**Design and architecture**

Software architecture and design are totally different, even though they do not always show that way from the outside. Software architecture serves as the planner for the system by giving an overview of how the system, that helps to make sure that software meets all requirements for operation and still has the quality attributes that are necessary , like security. Software architecture also includes determining how the software development will continue where it should fit into the existing architecture without making any weak. Software architecture looks at important elements like structural elements and their interfaces, the behavior of those elements, how the architectural decisions help meet business objective. Software design is a description of the plan of the system, which includes how the different elements of that system fit together and how they work together, and if they will meet the requirements of the system. Major artifacts of the software design process include Software requirements specification that describes the expected behavior of the system in the form of functional and non-functional requirements. These requirements should be clear, measurable, and traceable to business requirements, and high-level design that breaks the system’s architectural design into a less-abstracted view of sub-systems and modules and depicts their interaction with each other, and detailed design that involves the implementation of what is seen as a system and its subsystems in a high-level design. This activity is more detailed about modules and their implementations. The main different The main difference is that software architecture and design do the same things, through software architecture is a bit more hard and impactful on the overall system. Software architecture tools are important for software developers to build and design the basic software structure and go deeper to provide information about a system’s software architecture. Refactoring is the process of changing a software system in such a way that it does not alter the function of the code yet improves its internal structure.. Lack of legacy knowledge and software architecture documentation expand the process, often resulting in developers taking wrong turns and introducing faults in production. Software architecture principles order the process of converting and building software. They are put in place to help with flexibility, scalability, reusability, and security which are used to make sure that a solution meets business expectations. “Even the best architecture, most perfectly suited for the job, will be essentially useless if the people who need to use it do not know what it is, cannot understand it well enough to apply it, or (worst of all) misunderstand it and apply it incorrectly. All of the effort, analysis, hard work, and insightful design on the part of the architecture team will have been wasted.”**Documenting Software Architectures: Views and Beyond, Paul Clements, et. al**

الاسم: غدي محمد الشهري