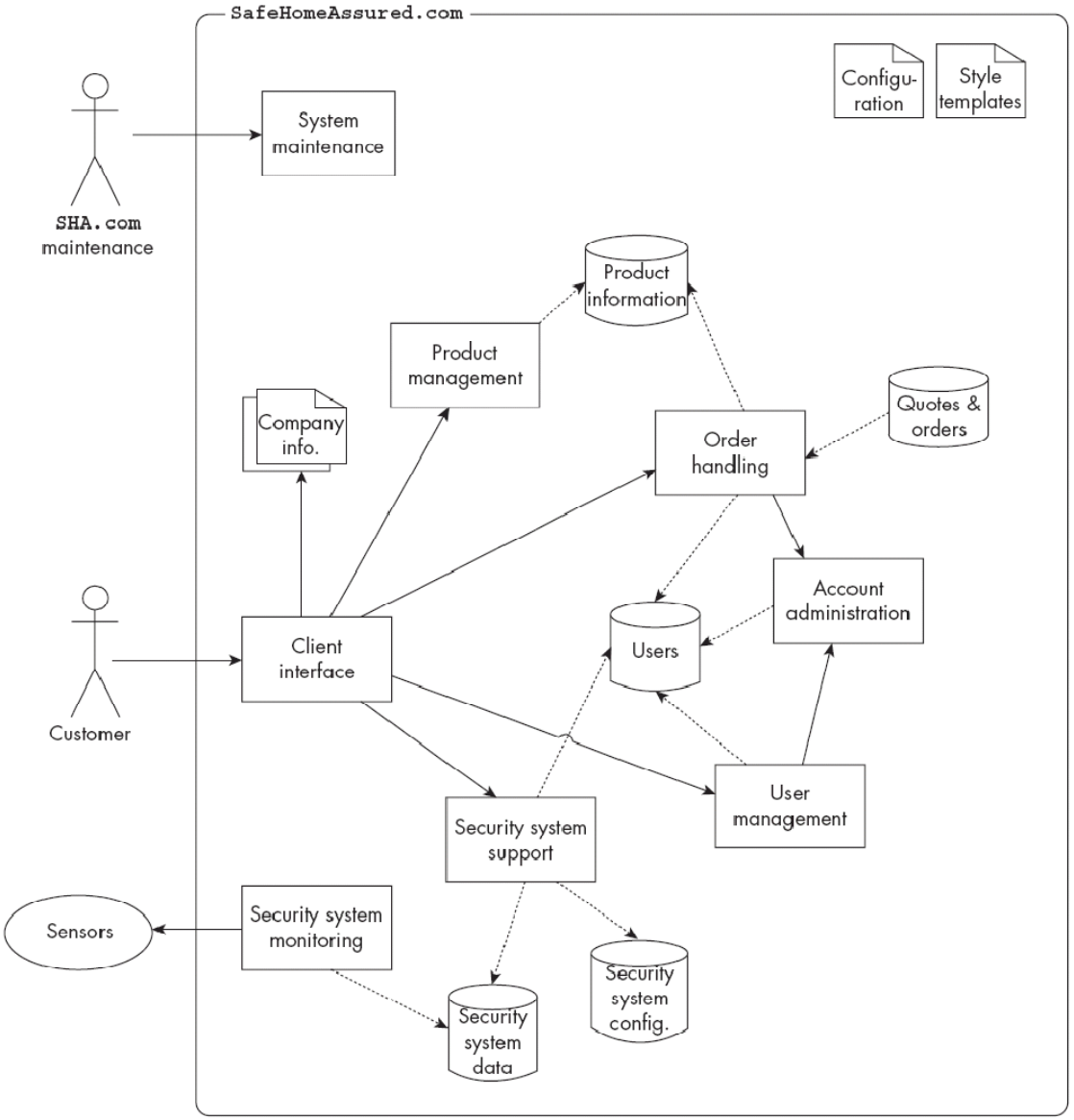
Developing the architecture-IV

- And, of course, there are overall management subsystems:
  - **ClientInterface**. Provides the interface between users and the other subsystems, as required to satisfy users needs.
  - **SystemMaintenance**. Provides maintenance functionality, such as database cleaning.

Technical Architecture

- Shows how the conceptual architecture can be mapped into specific technical components
- Any decision made about how one component might map into the technical architecture will affect the decisions about other components
  - For example, the WebE team may choose to design **SafeHomeAssured.com** in a way that stores product information as XML files. Later, the team discovers that the content management system doesn't easily support access to XML content, but rather assumes that the content will be stored in a conventional relational database. One component of the technical architecture conflicts with constraints imposed by another component.

Architecture



“Talent War Crunches Start-Ups” – 2/28/2011 WSJ