Architect

An architect is a person who plans, designs, and reviews the construction of buildings.

A **software architect** is a software expert who makes high-level design choices and commands technical standards, including software [coding standards](https://en.wikipedia.org/wiki/Coding_standard), tools, and platforms. The leading expert is referred to as the *chief architect*.

Almost every project needs an expert to make high-level design choices and define software coding standards, tools, and platforms. I talk about software architect, a person who will optimize your development process and your business as a result. Who can become a software architect? What software architect’s skills are the must? I’ll talk about the role. skills and duties of a software architect.

In the software industry, the role of a software architect is interpreted in many different ways. In some cases, an architect may work in an established enterprise company and hand down instructions on technology stacks to the developers. At the other extreme Agile development, a team may work without the involvement of an architect. Let’s clarify software architect’s role.

**The role of a software architect**

1. A software architect needs to *interact with clients*, product managers, and developers in order to *envision, model and provide initial models* and designs that can be built. This role also may cover the meeting potential or current customers.
2. A software architect has to constantly *review the code* to ensure the quality of the design by avoiding complexity, advocating clarity and to do this with the team. This usually requires hands-on work in terms of developing prototypes, contributing code or evaluating technologies.
3. The role of a software architect includes *collaborative working* with a degree of humility and providing mentoring as required. Such collaboration also allows the architect to become familiar with the skills and interests in the team and to share their knowledge with the rest of the team. Humility is required to ensure that all the team is listened to, as they may have more specific experience or knowledge for the problem at hand.

Taking into account all of the main aspects the software architect role includes, its obvious that this person should have knowledge in programming, management, psychology, communication and even finance. So, what are the main skills and qualities this specialist must have?

**The main characteristics of a software architect**

* *Broad and deep technical knowledge*. This should be obvious since one cannot become a software architect with a musical background. The architect usually has knowledge in several technological stacks at a decent level and should have a good understanding of a few other ones. The software architect should also be prepared to compose a large number of technical documentation, reports, and diagrams.
* *Responsibility*. A software architect should understand those architect decisions are usually the most expensive. A person in this position should take the most responsible approach to his work and to the decisions made. If the developer’s error costs a couple days of work of one person, then the architect’s mistake can cost person-years on complex projects.
* *Communicability*. A good specialist should be able to talk with customers in the language of business, managers of all levels, business analysts and developers in their languages. To explain all the action correctly, a software architect has to grow a natural charisma and ability to convince people. Usually, architects are laconic, eloquent and competent speakers. While software architects participate in discussions they should be able to persuade the others.
* *Management skills*. This includes both organizational and leadership skills. The ability to lead a team, which may be distributed and composed of very different specialists.
* *Stress resistance*. A software architect works with different people from different areas, rapidly changing demands or even with changing business environments. Therefore, it is necessary to be ready for stress and to look for some ways to escape negative emotions. Work is always more pleasant when you’re happy.
* *Analytic skills*. One of the most important tasks is the ability to represent an abstract problem in the form of some finite real object of the system, which can be evaluated, designed and developed.

**Main responsibilities of a software architect**

The most important responsibility is *complete technical support* of the project from the moment of inception, through product release, to development of enhancements. The other responsibilities considered among the main ones are:

● Identifying business requirements and requirements of the stakeholders on the project

● Designing the entire system based on the received requirements

● Choosing the system architecture and each individual component of this system at a high level

● Choosing the technologies for the implementation of each component and connections between the components

● Architectural review

● Code-review

● Writing project documentation and its support

● Creating unified development standards in the company

● Controlling the architecture during the next iteration of the system release

There are a lot. Crafting the right architecture to solve the problem at hand is only part of architects’ responsibilities. They must also:

* control over correct using the architecture
* control over timing and deadline
* control over synchronization of the software with the system architecture
* do performance quality control
* give input as needed to issues like the tool and environment selection
* interact with management and stakeholders
* resolve disputes and make tradeoffs
* resolve technical problems
* understand and plan for evolutionary paths
* plan for new technology insertion
* manage risk identification and risk mitigation strategies associated with the architecture

So, to become a software architect, you need to pass a long way of learning and improvement. Understanding several technological stacks is a must: server languages, iOS, Android and more… You have to read a lot of professional literature and find some mentor to ask questions. Don’t underestimate the influence of different courses and workshops. Be aware that the path of becoming a software architect will take at least several years.

**Do you need to have your own software architect?**

Not many companies can allow themselves to grow their own specialist in software architecture. That makes sense only if your main field is software development. Otherwise, you’ll ask for advise or collaborate with software development company to provide your project with their software architect. Startups have no time to grow their own software architects and no money to invite an experienced one. And a startup of two people, for example, cannot afford to have one team member focus on just the architecture. Everybody has to share the task and wear multiple hats.

The question of having your own software architect depends on many aspects like:

1. What industry does your business belong to? (banking, technology, telecommunication, broadcasting, etc …) Most technological problems are already solved, what most businesses need are implementers.
2. What is the size of your organization? Does the team of two need a software architect? Really, when was the last time you’ve heard of a successful small business/startup that had a dedicated software architect?
3. What is the size of your development team? If we take an example of another case, let’s say, a big business with software department. How does having a dedicated software architect role benefit your department?
4. What is the expected output of your team? (software products, services, innovation) To create a simple website you definitely have no need for a software architect.
5. Do you already have a development team or are you building a team?

So having your own specialist for high-level design choices is not always a good idea. However, in most cases, an outsource company has all kinds of specialists to develop the project of any complexity.