UNIT-IV
PART - A

Model-Based Software Architectures: A Management Perspective and A Technical Perspective.

Introduction

- Software architecture is the central design problem of a complex software system, in the same way that architecture is the central design problem of a complex skyscraper (a very tall building).
- However, software architecture has several additional dimensions of complexity.

Architecture: A Management Perspective

- The most critical technical product of a software project is its architecture: the infrastructure, control, and data interfaces that permit software components to cooperate as a system and software designers to cooperate efficiently as a team.
- Establishing accurate and precise communications among teams of people is a timeless problem in any organization.

Architecture: A Management Perspective

- When the communications media include multiple languages and intergroup literacy varies, the communications problem can become extremely complex and even unsolvable.
- If a software development team is to be successful, the inter project communications, as captured in the software architecture, must be both accurate and precise.

Architecture: A Management Perspective

- From a management perspective, there are three different aspects of architecture.

1. An **architecture** (the intangible design concept) is the design of a software system that includes all engineering necessary to specify a complete bill of materials.
2. An **architecture baseline** (the tangible artifacts) is a slice of information across the engineering artifact sets sufficient to satisfy all stakeholders that the vision (function and quality) can be achieved within the parameters of the business case (cost, profit, time, technology, and people).
3. An **architecture description** (a human-readable representation of an architecture, which is one of the components of an architecture baseline) is an organized subset of information extracted from the design set model(s). It includes the additional ad hoc notation (text and graphics) necessary to clarify the information in the models. The architecture description communicates how the intangible concept is realized in the tangible artifacts.

Architecture: A Management Perspective

- These definitions are necessarily abstract, because architecture takes on different forms across different system domains.
- In particular, the number of views and the level of detail in each view can vary widely.
- A model is a relatively independent abstraction of a system.
- A view is a subset of a model that abstracts a specific, relevant perspective.

Architecture: A Management Perspective

- The importance of software architecture and its close linkage with modern software development processes can be summarized as follows:
  - Achieving a stable software architecture represents a significant project milestone at which the critical make/buy decisions should have been resolved.
  - Architecture representations provide a basis for balancing the trade-offs between the problem space (requirements and constraints) and the solution space (the operational product).
  - The architecture and process encapsulate many of the important (high-payoff or high-risk) communications among individuals, teams, organizations, and stakeholders.

Architecture: A Management Perspective

- Poor architectures and immature processes are often given as reasons for project failures.
- A mature process, an understanding of the primary requirements, and a demonstrable architecture are important prerequisites for predictable planning.
- Architecture development and process definition are the intellectual steps that map the problem to a solution without violating the constraints; they require human innovation and cannot be automated.

Architecture: A Technical Perspective

- Software architecture encompasses the structure of software systems, their behaviour, and the patterns that guide these elements, their collaborations, and their composition.
- The context of software architecture structure, behaviour, and patterns must include functionality, performance, resilience, comprehensibility, economic trade-offs, technology constraints, and aesthetic concerns.

Architecture: A Technical Perspective

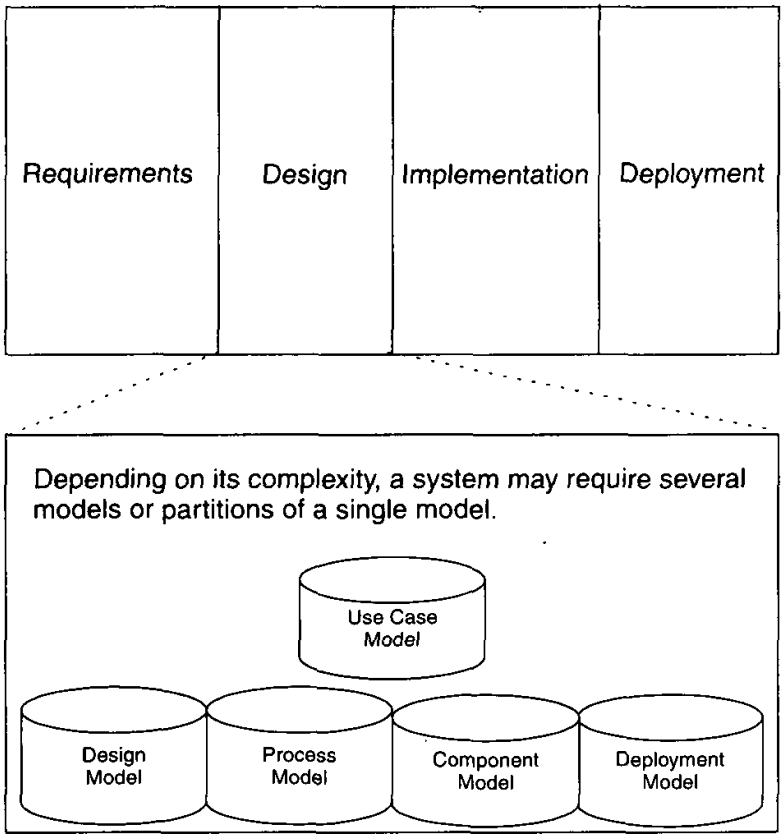
- An architecture framework is defined in terms of views that are abstractions of the UML models in the design set.
- The design model includes the full breadth and depth of information.
- An architecture view is an abstraction of the design model; it contains only the architecturally significant information.
- Most real-world systems require four views: design, process, component, and deployment.

