

"Good judgement is usually the result of experience. And experience is frequently the result of bad experience judgment. But to learn from the experience of others requires those who have experience to share the knowledge with those who follow."

-Barry LePatner

Software Architecture:-

The software architecture of a system is a set of structures, needed to reason about the system which comprise software elements, relations among them & properties of both.

structures → set of software elements

Rules:-

→ if software architecture is not being documented, it can not be validated.

→ Background engineering can not be done

Architecture is not made through the code of the software.

Types of Structures:-

- ① Module Structure
- ② Annotations Structure
- ③ CNC (Components & Connectors) Structures

A structure is simply a set of elements held together by a relation.

Software systems are composed of many structures & no single structure holds claim to being lead architecture.

3 categories of architectural structure which will play an important role in the design documentation & analysis of architecture.

cloudcomputingresearch.com

What makes a good Architecture?

* Process Related.

- The architecture should be the product of a single architect or a small group of architects which an identified technical leader.

* There should be a strong connection b/w architects & developing team.

- The architect should on ongoing basis based the architecture on a prioritized list of well-specified quality requirements. These will inform the trade-offs that always occur

→ Prioritizing requirements also prioritizes the architecture

(Quality Attributes)

(spiral model of requirement gathering)

- The architecture should be documented using views.

→ views refer to the views of stakeholders, not the viewpoint

The views should address the concerns of the most important stakeholders in support of the project timeline.

This might mean minimal documentation at first, elaborate it later.

→ educating the stakeholders may also be a view but it will affect the architecture of the system.

- The architecture should be validated for its ability to deliver a system's important quality attributes. This should occur early in the life cycle when it

returns the most benefit and repeated as appropriate to ensure that changes to the architecture have not rendered the design of the software.

→ early-on-evaluating the architecture of the system means evaluating the system early (before it is even made to check whether it satisfy the benefits it claims to be. else

it will be a technical dead

→ most of the methods are adopted from stockmarkets (economics) & then merged with software & after that evaluating of architecture is done.

• there is still a possibility of technical dead may occur.

→ Personal Lw Process → where you keep a record of everything you have done (# of lines of code, code, # of working hours etc)

(We can minimize it but not completely reduce it)

Team Lw Process → log of team it kept

Collaborative Lw Process
→

• The architecture should lend itself to incremental implementation to avoid having to integrate everything at once as well as to discover problems early.

* Product Related -

- Architecture should feature well defined modules whose functional responsibilities are assign on the principle of information hiding. & separation of concerns.

→ information sharing should be on need-to-know basis.

The information hiding modules should encapsulate things likely to change does effecting the software to change.

- The architecture should never depend on a particular version of a commercial product or tool, if it must, it should be structured so that the changings to a different version is straight forward & inexpensive.

Quality Attributes → Availability, Performance.

for e.g. don't rely on Oracle, search for a better one.

- Modules that produce data should be separate from the modules that consume data. This tends to increase ^{maintainability} modifying because changes are frequently confined on either the production and consumption site.
→ e.g. structures in C++.

- Every process should be written so that its assignment to a specific processor can be easily changed perhaps even at run time.

- Don't expect one-to-one correspondance between modules and components.

- The architecture should contain a ^{small set of} specific set of contention areas.

The resolution of which is clearly specified & maintained.

- The architect should identify possible contention areas in the architecture of software.

- The architecture should feature a small number of ways for components to interact. i.e. the system should do the same thing, the same way throughout.

Types of Structures:-

1- Module Structures:-

Some structures partition system into implementation units called modules.

Modules are assigned specific computational responsibilities & are basis of work assignments called for programming teams.

OOAD → Object Oriented Analysis & Design

Task is decomposed into layers & architecture is made.

i.e. what activities are carried out by which stake holder.

These decompose structures are known as mo

Pipe & filter architecture \rightarrow use CNN structure.

2- Components & Connectors:- Structures

Components - & - Connector structures embody decisions as to how the system is to be structured as a set of elements that have run time behaviour as a component & interactions

3- Annotation Structures:-

Allocation structures embody decision as to how the system will relate to non-software structures in its environment.

These structures show the relationship b/w software elements & elements in one or more external environment in which software is created & executed. agents, sw etc how are they going to interact with hardware &

They are the algo which determine how the is going to interact with processor. (hardware)

At the same time, it also contains C & N structures as we are working with.

It also contains module structures, bcz at the beginning we determine which component carries which task.

System Architecture:-

A systems architecture is a representation of a system in which there is a mapping of functionality onto hardware or software components.

A mapping of the software architecture onto the hardware architecture & a concern with the human interaction with these components i.e. system architecture is concerned with the total system including hardware, software & humans. (people, process & products)

MIS →

Enterprise Architecture:-

Enterprise architecture is a description of the structure & behaviour of an organization's processes, personnel & organizational sub-units along with organizational core goals.

