Software architecture

Software systems are becoming very complex day by day. There is a need for coming up with top class design of the system before kicking off with the development of it. Enterprises are investing heavily in the design process of any software development to come up with efficient software systems (Tomi Männistö, 2014).

Any software system prior to its development would have gone through the phase of software system design. This includes answering many questions like what are the subsystem involved, what framework should be used, how the subsystems are controlled, how the systems communicate, what all components are necessary etc. The answers to these questions forms the **software architecture**. On a high level software architecture usually represented by **block** and **line diagrams** that depicts the main components and their interoperability. This helps the stakeholders relate and understand an **abstract view** of the system (Sommerville, 2011).

Architectural design is an innovative process and differs depending on the type of system being developed. All software architects will have to answer some generic questions to get started with the design of the software system keeping in mind the major system characteristics (Sommerville, 2011)

- performance ,
- security
- safety
- availability
- maintainability

Some major decisions made by the architect will involve the type of application, the distribution of the system, the architectural styles to be used (Sommerville, 2011).

The architecture plays a very crucial role in analysing and focusing on the **non-functional requirements** like quality, security etc. The software architecture decision helps to manage and calculate the risk related to the complexity of the system. There is a lot of thought that has to be put in designing a system taking into various factors like usability and performance. The architecture decisions are very important for the stakeholders and the actual end users. The design process must take into account the concerns of the stakeholders and how the requirements can be met (Tomi Männistö, 2014).

Software architect holds the main responsibility of taking architectural significant decision in a waterfall approach. Better decisions come with experience. They should have domain knowledge of both requirement engineering and development process and should act as an **interface** between the two teams. Some trade-offs should be made during the decisions of the architecture and these have to be approved by the stakeholders by the inputs provided by the software architect. For example if the stakeholders need a high security system and this may affect the performance of the systems, these should be explained to the stake holders and critical decisions have to made on actual facts and figures provided by the architects (Tomi Männistö, 2014). **Crosscutting** should be considers while doing the design i.e. number of components that will be affected if any change is made. All these architectural significant decision are to be made during the design phase with at most precision and standard. Most of the huge firms will have team of architects to make the design decisions (Tomi Männistö, 2014).

Agile development has made software architecture design very superfluous. Most of the agile developments used predefined architectures like Model View Controller (Sommerville, 2011). There also might be sprints for design to make minor changes in generic models of predefined system architecture (Tomi Männistö, 2014).

Architectures are documented from several different perspectives - a conceptual view, a logical view, a process view, and a development view (Sommerville, 2011). Most of the software systems will be a slight deviations from the major architectural designs like Model-View-Controller, Layers architecture, client-server model etc. Hence the reuse and documentation of architecture has helped to come up with dynamic and rapid design approaches.

Architectural significant decisions are very hard to refactor. Lot of capital and time is needed to change an architectural design decisions. It is very hard to change the system if they do not scale up due to bad decisions made. Good planning and organization is necessary for making efficient and effective architectural design decision.

References:

Sommerville. I. (2011) Software Engineering, 9th edition, U.S.A: Pearson Education, Inc.

Podcast interview between Prof. Tomi Männistö and Prof. Casper Lassenius, 2014

Pekka Abrahamsson, Muhammad Ali Babar, Philippe Kruchten (2010) “Agility and Architecture: Can They Coexist?”, IEEE Software, 25 (1), pp. 16-22.