Architecture: A Technical Perspective

- The purposes of these views are as follows:
- Design: describes architecturally significant structures and functions of the design model
- Process: describes concurrency and control thread relationships among the design, component, and deployment views
- Component: describes the structure of the implementation set
- Deployment: describes the structure of the deployment set

Architecture: A Technical Perspective

- Below figure summarizes the artifacts of the design set, including the architecture views and architecture description.
- The architecture description is usually captured electronically but is always maintained so that it is printable as a single cohesive document.
- The engineering models and architectural views are defined as collections of UML diagrams.



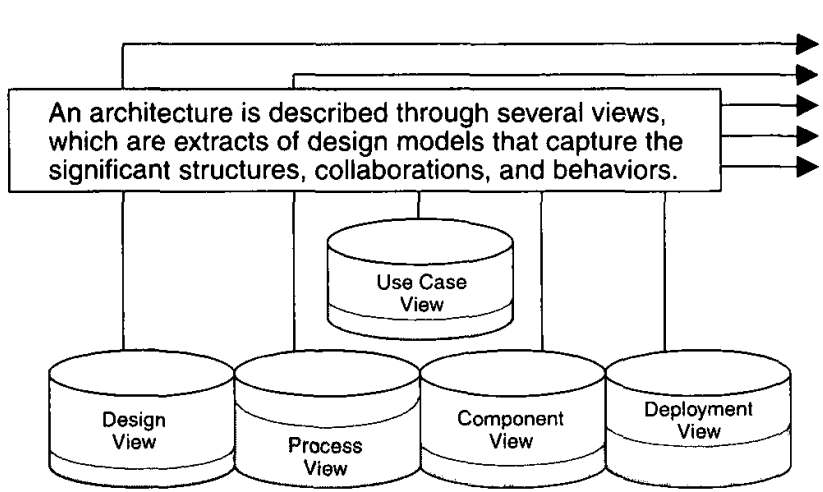
The requirements set may include UML models describing the problem space.

The design set includes all UML design models describing the solution space.

The *design, process,* and *use case models* provide for visualization of the logical and behavioral aspects of the design.

The *component model* provides for visualization of the implementation set.

The *deployment model* provides for visualization of the deployment set.



Architecture Description Document

Design view
Process view
Use case view
Component view
Deployment view
Other views (optional)
Other material:
- Rationale
- Constraints

Architecture: A Technical Perspective

- The requirements model addresses the behavior of the system as seen by its end users, analysts, and testers.
- This view is modeled statically using use case and class diagrams, and dynamically using sequence, collaboration, state chart, and activity diagrams.

Architecture: A Technical Perspective

- The **use case view** describes how the system's critical use cases are realized by elements of the design model.
- The **design view** describes the architecturally significant elements of the design model.
- The **process view** addresses the run-time collaboration issues involved in executing the architecture on a distributed deployment model, including the logical software network topology, interprocess communication, and state management.
- The **component view** describes the architecturally significant elements of the implementation set.
- The **deployment view** addresses the executable realization of the system, including the allocation of logical processes in the distribution view to physical resources of the deployment network.

Architecture: A Technical Perspective

- Generally, an architecture baseline should include the following:

- **Requirements:** critical use cases, system-level quality objectives, and priority relationships among features and qualities
- **Design:** names, attributes, structures, behaviors, groupings, and relationships of significant classes and components
- **Implementation:** source component inventory and bill of materials (number, name, purpose, cost) of all primitive components
- **Deployment:** executable components sufficient to demonstrate the critical use cases and the risk associated with achieving the system qualities