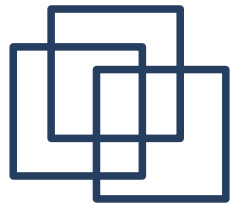


CMSC 128

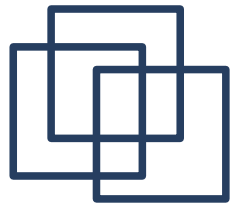
Introduction to Software Engineering Second Semester AY 2007-2008

jachermocilla@uplb.edu.ph



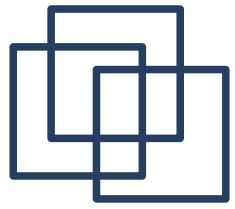
Analysis Concepts

- Complete understanding of software requirements is essential to project success
- Requirements analysis task is a process of discovery, refinement, modeling, and specification
 - Starting point is the software scope, further refined
 - Models of data, information, control flow, and operational behavior are created



Analysis Concepts

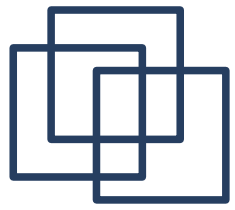
- Both the developer and the customer plays an active role in requirements analysis and specification
 - Customer reformulates software function into concrete detail
 - Developer acts as interrogator, consultant, and problem solver
- Requirements analysis and specification requires heavy communication
 - Misinformation, misinterpretation, ambiguity



Analysis Concepts

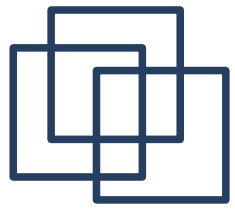
Customer:

'I know you believe you understood what you think I said, but I am not sure you realize that what you heard is not what I meant'



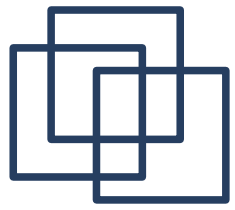
Requirements Analysis

- Bridges the gap between system-level software allocation and software design
 - System engineer: specify software function and performance
 - Analyst: refine software allocation and build models
 - Software Designer: use models to create design



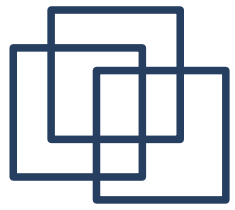
Requirements Analysis

- Areas of effort
 - 1) Problem recognition
 - 2) Evaluation and synthesis
 - 3) Modeling
 - 4) Specification
 - 5) Review



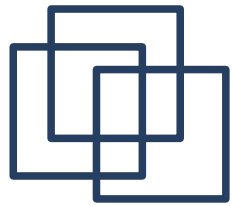
Requirements Analysis

- Problem Recognition
 - Achieved by communicating with the clients
- Problem Evaluation and Synthesis
 - Define all externally observable data objects, evaluate flow and content of information, define and elaborate all software functions, etc.
 - Solution is synthesized based on the result of evaluation



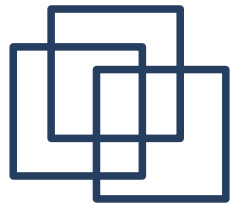
Requirements Analysis

- Problem Evaluation and Synthesis
 - Stops when customer and analyst feel confident that software can be adequately specified
 - Focus on 'what'
- Modeling
 - Data, functions, flow
 - Model serves as foundation of design and creation of the specification



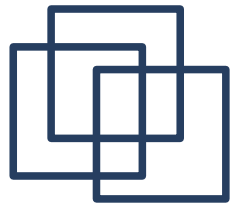
Communication

- Initiating communication
 - Initial interview
 - Ask context-free questions to establish scope
- Facilitated Application Specification Techniques (FAST)
 - Forces developers and customers to work as a team
 - Meeting is conducted in a neutral site with a facilitator
 - Goal is to identify problems and propose solutions



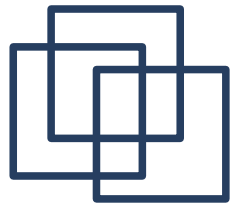
Analysis Principles

- Information domain of the problem must be represented and understood
- Functions that the software is to perform must be defined
- Behavior of software must be represented
- Models created must be partitioned in a manner that uncovers detail in a layered fashion
- Analysis should move from essential to ~~implementation view~~



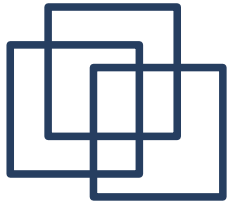
Analysis Principles

- Understand the problem before creating the analysis model
- Develop prototypes to understand how human-machine interaction will occur
- Record the origin of and reason for every requirement
- Use multiple views of requirements
- Prioritize requirements
- Work to eliminate ambiguity



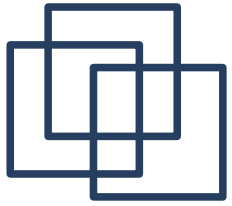
Information Domain

- Contains data and control
- Three views
 - Information content
 - Individual data and control object objects
 - ex. paycheck
 - Information flow
 - Represents the manner in which data and control change as each moves through a system
 - Information structure
 - Internal organization of various data and control items



