

CSCI 461  
Senior Project  
Deliverables 9/23/09

#### Copy of Presentation II (Electronic)

The presentation will be a summary of the Specification Document. It will include the major elements and “screen shots” of the intended project.

#### Copy of Specification Document (Paper and Electronic)

#### Specification Document Guidelines\*

The exact content of the specification document will depend on your project. The goal of this document is to completely describe what the system will do.

You can use any of the analysis techniques from your System Analysis and Design class.

- Structured Analysis
- Object-Oriented Analysis
- Combined Features of Both Approaches

Your Specification Document may include

- Scenario-based elements
- Flow-oriented elements
- Class-based elements
- Behavioral elements

#### Objective

1. to describe what the customer requires
2. to establish a basis for the creation of a software design
3. to define a set of requirements that can be validated once the software is built

#### Rules of Thumb

The model should focus on requirements that are visible within the problem or business domain. The level of abstraction should be relatively high. Don't get bogged down in details that try to explain how the system will work.

Each element of the analysis model should add to an overall understanding of the software requirements and provide insight into the information domain, function, and behavior of the system.

Delay consideration of infrastructure and other non-functional models until design.

Minimize coupling throughout the system.

Be certain that the analysis model provides value to all stakeholders.

Keep the model as simple as it can be. Don't add additional diagrams when they provide no new information.. Don't use complete notational forms, when a simple list will do.

## Validating Requirements

Is each requirement consistent with the overall objective for the system/product?

Have all requirements been specified at the proper level of abstraction? That is, do some requirements provide a level of technical detail that is inappropriate at this stage?

Is the requirement really necessary or does it represent an add-on feature that may not be essential to the objective of the system?

Is each requirement bounded and unambiguous?

Does each requirement have attribution? That is, is a source noted for each requirement?

Do any requirements conflict with other requirements?

Is each requirement achievable in the technical environment that will house the system or product?

Is each requirement testable, once implemented?

Does the requirements model property reflect the information, function, and behavior of the system to be built?

Has the requirements model been “partitioned” in a way that exposes progressively more detailed information about the system?

Have requirements patterns been used to simplify the requirements model? Have all patterns been properly validated? Are all patterns consistent with Customer requirements?

\* All information was taken from *Software Engineering A Practitioner's Approach*, sixth edition, by Roger S. Pressman