An enterprise architecture need not include information system.

Types of Module Structures:-

→ Decompose Structures:-

Decompose structure is a sub module of module structure & is useful for resource allocation & data hiding.

It helps in modifiability.

* Class diagrams is an example of module structures.

→ Data Structures:-

→ helps in consistency.

→ Class Structures:-

→ Layered Structures

Types of Allocation Structures:-

(GANTT chart)

It includes deployment structures

Something has been allocated.
Team A has been allocated to do a certain task

Adopting or migrating towards a service (i.e. cloud etc)

→ Implementation Structures:-

It can include / useful for integration, testing, implementation & work assignments.

→ Work Assignment Structures:-

→ scheduling of tasks.

CHAPTER No 2

Science
for all

"Software Architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled."

- Eoin Woods.

Architecture serves as a mean to communicate stakeholders.

Why is Architecture Important? (Technical Perspective)

- A documented architecture enhances communication among stakeholders.
- Architecture based development focuses attention on the assembly of components rather than simply on their creation.
- By restricting design alternatives, architecture channels the creativity of developers, reducing design & system complexity.
- An architecture can be the training foundation for training a new member.

Reasoning about & managing change

local change ⇒ changing single element in architecture.

non-local change ⇒ modifying / adding different elements (may belong to different structure)

architectural change ⇒ changing architecture itself.

Software Architecture in Process:-

Relation of SW Architecture to other Development Tasks

When we use the term software Architecture, we are speaking of the purposeful design/plan of a system.

This designed plan isn't a project plan that describes activities & staffing for designing the architecture or developing the product, instead it is a structural plan that describes the element of the system, how they fit together & how they work together to fulfil the system requirements.

It is used as a blue print during the development process & it is also used to negotiate

system requirements & to set expectations with customers & marketing & management personal.

The project manager uses the design plan as input to the project plan.

Design plans are used to know what makes our architecture & software different from the existing one.

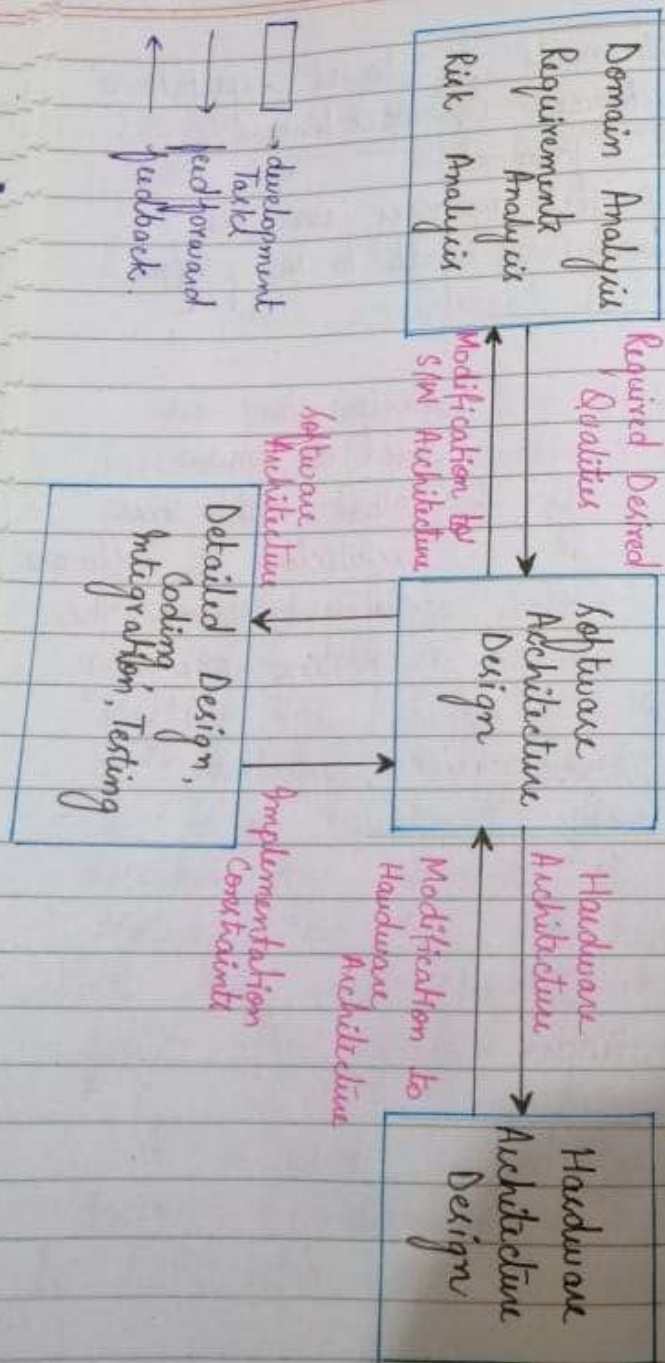
Architectural Influence

• Technical Perspective / Context

The architecture can affect the stakeholder requirements for the next system by giving the customer the opportunity to receive a system in a more reliable, timely & economical manner i.e. the subsequent system were to be built from scratch & typically with fewer defects.

