Software Architecture & Design

This presentation will explore key concepts, factors, advantages, and stakeholders involved in software architecture to build a strong foundational knowledge.

 by FAZLA RABBY RAIHAN

Lesson Objectives

What Is Software Architecture?

Learn the definition and scope of software architecture at the highest design level.

Key Considerations

Identify critical factors that influence architectural decisions in software projects.

Importance and Benefits

Understand why robust architecture is vital and the benefits it delivers.

Stakeholder Recognition

Discover who cares about architecture and their roles in software development.

Lesson Outcomes

1 **Comprehensive Architecture Overview**

Gain a broad understanding of the role and essence of software architecture.

2 **Factors for Architects**

Know what considerations guide an architect during design choices.

3 **Importance of Architecture**

Recognize the impact architecture has on effective software development.

4 **Stakeholder Insight**

Identify key stakeholders and how architecture affects them.

What is Software Architecture?

Definition and Structure

- Software architecture is all about how a software system is constructed at the highest level. It is the **fundamental design** of the entire software system.
- What elements are included in the system
- What function each element has
- How each element relates to one another

The Big Picture

In short, it is the **big picture** or **overall structure** of the whole system — how everything works together

Factors That Need to Be Considered

1

Software architecture is tied to many practical concerns such as:

- Team structure
- Business requirements
- Development lifecycle phases
- Software intent and purpose

2

Software Architect's Considerations

- What will the system be used for?
- Who will be using the system?
- What qualities matter most to them?
- Where will the system run?

Why Do We Need Software Architecture?

Essential for Large Systems

Architecture provides a clear blueprint, addressing complexity and scale challenges.

Guides Development Efforts

- Helps developers understand:
  - What needs to be implemented
  - How elements relate to meet needs efficiently

Ensures Efficient Interaction

Defines element relationships to optimize performance and reliability.

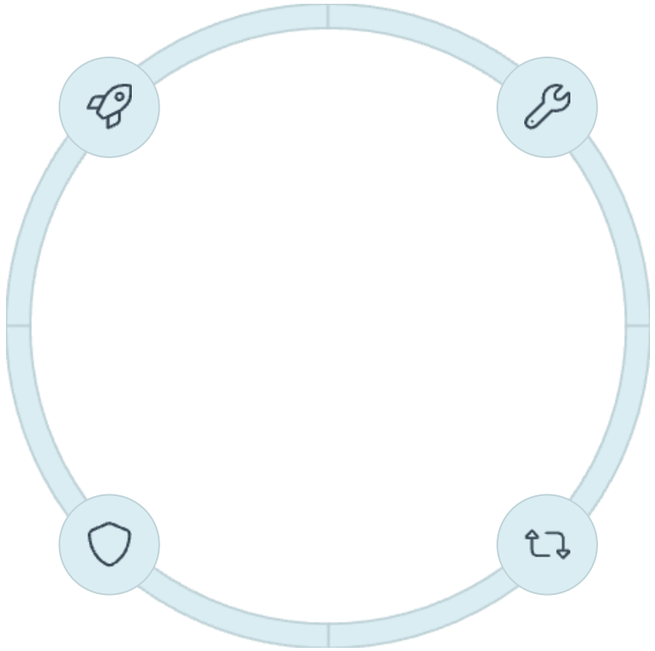
Advantages of Software Architecture

Higher Productivity

Clear design accelerates development and facilitates team coordination.

Achieve Quality

- Separation of concerns
- Maintainability
- Reusability



Effective Software Development

Defined structures prevent confusion and improve implementation quality.

Easier Changes

Well-structured architecture simplifies maintenance and adapts to evolving needs.

Software Architecture – Stakeholders

Developers

Use the architecture to organize work and guide implementation.

Project Managers

Leverage architecture understanding to manage risk and coordinate tasks.

Clients

Make informed decisions about funding and priorities based on architecture.

End Users

Benefit from system performance and user experience shaped by the architecture.

Stakeholders (continued)

Developers

Gain clarity on tasks and system organization easing development processes.

Project Managers

- Identify project risks early
- Understand dependencies across tasks
- Manage changes effectively

Clients

Rely on architectural decisions to shape investment and project direction.

End Users

Focus on final system performance and usability rather than internal workings.