“Communication” Phase

Week 3

Agenda (Lecture)

- Web Engineering process
- “Communicate phase” or equivalent phase
- Web 1.0 and Web 2.0 examples

Agenda (Lab)

- Project proposal
- Weekly progress report
- Lab/homework assignments

Weekly Progress Report

- From now on, each team is required to submit a weekly project progress report to the instructor by the end of the Wednesday lab session. The report should be typed up and should include
  - The team name and a list of team members' names
  - A list of activities that have done in the previous week and the names of the corresponding contributors
  - A list of activities that will be conducted next week

Team Homework Assignment #5

- Study the modeling phase of the Web Engineering Process (Ch. 6 or related materials) and prepare for presentation slides.
- Due date is 7:00 pm, February 14th

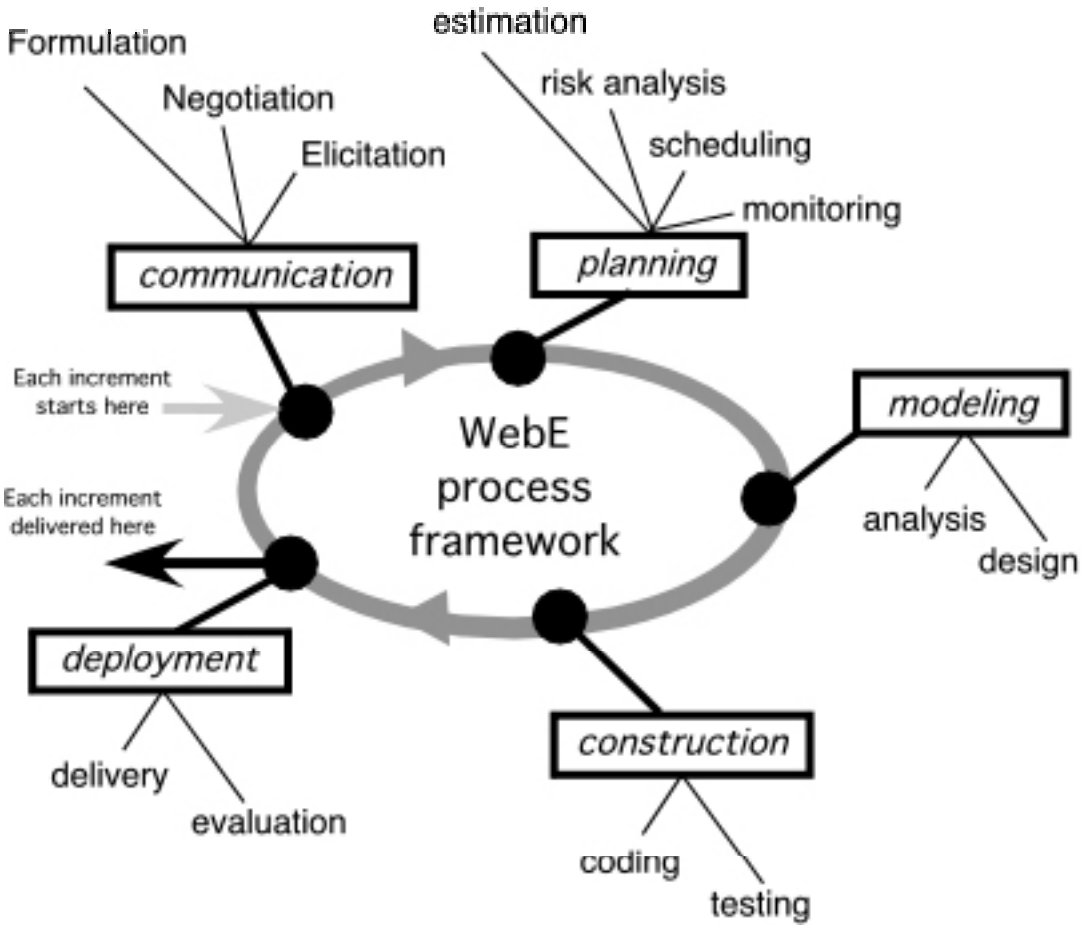
Team Lab Assignment #3

- Defining the *needs* and *scope* of your project
  - Submit the first draft of the document that includes project needs and scope (refer to page 10 of this power point file)
  - Make slides for presentation
- Due date
  - The beginning of the 2/14 lab session

Team Lab Assignment #4

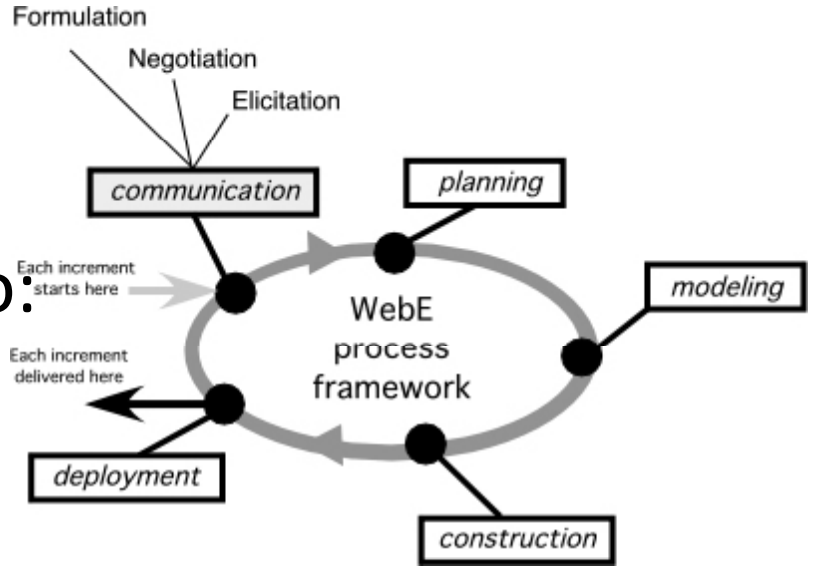
- Submit the first version of a use case diagram (or other equivalent analysis modeling diagram) for your group project
  - Submit a use case diagram (other equivalent analysis modeling diagram)
  - Make slides for presentation
- Due date
  - The beginning of the 2/14 lab session

