

# REQUIREMENT ENGINEERING

Lecture #18

Lecture by Engr. Sidra

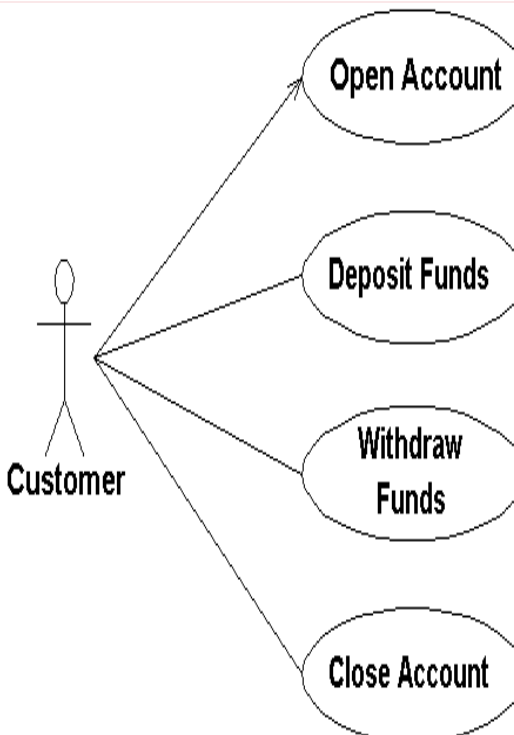
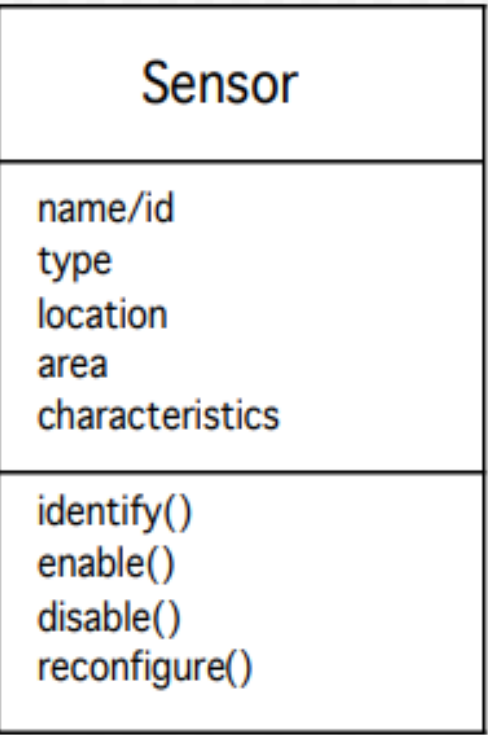
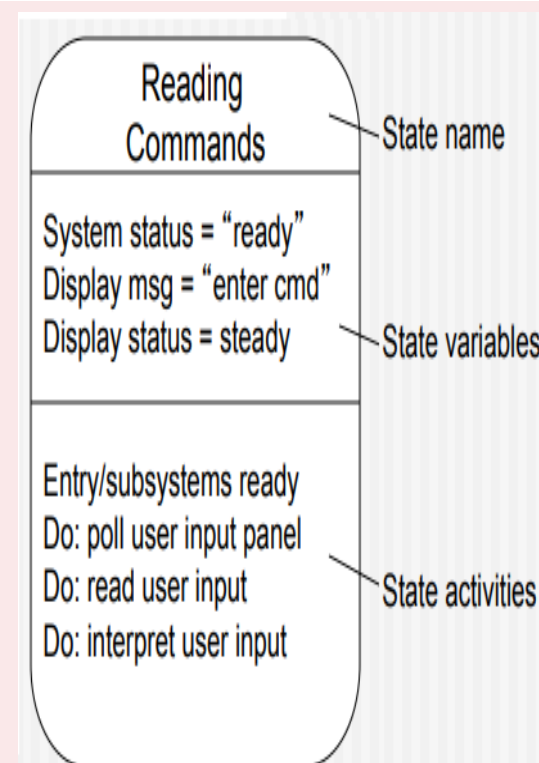
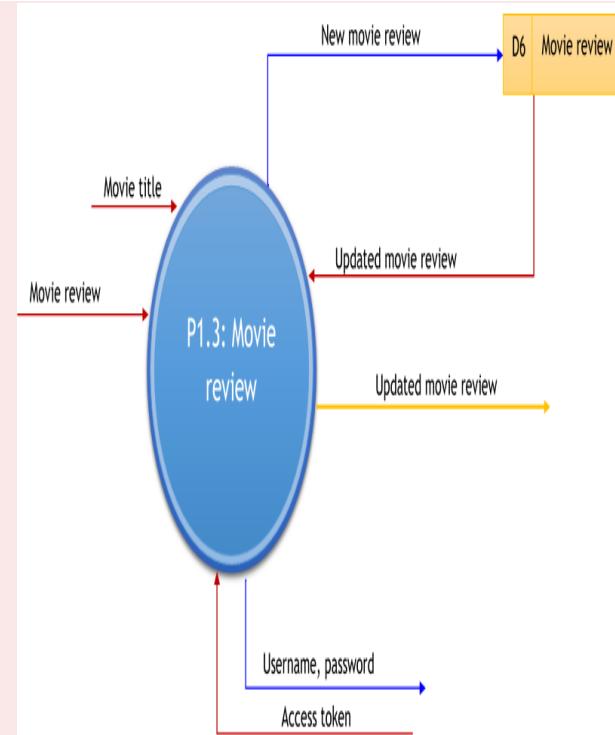


# ELABORATION

- During elaboration, the software engineer takes the information obtained during inception and elicitation and begins to expand and refine it.
- It is an analysis modeling task, The end result is an analysis model that defines the functional, informational, and behavioral model of the problem
- It is easy to argue that the Elaboration task is the most critical activity.
- At the end of this task, the decision on whether or not to commit to the production phases.
- While the process must always accommodate changes it must ensure that the architecture, requirements and plans are stable enough, and the risks sufficiently mitigated to be able to predictably determine the cost and schedule for the completion of the development.



