**Why Software Architecture Matters??**

Software architecture matters because it's the foundation upon which a successful software project is built. Here's why:

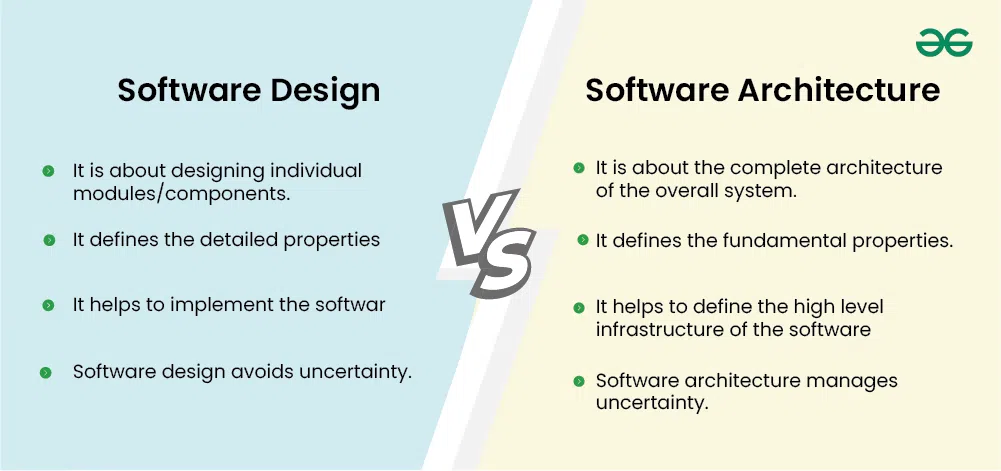
* **It ensures the system meets its requirements:** A well-defined architecture helps to ensure that the software system meets all its functional and non-functional requirements.
* **It makes the system easier to understand:** A good architecture makes the system easier to understand, which makes it easier to develop, maintain, and evolve.
* **It promotes reuse:** A well-defined architecture can promote reuse of components, which can save time and money.
* **It reduces risk:** A good architecture can help to reduce the risk of problems occurring during development and after the system is deployed.

**Software Design vs Architecture**

**What is Software Architecture?**

Software architecture refers to the fundamental structure or the process of creating the high-level structure of a software system. Initially, software architecture is created, and when it is agreed then that software design is carried out.

**What is Software Design?**

Software design refers to the process of creating a specification of software artifacts that will help developers implement the software. It is about designing individual modules/components means it defines what is module is performing, the classes, functions, and their usages, etc.****

**Lec#02**

**Why Software Engineering?**

Imagine you're building a house. You wouldn't just start laying bricks without a plan, right? Software engineering is like having that architectural blueprint for your software project. It's all about:

* Understanding the user's needs: What kind of software do they want? What problems should it solve?
* Keeping costs down: Building software can be expensive, so we want to do it efficiently.
* Making sure it works well: The software needs to be fast, reliable, and easy to use.
* Delivering on time: Nobody wants to wait forever for their software.

****

**Key Aspects of Software Engineering:**

* + It involves all stages of software production, from planning and designing to testing and maintaining the software after it’s released.
  + It’s not just about coding; it’s about solving problems, designing algorithms, and ensuring the software is reliable, efficient, and meets user requirements.
  + Software engineers use principles from computer science, engineering, and mathematics to build software systems.

**Goals of Software Engineering:**

* **Meeting User Needs:** The software should do what the users want it to do.
* **Low Cost:** It should be affordable to develop and maintain.
* **High Performance:** The software should run efficiently.
* **Portability**: It should work on different platforms (e.g., Windows, Mac, Linux).
* **Reliability**: It should work correctly and consistently.
* **On-Time Delivery:** The software should be delivered on schedule.

**Key Factors for Project Success**

To make a software project successful, you need:

* **Good Planning**: Properly define the project scope, budget, and timeline.
* **Clear** **Requirements**: Understand what the users need and build software that meets those needs.
* **Skilled** **Team**: Have a team with the right technical and managerial skills.
* **Testing and Quality Assurance:** Rigorously test the software to find and fix bugs before release.
* **Communication**: Ensure good communication between team members, stakeholders, and users.