

Analysis Modeling

Requirements analysis

- ☐ Specifies software's **operational characteristics**
- ☐ Indicates software's **interface with other system elements**
- ☐ Establishes **constraints** that software must meet

Requirements analysis allows the software engineer (called an *analyst* or *modeler* in this role) to:

- ☐ Elaborate on basic requirements established during earlier requirement engineering tasks
- ☐ Build models that depict user scenarios, functional activities, problem classes and their relationships, system and class behavior, and the flow of data as it is transformed.

Analysis Modeling Principles

1. **Information domain** encompasses that the data flow into the system, out of the system and data stored.

2. **Function** provide direct benefit to end-users and also provide internal support for those features that are user visible.

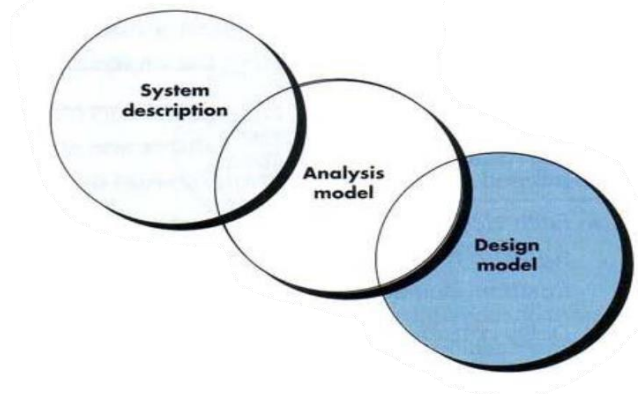
3. **Behavior** driven by its interaction with the external environment.

E.g. Input provided by end-users, control data provided by an external system, or monitoring data.

4. Key strategy of analysis model, divide complex problem into sub-problem until each sub-problem is relatively easy to understood. This concept is called **partitioning**.

5. The “**essence**” of the problem is described without any consideration of how a solution will be implemented.

Analysis Model - A Bridge



An **analysis rule of thumb** is that the model “should focus on requirements that are visible within the problem or business domain.” What types of requirements are not visible in these domains? Provide a few examples. The model should focus on requirements which are visible with in problem or business domain. (ВИДЛИВИ СЕ СИТЕ БАРАЊА КОИ СЕ НАОЃААТ НЕПОСРЕДНО ОД ПРОБЛЕМОТ ИЛИ БИЗНИС ДОМЕНОТ.)

Don't get bogged into the details.

Elements of Analysis model

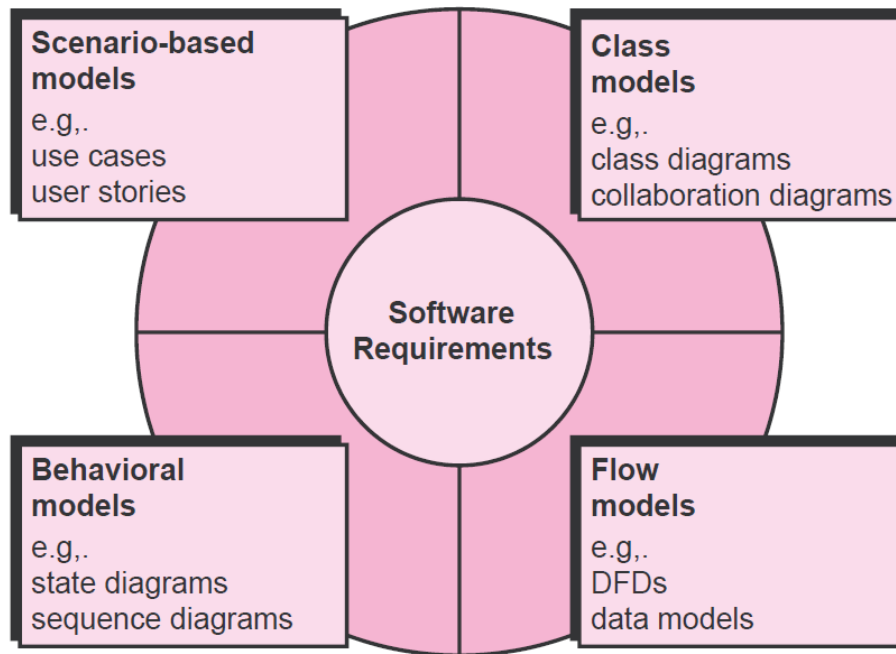
☐ There are two approaches

1. Structured Analysis(СТРУКТУРНА АНАЛИЗА):-

- ☐ Data objects are modeled in a way that defines their attributes and relationships.
- ☐ Processes that manipulate data objects in a manner that shows how they transform data as **data objects flow** through the systems.

2. Object Oriented Analysis(ОБЈЕКТНО ОРИЕНТИРАНА АНАЛИЗА):-

- ☐ Focuses on the definition of **classes** and the manner in which they collaborate with one another.
- ☐ UML is predominantly object oriented.



