

SOFTWARE ENGINEERING

SOFTWARE ARCHITECTURE

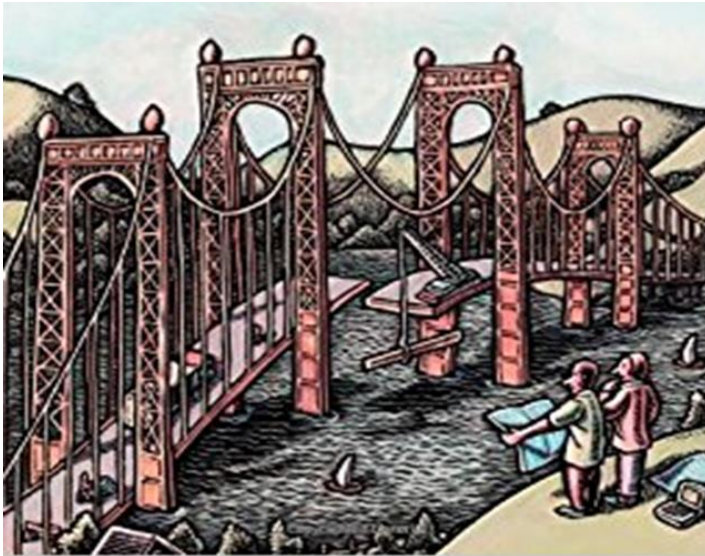
Prof. Phalachandra H. L

Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet and supplemented by my experience. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

SOFTWARE ARCHITECTURE

Architectural Views



Prof. Phalachandra H. L

Department of Computer Science and Engineering

Architectural View

Views represent ways of describing the Software Architecture, enabling the system to be viewed by different stake holders in perspectives of their interest.

E.g. UI View, Process View

Architectural Styles

Demonstrates how the subsystems or elements are organized or structured
It's a way of organizing code

E.g. Pipe & filters, Client-Server, Peer-to-Peer etc.

Architectural Pattern

Is a known or a proven solution to the architectural problem of structuring and functioning of the subsystems which has been used earlier and is known to work for the problem scenario

E.g. MVC separating UI from the rest

Architectural Views

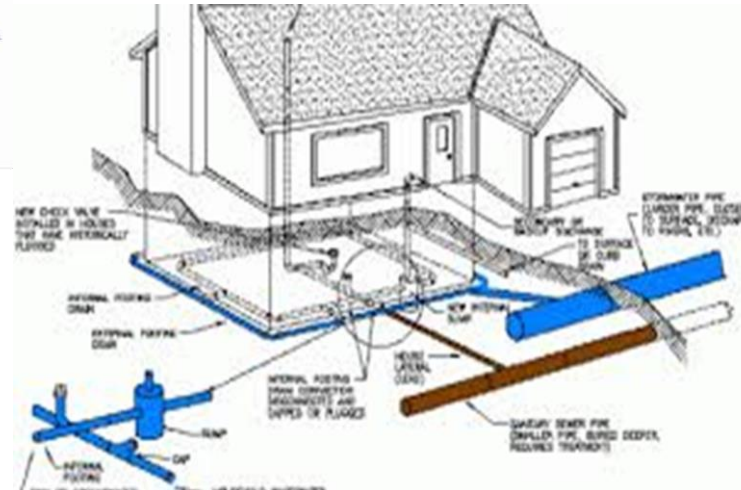
Example Architectural views of a House



Elevation



Plan



Plumbing View

There are 4 ways architectural elements can be viewed

1. As a Structure of Modules (or Module View point)

- Modules are units of (code) Implementation with some functional responsibility
- Structure the system as a set of code units (modules)
- An Architect enumerates what the units of software will have to do and assigns each item to a module
- The larger modules may be decomposed to sub-modules of the acceptable fine level by ensuring that likely changes fall within the purview of at most a few small modules.
- It is often used as the basis for the development project's organization and deliverables like documentation.
- There is less emphasis on how the resulting software manifests itself at runtime

2. As a component-and connector structure (or View point)

- Dynamic View of considering the system in execution or runtime

E.g. Processes view which would include a set of processes connected by synchronization links

Component or Processing Element is a software which converts inputs to outputs. It could do a computation or act as a server with state and operations or something which governs a sequence of events. E.g. filter or a controller

Data Element information needed for processing or information to be processed e.g. memory with data like a DB

Connecting Element is the glue between the components and the Data element like procedure calls, RPCs.

3. As an allocation structure (or View point) includes the

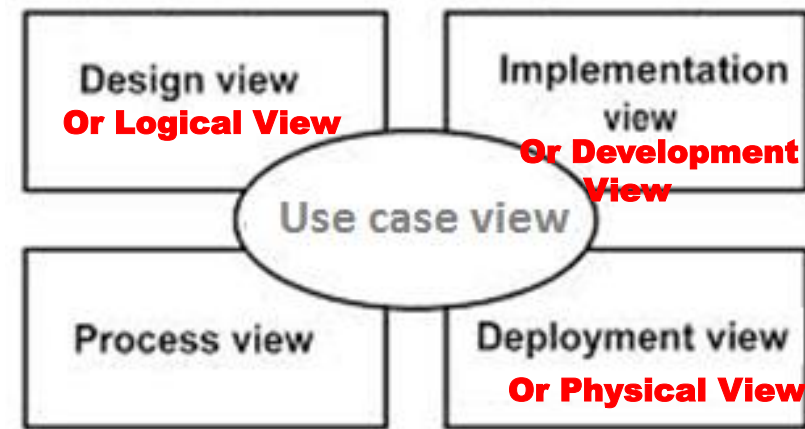
Deployment structure which shows how software is assigned to the hardware elements and which communication paths are used. Relations are "allocated-to", showing on which physical units the software elements reside, and "migrates-to" if the allocation is dynamic. This view allows an engineer to reason about performance, data integrity, availability, and security. It is of particular interest in distributed or parallel systems

Implementation structure which indicates how software is mapped onto file structures in the system's development, integration, or configuration control environments

Work Assignment structure which shows who is doing what and helps to determine which knowledge is needed where. E.g. Functional commonality to a single team.

4. Krutchens (4+1 View) or model

- **Use case view** (exposing the requirements of the system or the scenarios)
- **Design view** (exposes vocabulary of the problem space and the solution space) Class Diagrams, Sequence diagrams etc.
- **Process view** encompasses the dynamic aspects or the runtime behavior of the system. Threads and processes that form the systems. Addresses performance, Concurrency etc.
- **Implementation view** (addresses the realization of the system. UML diagrams like package diagrams are used)
- **Deployment view** (focuses on system engineering issues)





THANK YOU

Prof. Phalachandra H.L.

Department of Computer Science and Engineering

phalachandra@pes.edu