• Project Context

The architecture affects the structure of the developing organization. An architecture prescribes a structure for a system which



particularly & prescribes the units of s/w that must be implemented & integrated to form the system.

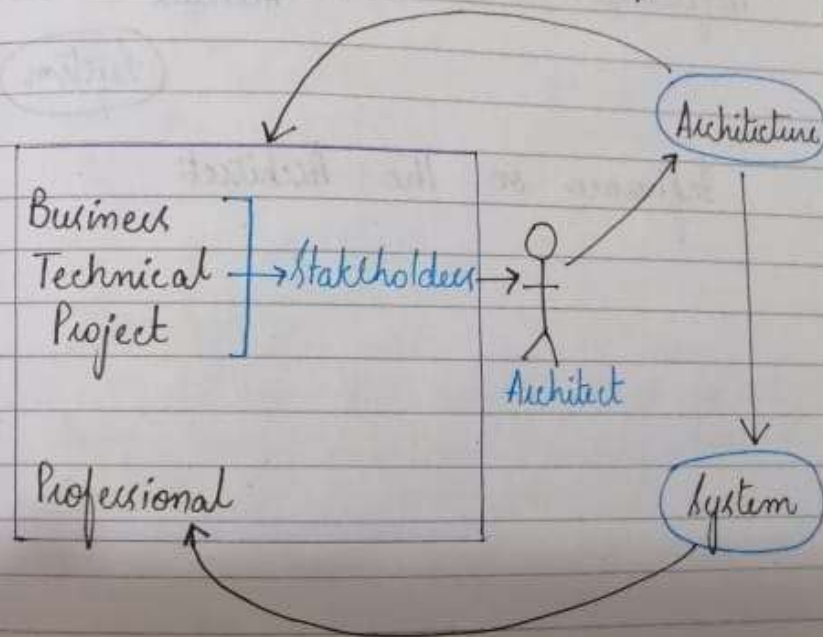
• Business Context

The architecture can affect the business goals of the developing organization. A successful system built from an architecture can enable a company to establish a foothold in a particular market segment.

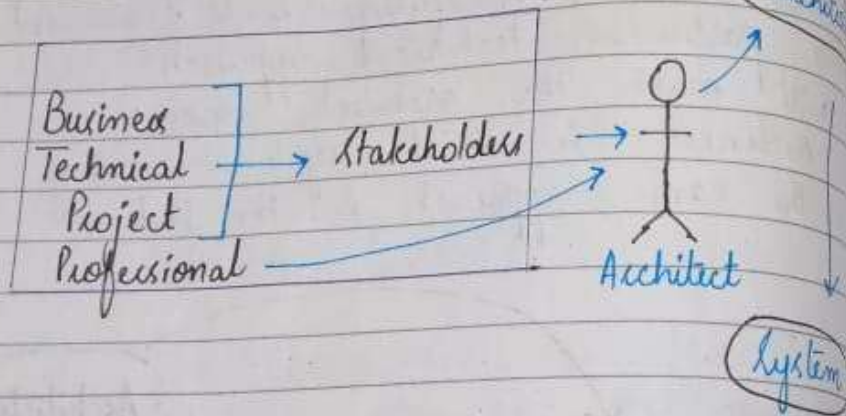
• Professional Context

The process of system building will affect the architect's experience with subsequent systems by adding to a

corporate experience base. A system that was successfully built around a particular technical approach will make the architect more inclined to build systems using the same approach in the future.



Architectural Influence Cycle



Influence on the Architect

CHAPTER No: 4

Quality Attribute Requirements

These requirements are qualifications of the functional requirements or of the overall products.

Constraints

When the decision has already been made & you do not have the freedom to choose.

Problems with Quality Attributes

There are 3 problems with System Quality Attributes.

① → The definitions provided for an attribute are not testable.

② → Discussion often focuses on which quality a particular concern belongs to, is a system failure due to DoS attack & aspect of availability, as aspect of performance, an aspect of security or an aspect of usability.
(Aspect-Oriented RE)

③ → Each attribute community has developed its own vocabulary.
The performance community has events arriving at a system

The security community has attacks, the availability community has failures & usability community has inputs at a system. All of these may actually refer to the same occurrence but they are described using different terms.