# ELEMENTS OF THE ANALYSIS

Scenario-based elements	Class-based elements	Behavioral elements	Flow-oriented elements
Use-case	Implied by scenarios	State diagram	Data flow diagram
			



# NEGOTIATION

- During negotiation, the software engineer reconciles the conflicts between what the customer wants and what can be achieved given limited business resources
- Requirements are ranked (i.e., prioritized) by the customers, users, and other stakeholders
- Risks associated with each requirement are identified and analyzed
- Rough guesses of development effort are made and used to assess the impact of each requirement on project cost and delivery time
- Using an iterative approach, requirements are eliminated, combined and/or modified so that each party achieves some measure of satisfaction



# SPECIFICATION

- A specification is the final work product produced by the requirements engineer
- It is normally in the form of a software requirements specification
- It serves as the foundation for subsequent software engineering activities
- It describes the function and performance of a computer- based system and the constraints that will govern its development
- It formalizes requirements of the proposed software in both a graphical and textual format





# TYPICAL CONTENTS OF A SOFTWARE REQUIREMENTS SPECIFICATION

- Requirements
  - Software requirements grouped by capabilities (i.e., functions, objects)
  - Software external interface requirements
  - Software internal interface requirements
  - Software internal data requirements
  - Other software requirements (safety, security, privacy, environment, hardware, software, communications, quality, personnel, training, logistics, etc.)
  - Design and implementation constraints
- Qualification provisions to ensure each requirement has been met
  - Demonstration, test, analysis, inspection, etc



# VALIDATION

- During validation, the work products produced as a result of requirements engineering are assessed for quality
- The specification is examined to ensure that
  - all software requirements have been stated unambiguously
  - inconsistencies, omissions, and errors have been detected and corrected
  - the work products conform to the standards established for the process, the project, and the product
- The formal technical review serves as the primary requirements validation mechanism
  - Members include software engineers, customers, users, and other stakeholders
- Prototyping
- Test case Generation



# REQUIREMENTS CHECKING

- Validity.
  - Does the system provide the functions which best support the customer's needs?
- Consistency.
  - Are there any requirements conflicts?
- Completeness.
  - Are all functions required by the customer included?
- Realism.
  - Can the requirements be implemented given available budget and technology
- Verifiability.
  - Can the requirements be checked?





# QUESTIONS TO ASK WHEN VALIDATING REQUIREMENTS

- Is each requirement consistent with the overall objective for the system/product?
- Have all requirements been specified at the proper level? That is, do some requirements provide a level of technical detail that is inappropriate at this stage?
- Is each requirement unambiguous?
- Does each requirement have attribution? That is, is a source (generally, a specific individual) noted for each requirement?
- Do any requirements conflict with other requirements?



# QUESTIONS TO ASK WHEN VALIDATING REQUIREMENTS (CONTINUED)

- Is each requirement achievable in the technical environment that will house the system or product?
- Is each requirement testable, once implemented?
- Approaches: Demonstration, actual test, analysis, or inspection
- Does the requirements model properly 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?



# REQUIREMENTS MANAGEMENT

- During requirements management, the project team performs a set of activities to identify, control, and track requirements and changes to the requirements at any time as the project proceeds
- Each requirement is assigned a unique identifier
- The requirements are then placed into one or more traceability tables
- These tables may be stored in a database that relate features, sources, dependencies, subsystems, and interfaces to the requirements
- A requirements traceability table is also placed at the end of the software requirements specification



# REQUIREMENTS MANAGEMENT PLANNING

- Establishes the level of requirements management detail that is required.
- Requirements management decisions:
  - **Requirements identification** Each requirement must be uniquely identified so that it can be cross-referenced with other requirements.
  - A **change management process** This is the set of activities that assess the impact and cost of changes.
  - **Traceability policies** These policies define the relationships between each requirement and between the requirements and the system design that should be recorded.
  - **Tool support** Tools that may be used range from specialist requirements management systems to spreadsheets and simple database systems.



# REQUIREMENTS CHANGE MANAGEMENT

- Deciding if a requirements change should be accepted
- Problem analysis and change specification
  - During this stage, the problem or the change proposal is analyzed to check that it is valid. This analysis is fed back to the change requestor who may respond with a more specific requirements change proposal, or decide to withdraw the request.
- Change analysis and costing
  - The effect of the proposed change is assessed using traceability information and general knowledge of the system requirements. Once this analysis is completed, a decision is made whether or not to proceed with the requirements change.
- Change implementation
  - The requirements document and, where necessary, the system design and implementation, are modified. Ideally, the document should be organized so that changes can be easily implemented.