1) Scenario-based elements

☐ **Use-case**—How external actors interact with the system (use-case diagrams; detailed templates)

☐ **Functional**—How software functions are processed in the system (flow charts; activity diagrams)

☐ **Activity**—can be represented at many diff. level of abstraction.

Formal Use-Case

- Use case:
- Iteration:
- Primary actor:
- Goal in context:
- Preconditions:
- Trigger:
- Scenario:
- Exceptions:
- Priority:
- When available:
- Frequency of use:
- Channel to actor:
- Secondary actors:
 - ☐ Channels to secondary actors:
- Open issues:

Имаме два вида на дијаграми:

1. Дијаграм на Активност
2. Swimlane дијаграм

2)Class-based elements

☐The various system objects (obtained from scenarios) including their attributes and functions (**class diagram**).

3)Behavioral elements(МИЛИЕВ И МУРОВ ТИП)

☐How the system behaves in response to different events (**state diagram**).

4)Flow-oriented elements

☐How information is transformed as if flows through the system (**data flow diagram**)

☐System accepts input in a variety forms; **applies functions** to transform it; and produces output in variety forms.

=====МОДЕЛИ НА АНАЛИЗА=====

1)Data modelling

Analysis model often begin with data modeling.

Data model consists of three interrelated pieces of information:

- ☐The **data object**,
- ☐The **attributes** that describe the data object, and
- ☐The **relationships** that connect data objects to one another.

Cardinality and Modality

How many occurrences of object X are related to how many occurrences of object Y called **cardinality**.

Cardinality is usually expressed as simply 'one' or 'many'.

☐**1:1** –One object can relate to only one other object.

☐**1:M** –one object can relate to many objects.

☐**M:N** –Some no. of occurrences of an object can relate to some other no. of occurrences of another object.

Modality of a relationship is **0** if there is no explicit need for the relationship to occur or the relationship is optional.

☐The modality is **1** if an occurrences of the relationship is mandatory.

2)Creating a Data Flow Model

☐It enables software engineer to develop models of the information domain and functional domain at the same time.

☐**Data flow diagram** may be used to represent a system or software at any level of abstraction

☐As DFD is refined into greater levels of detail, the analyst performs an implicit functional decomposition of the system.

☐As DFD refinement results in corresponding refinement of data as it moves through the processes that represent the application

CSPEC(CONTROL SPECIFICATIONS) represent the **behavior** of the system in two different ways.

PSPEC(PROCESS SPECIFICATIONS) is used to describe all flow model processes that appear at the final level of refinement.

☐It include narrative text, a program design language (PDL) description of the process algorithm, mathematical equations, tables, diagrams or charts.

3)Class based modelling

☐**Identifying Analysis Classes:** КАКО СЕ МАНИФЕСТИРААТ САМИТЕ КЛАСИ НА АНАЛИЗА.

☐**Specifying Attributes:** МНОЖЕСТВО ОД ПОДАТОЧНИ ОБЈЕКТИ КОИ ЦЕЛОСНО ЈА ДЕФИНИРААТ КЛАСАТА ВО КОНТЕКСТ НА ПРОБЛЕМОТ.

☐**Defining operations:** ОДНЕСУВАЊЕТО НА ОБЈЕКТОТ

☐**CRC modelling(Class-Responsibility-collaborator):** КАРТИ ОД КЛАСИ, ВО ВИД НА ОБЈЕКТИ КАКО ВО ООП.

Class: FloorPlan	
Description:	
Responsibility:	Collaborator:
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

4)Behavioral Modeling

Behavioral model indicates how software will respond to external events.

Passive state is simply the current status of all of an object's attributes.

Ex. Player –class

current position and orientation –attributes.

Active State is current state of the object as it undergoes a continuing transformation or processing.

Ex. Player –class

active state –moving, injured, trapped, lost etc.

UML state diagram that represents active states for each class and events that causes changes between these active state.

Sequence diagram, it indicates how events cause transitions from object to object.

5)Mechanics of Structured Analysis

It all about

☐ **Entity relationship diagram (ERD)**- ИНТЕРАКЦИЈА СО КЛИЕНТОТ. ТОЈ СЕ ПРАШУВА ПОВЕЌЕ ПРАШАЊА ШТО СЕ ОДНЕСУВААТ НА БИЗНИС И АПЛИКАЦИСКИОТ МОДЕЛ НА СИСТЕМОТ.

☐ **Data flow diagram (DFD)**

☐ **State transition diagram (STD)**

Data dictionary

Data dictionary is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definitions so that both user and system analyst will have a common understanding of inputs, outputs, components of stores and [even] intermediate calculations.

Содржи најчесто информации за: **ИМЕ, АЛИАС, КАДЕ/КАКО СЕ КОРИСТИ**

Content description—a notation for representing content.

Supplementary information—other information about data types, preset values (if known), restrictions or limitations, and so forth.

The notation used to develop a content description is noted in the following table:

Data	Construct Notation	Meaning
	=	is composed of
Sequence	+	and
Selection	[]	either-or
Repetition	{ } ⁿ	<i>n</i> repetitions of
	()	optional data
	* ... *	delimits comments