On the role of a Software architect

By Fernando C.A, Pelayo S.L, Raquel S.S and Paula D.A | February 14, 2025

Software is constantly evolving, and its design must be able to evolve alongside it. This evolution is guided by the **software architect**, a key role dedicated to maintaining flexible designs and planning how to confront future changes.

A software architect represents the bridge between the **what** and the **how** of the project, the technical and non-technical solutions, and their role is to understand both worlds in order to be able to ask the right questions, as well as seeing beyond the solution, that is, understanding the problem.

It is said that software architects are *"the first response to product"*. This means that they must be able to mitigate and arbitrate product requirements, whether these are coming from professionals, customers or executives. Understanding both sides enables them to act as **arbitrators** who are able to balance the needs of the project with the technical feasibility of the solution. This also allows the solution not to be **coupled to implementations**, which often makes systems struggle.

Why do we need design?

It's needed to plan how new features affect the old and future ones. Otherwise, we won't know where to start, or even break everything with this addition.

Moreover, design isn't a task done just at the beginning of the project, it's an ongoing task that the software architect owns but done with the collaboration of all the development team.

How do you become a software architect?

A software architect is a person that starts as a programmer and over the years, he wants to become a better professional by having a bird eye view, instead of specializing in a concrete methodology or technology.

That is, they are interested in understanding how all components interact, facilitating integration, and providing feedback to improve the product.

So, a successful architect can maintain a vision of what architecture and design should be and ensure that the design vision is consistent and flexible through the whole project, as the design can change and should be adapted. Basically, good designs are **dynamic**, as static changes will almost certainly always break.

It's not just creating a design and leaving it, instead it should be continuously improving and revisiting old designs that used to work but no longer meet the requirements.

They also have the job of facilitating software development to implement the right features through that design or vision. They help developers successfully deliver on the actual requirements, and add value to the product.

Skills of a software architect

There are two types of skills:

- **Soft skills**: they are focused on the more personal side, the process and people skills. That is, learning to be a person that everybody comes to for help as they perceive you as confident and trustworthy.

  Some specific soft skills are being a good listener and being a mediator. They need to facilitate the conversation and translate between technical and not technical, so both sides understand each other.

  The architect has to be able to make others see their point from a non-adversarial way. This is because people react poorly to change, so it is fundamental that software architects act as **mediators** instead of

gatekeepers, that is, they must help people who established wrong requirements to understand what the right requirements are. Basically, *figure out the problem together*.

- **Hard skills**: they are focused on the technical and the design skills.

In most of the cases architects are skilled in the hard skills, but lack the soft skills so they learn them after.
In any case, you can learn both types of skills.

What are the challenges of software architects?

These are the difficulties that software architects deal with:

- **Not spending a proper time on design** due to the executive's pressure to get things done as fast as possible to meet the deadlines. But this provokes many delays and changes due to a poor design or planification.
- **Not being listened to** as they're not the owners of the product. Also the team leader wants to start coding instead of spending time on designing, or even if they do it they have a very narrower scope.
- **Spending much time on writing and maintaining useless documentation**. Documentation should be quick, clean and reflect the design so it helps to understand it.
- **The constraints**, as something that comes in all levels and you cannot negotiate it.
  One of them is **quality**. Quality is the degree to which a system satisfies the needs of its stakeholders, providing value, and it's everyone's job to improve it.

The problem of not having a software architect

Knowing that software architects are in charge of **planning**, it is no wonder that the lack of them brings several problems to the table. Firstly, software becomes less organized, which results in an increase in **technical debt** (which is tightly coupled to complexity). From an architect's perspective, if they are moved to a new project, they lose the continuity with ongoing changes, which leads to a lack of ownership and pride in seeing the project come to life.

During the **Waterfall** model era, the role of the architect was clearer. They worked closely with others in the design and actively got involved in the process, overseeing the execution.
However, with the rise in popularity of **Agile**, there's less need for the architect to be actively involved (apart from the initial design), so the architect takes a more consultative role and the rest of the team assumes greater responsibility.

However, no matter the model, it is crucial that the software architect remains involved throughout the entire development cycle of a software product, as the architecture will almost certainly suffer from changes and adjustments, and it is their role to help the project evolve along these changes.

Final thoughts

A software architect plays a vital role in ensuring that a software product remains adaptable and efficient throughout its development. They are in charge of guiding the project's evolution, planning both immediate and future needs to ensure efficient adaptation to change.

Architects are not just designers, but also arbitrators who help teams react to changes, and they must account with both technical knowledge and soft skills to mediate between both worlds. This is why their role is so important and must be taken into account for producing successful software products.

References

- SE-Radio.net - Podcast 616
- Theory slides: Unit 2, Unit 4 and Unit 5