WebE Process Activities & Actions



Chapter 4: *Communication*

- *Understand the problem before you begin to solve it, and be sure that the solution you conceive is one that people really want*
- To do this, you'll need to:
  - *Formulate*
  - *Elicitate*
  - *Negotiate*



Formulation

- Focuses on defining the project needs and scope
  - begins with the identification of a business need
  - moves into a description of WebApp objectives
  - defines major WebApp features, and establishes a basis for the elicitation action that follows.
  - allows stakeholders and the WebE team to establish a common set of goals and objectives for the creation of each WebApp increment
  - identifies the scope of the development effort and provides a means for determining a successful outcome

What Questions Do We Ask?

- What is the main motivation (business need) for the WebApp?
- What are the objectives that the WebApp must fulfill?
- Who will use the WebApp?

Elicitation

- The intent is to gather detailed requirement collaboratively with all stakeholders

- To do this:
  - A meeting (either physical or virtual) is conducted and attended by all stakeholders.
  - Rules for preparation and participation are established.
  - An agenda is suggested that is formal enough to cover all important points but informal enough to encourage the free flow of ideas.
  - A facilitator (can be a customer, a Web engineer, or an outsider) controls the meeting.
  - A definition mechanism (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room, or virtual forum) is used.

Elicitation Tasks

- Define user categories, and develop descriptions for each category.
- Define content and functionality using the lists each person prepared.
- Consider specific constraints and performance issues.
- Write user scenarios for each user class.

User Descriptions

- What is the user's overall objective when using the WebApp?
- What is the user's background and sophistication level relative to the content and functionality of the WebApp?
- How will the user arrive at the WebApp?
- What generic WebApp characteristics does the user like and dislike?

Content and Functionality

- Each stakeholder has begun this work by preparing lists of content objects and WebApp functions.
- Once the meeting begins these lists can be:
  - displayed on large sheets of paper pinned to the walls of the room
  - displayed on adhesive-backed sheets stuck to the walls, or
  - written on a whiteboard.
  - posted on an electronic bulletin board, at an internal website, or posted in a chat room environment for review prior to the meeting.
- Ideally, each listed entry should be capable of being manipulated separately so that lists can be combined, entries can be deleted, and additions can be made. At this stage, critique and debate are strictly prohibited.

Constraints and Performance

- *Internal constraints* are best understood by thinking about the technical environment in which the WebApp will reside and the project environment in which the WebApp will be built.
  - *technical environment*—specialized database protocols, the vagaries of different Web browsers, operating system characteristics, and client-server issues
  - *project environment*—available WebE tools, development hardware, software standards, and staff skill levels with various WebE technologies.

Constraints and Performance

- *External constraints* can be enumerated by considering the business and usage environment for the WebApp.
  - Business rules, end-user idiosyncrasies, security demands, privacy issues, run-time performance, interoperability requirements, legal restrictions, and government regulations are but a few of possible external constraints

Capturing Interaction: Use Cases

- Use cases describe how a specific user category (called an *actor*) will interact with the WebApp to accomplish a specific action.
- Use cases are developed iteratively. Only those use cases necessary for the increment to be built are developed during the communication activity for the increment.
- Use cases enable you to:
    - provide the detail necessary for effective planning and modeling activities.
    - help you to understand how users perceive their interaction with the WebApp.
    - help to compartmentalize Web engineering work because they can be organized into WebApp increments.
    - provide important guidance for those who must test the WebApp.

From Use Cases to Increments

- A stack of “cards” that contains one usage scenario or use case per card
  - Each card contains the name of the use case, a brief description, and an *effort indicator*—usually a number between 1 and 4
- The cards are:
  - shuffled into random order
  - distributed to selected stakeholders who are asked to arrange the cards into groupings that reflect how they would like content and functionality (implied by the usage scenarios) to be delivered
- The manner in which cards are grouped is constrained by an *effort maximum M.*
  - No grouping of cards can have a cumulative effort indicator value that is greater than *M,* where *M* is defined by the WebE team and is a function of available resources and the desired delivery time for each increment.

Negotiation

- Ideally, requirements are defined in sufficient detail to proceed
- BUT, in reality, requirements are often contradictory or infeasible (within the context of real-world constraints, such as cost or time).
- Negotiation involves working with the stakeholders to balance functionality, performance, and other product or system characteristics against cost and delivery time.
- The best negotiators strive for a win-win result.
  - it's a good idea to determine each of the stakeholders' "win conditions".

Negotiation

- *Recognize that it's not a competition.* To be successful, both parties have to feel they've won or achieved something. Both will have to compromise.

- *Map out a strategy.* Decide what you'd like to achieve, what the other party wants to achieve, and how you'll go about making both happen.

- *Listen actively.* Don't work on formulating your response while the other party is talking. Listen. It's likely you'll gain knowledge that will help you to better negotiate your position.

Negotiation

- *Focus on the other party's interests.* Don't take hard positions if you want to avoid conflict.
- *Don't let it get personal.* Focus on the problem that needs to be solved.
- *Be creative.* Don't be afraid to think outside of the box if you're at an impasse.
- *Be ready to commit.* Once an agreement has been reached, don't waffle; commit to it and move on.