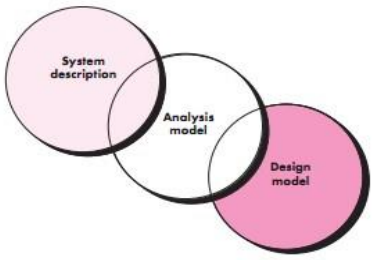
Q: Describe Requirement Analysis (Diagram)?
Ans:



Requirements analysis results in the specification of software's operational characteristics that *indicate software's interface with other system elements and establish constraints that software must meet.*

Requirements analysis allows you to elaborate on basic requirements established during the inception, elicitation, and negotiation tasks that are part of requirements engineering.

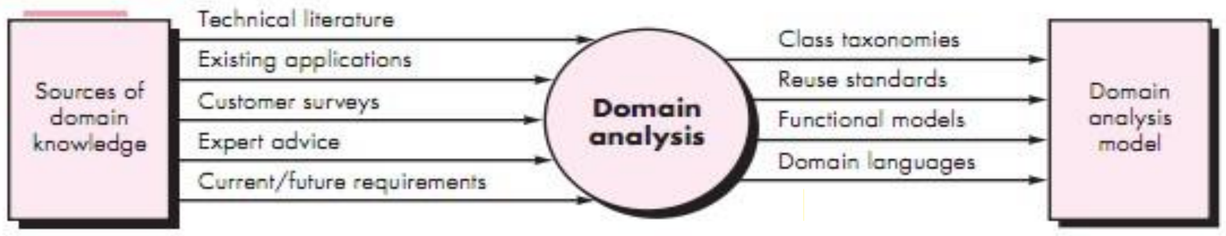
Q: What do you know about Analysis Rules of Thumb?
Ans:
***Arlow and Neustadt** suggest several rules of thumb that should be followed when creating the analysis model:*

1. The model should focus on requirements that are visible within the problem or business domain. The level of abstraction should be relatively high.
2. Each element of the requirements model should add to an overall understanding of software requirements and provide insight into the information domain, function, and behavior of the system.
3. A database may be required, but the classes necessary to implement it, the functions required to access it, and the behavior that will be exhibited as it is used should be considered only after problem domain analysis has been completed.
4. Minimize coupling throughout the system. It is important to represent relationships between classes and functions.
5. Be certain that the requirements model provides value to all stakeholders. Each constituency has its use for the model.
6. Keep the model as simple as it can be. Don't create additional diagrams when they add no new information. Don't use complex notational forms, when a simple list will do.

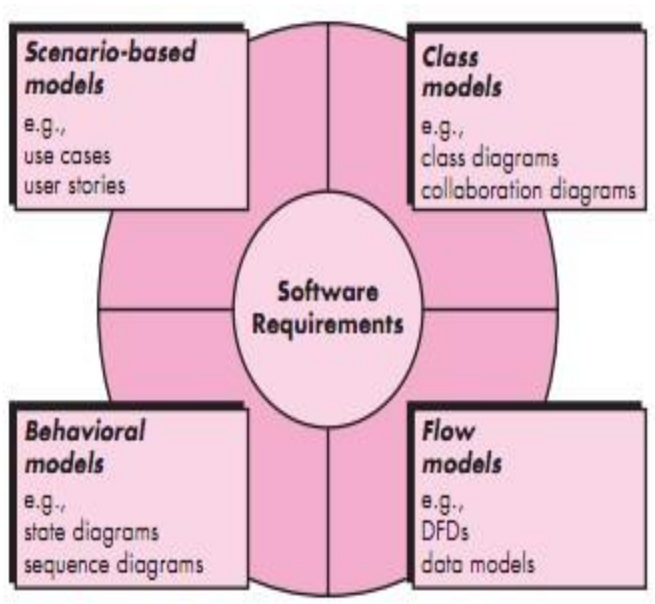
In software engineering , domain analysis is the process of analyzing related software systems in a domain to find their common and variable parts.It is a model of wider business context.

Q: Define Domain Analysis (Diagram)?
Ans:
Software domain analysis is the identification, analysis, and specification of common requirements from a specific application domain, typically for reuse on multiple projects within that application domain. [Object-oriented domain analysis is] the identification, analysis, and specification of common, reusable capabilities within a specific application domain, in terms of common objects, classes, subassemblies, and frameworks.

FIGURE 6.2 Input and output for domain analysis



Q: Explain Requirements Modeling approaches (Diagram)?
Ans:
*One view of requirements modeling, **called structured analysis**, considers data and the processes that transform the data as separate entities. Data objects are modeled in a way that defines their attributes and relationships. Processes that manipulate data objects are modeled in a manner that shows how they transform data as data objects flow through the system.*
A second approach to analysis modeling, called object-oriented analysis, focuses on the definition of classes and how they collaborate to affect customer requirements.



Each element of the requirements model presents the problem from a different point of view.
Scenario-based elements depict how the user interacts with the system and the specific sequence of activities that occur as the software is used.
Class-based elements model the objects that the system will manipulate, the operations that will be applied to the objects to affect the manipulation, relationships between the objects, and the collaborations that occur between the classes that are defined.
Behavioral elements depict how external events change the state of the system or the classes that reside within it.
Finally, flow-oriented elements represent the system as an information transform, depicting how data objects are transformed as they flow through various system functions.

Q: What do you know about Scenario-Based Modeling?
Ans:
Scenario-based modeling is one of the **sub-stages of requirements modeling**. It's also typically the first stage of requirements modeling since it identifies the primary use cases for the proposed software system or application.
Requirement modeling with UML begins with the creation of scenarios in the form of use cases, activity diagrams, and swim lane diagrams.
It includes
1. Creating a preliminary case
2. Refining a preliminary case
3. Writing a formal use case

Q: Differentiate between Activity Diagram and Swim-Lane Diagram?
Ans:
Activity Diagram: The activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario.
It is similar to flow chart diagrams
Swim-Lane Diagram: The swim-lane diagram is a useful variation of the activity diagram and allows you to represent the flow of activities described by the use case.
Responsibilities are represented as parallel segments that divide the diagram vertically.
Q: Define ER diagram with example?
Ans:
The object-relationship pair is the cornerstone of the data model. These pairs can be represented graphically using the entity-relationship diagram (ERD). The primary purpose of the ERD is to represent data objects and their relationships.