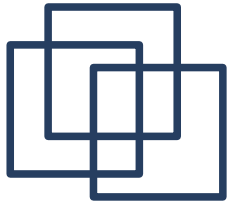
Modeling

- Used to gain a better understanding of the actual entity to be built
- Focus on what the system must do and not how it does it
- Uses graphical notation, portions use text, others use specialized languages
- Functional Models and Behavioral Models



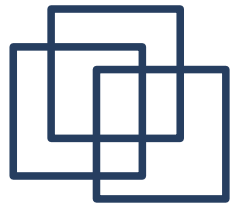
Modeling

- Roles of models
 - Aids in understanding the information, function, and behavior of a system, makes requirements analysis more systematic
 - Models become a focal point of review, key to determination of completeness, consistency, and accuracy of specifications
 - Model becomes the foundation of design



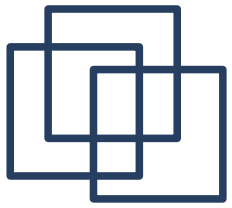
Partitioning

- Decomposes a problem into its constituent parts
- Establish a hierarchical representation of information and function
- Partition uppermost element by
 - Exposing increasing detail by moving vertically
 - Decomposing the problem by moving horizontally



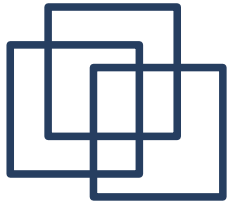
Essential and Implementation View

- Essential view
 - Presents the function to be accomplished and information to be processed without regards to implementation details
- Implementation View
 - Presents real-world manifestations of processing functions and information structures
 - Not necessarily the 'how' but represents the current mode of operation



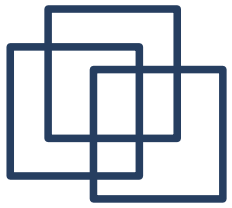
Prototyping

- Model of software to be built is developed
- Throw-away Prototyping
 - When requirements are unstable
 - Requirements are ambiguous
- Evolutionary Prototyping
 - Stable requirements
 - No contradictions in requirements



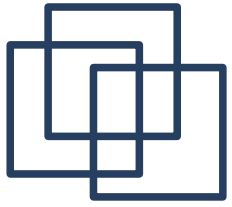
Prototyping

- Methods and Tools
 - 4GT
 - Reusable software components
 - Formal Specification and Prototyping environments



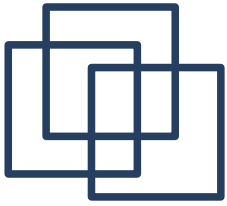
Specification

- May be viewed as a representation process
- Some principles
 - Separate functionality from implementation
 - Establish the context in which software operates
 - Develop cognitive model as perceived by the user community
 - Must be tolerant of incompleteness
 - Amenable to change



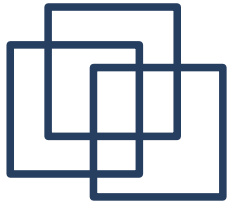
Specification

- Representation
 - Format and content should be relevant to the problem
 - Information contained within the specification should be nested
 - Diagrams and other notational forms should be restricted in number and consistent in use
 - Representations should be revisable



SRS

- Software Requirements Specification
- Produced at the culmination of the analysis task



SRS

I. Introduction

II. Information Description

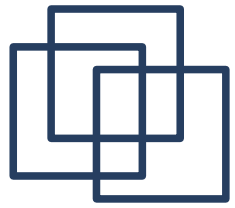
III. Functional Description

IV. Behavioral Description

V. Validation and Criteria

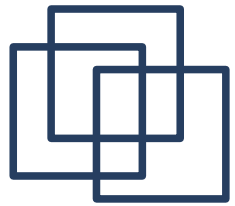
VI. Bibliography

VII. Appendix



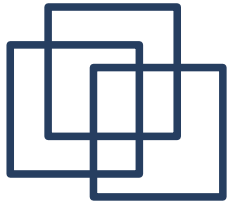
Specification Review

- Conducted by both the developer and customer
- Check to see whether the specification is complete, consistent, and accurate
- Some questions that must be answered
 - Do stated goals and objectives for software remain consistent with system goals and objectives?
 - Have all important interfaces to all system elements been described?



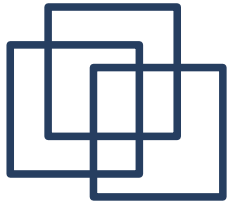
Specification Review

- Some questions that must be answered
 - Is information flow and structure adequately defined?
 - Are design constraints realistic?
 - Have validation criteria been stated in detail?
 - Has the user reviewed the preliminary user's manual or prototype?



Summary

- Requirements analysis is the first technical step in the software engineering process
 - Software scope is refined into a concrete specification
 - Analysis must focus on the information, functional, and behavioral domains of a problem
 - Modeling, partitioning, representation, and specification are important
-



Reference

- Roger S. Pressman. Software Engineering: A Practitioner's Approach, 4th Ed. McGraw-Hill, 1997. Chapter 11