

Week 02

SCD

Lecture 03

There are two parts of software construction:

- i. logic Building → software Engr.
- ii. implementation → coding

Understanding the problem domain is very important.

Requirement gathering is very important.

Functional Requirement:

- i - user requirement
- ii - system requirement

User requirement tell us that what is required not that how to solve it - it is also mentioned in problem domain.

It focuses on "what" not "How".
System requirements tell that how these requirements are implemented.

FUR Post

FSS Login | Block

Non-Functional Req:

- 1- product Req
- 2- organizational
- 3- Req External

$S \rightarrow R$

$PRM \rightarrow S$

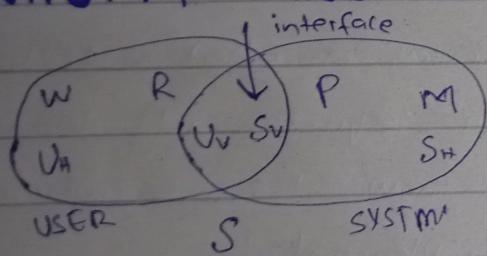
1- Domain

2- Requirement

3- architecture

4- Design

WRSPM Model:



W = world (domain)

R = requirement

S = specification

P = program

M = Hardware machine

Architecture :

- * Decompose an enterprise system into independent sub-system that have value in the system
- * How these sub-systems interact?

* principles and guidelines for the design evolution over time.

Components:

Components exist on their own -

component is made up of module and module is dependent on classes.

Statement :-

Persons or companies may own car. The car owner ID is the ID either the person or the company that owns a car.

A car may have only one owner (the person or the company).

A car may have loan or multiple loans. A bank provides a loan to a person or a company for the purchase of a car. only the loan obtainer may obtain a loan on the car. The car owner type and customer loan type indicate whether the car owner / loan holder is a person or company.

List of nouns:

- | | |
|-------------|------------|
| 1 - company | 2 - Car |
| 3 - Bank | 4 - Person |
| 5 - ownerID | 6 - ID |
| | 7 - loan |

List of verbs:

- 1- own
- 2- purchase
- 3- provide
- 4- indicate
- 5- obtain

Classes:

- 1- Person
 - 2- Company
 - 3- Car
 - 4- Loan
 - 5- Bank
- 3- Car
 |
 | Type ID
 | owner ID
 | (P, C) 1
 | attributes

"there are total 14-15
design goals."

- [1- Why the client need that software ?
- 2- What problems he/she wants to resolve ?

Attributes:

1- Person :

name, CNIC, address, Ph#

2- Company :

name, address

After viewing the description of any problem statement the first step is to find the classes.

ID's tell us the relationship b/w different attributes.

Quiz 01 : SDLC 27 Sep 2023

Week 03 SCD lecture 05

- = I was absent in this lecture the previous lecture was explained and man told about multiplicity.
- i - one-to-one
 - ii - one-to-many
 - iii - many-to-many

Week 03 SCD lecture 06

Step 01: Domain modelling:

while designing a software must understand your domain. understanding your domain is called domain modelling.

for example:-

- 1- Medical Domain
- 2- Police Station Domain
- 3- Education Domain

is used to check either the work is good that we are doing.

WRSPM → it is a reference model.

Step 02 : Requirement elicitation:

your whole software is
useless.

User requirements are what users
want -

for example Buzzle should produce
a sound when heart beat of a
person stops.

it is user requirement.

Buzzle should present at that
place where nurses are
present all time. this
is domain.

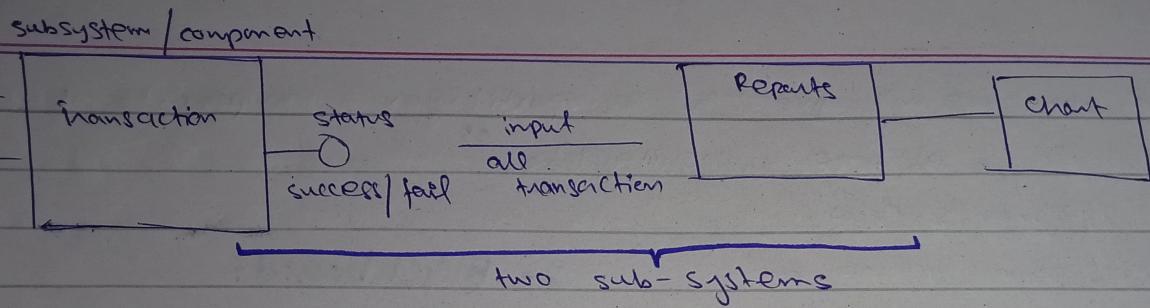
* if your requirement
elicitation phase is
not correct than

specifications:

- * it is interface
that is collapsing
the both pants.
- *

Software architecture :

- * what will be your
components and how
they will interact.
- * it is a complete
module.
- * components are independent



- These all sub-systems are acting on their own.

Ques 1- What is the difference b/w software architecture and software design?

“ multiple modules can exist in a sub-system.”

“ At upper level all your decisions about software comes in software architecture.”

Software Architecture Models:

- ① Pipe & filter
- ② Blackboard
- ③ Layered
- ④ Client server
- ⑤ Event Based.

How much time we will give to a particular components if decided in Software architecture."

independ components that

have their own value and

how they interact called software architecture."

1- Pipe and filters:

input will be same,

output will be same,

and work same in

every environment called
pipe and filter.

Example:

- Compiler.

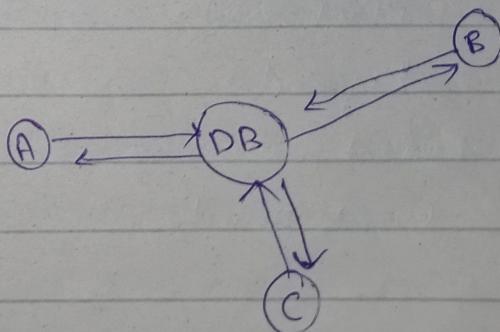
- Unix / Linux / MacOS

2- Blackboard:

- Central DataBase.

- Data is present in high integrity.

- multiple modules and components can exist.



3- Layered :

Components are divided into layers.

↳ How communication will

be in model and view?

→ model → model

→ view → view

→ controller → model

5- Event Based:

Something happens when an event occurs. • Similar to Blackboard.

• These five models are not separately present in any software.

• These models are followed all over the world.

4- Client Server:

When you want to check merit list

then in response you will get merit list.

Software Concerns:

1- System structure (decompose into components).

2- interaction of components.

3- Modular decomposition.

Modular Decomposition: → How will you divide your components into classes.
it is the part of architecture and is also named as software design.

parts of a Component:

- 1- Classes
- 2- How classes interact
- 3- What functionality classes will perform.

Properties of modules:

there are 4 properties of modules :-

Modularity:

Breaking down of a component and reassembling / interaction -

Modules:

- 1- coupling (low)
- 2- cohesion (highly)
- 3- information hiding
- 4- Data encapsulation

“ collection of modules make a component or Sub-system.”