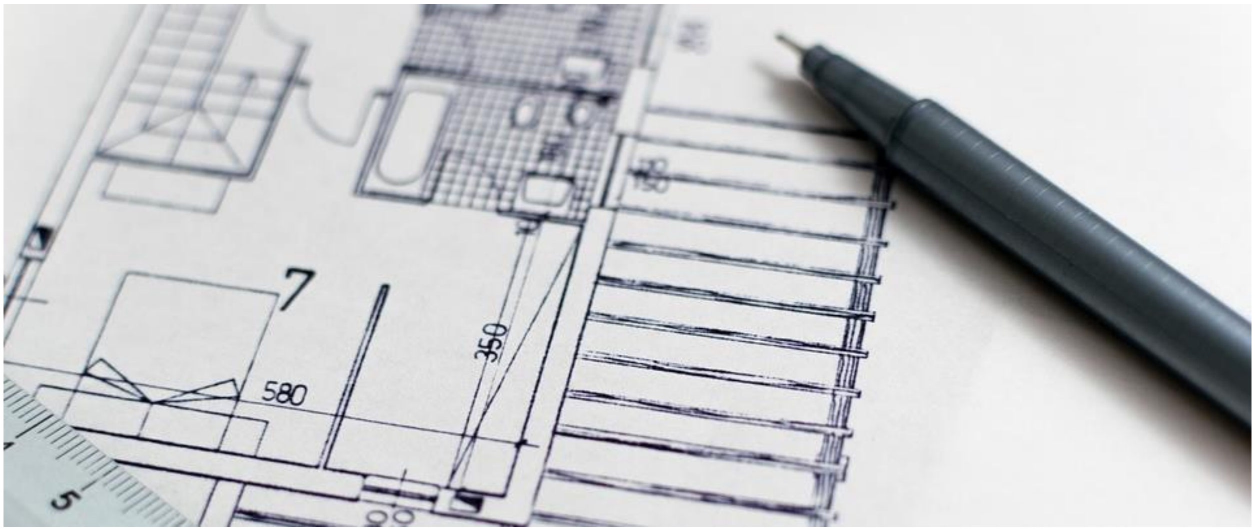
The Architect Defined



Introduction

I have been a software architect for over 20 years, worked on a wide range of projects, and have learned a few things. The most valuable thing I've learned is that I have a lot to learn. I write this not as one who has mastered the role, but rather one who has had a few successes interspersed among many mediocre outcomes and one or two abject failures. The skills and behaviors listed below are things I wish I knew or at least appreciated more fully 20 years ago.

The world changes around you. As you focus on one thing, new technology appears, practices evolve, tools improve, and before you know it, the landscape is unrecognizable compared to where you started. But despite the constant change, some patterns have emerged and have become associated with architects who are good at their craft. This article attempts to inventory those essential patterns. The software architect role is difficult to define. In the most general sense, the architect (arch) is perhaps ultimately responsible for project delivery and whatever that entails. In practice, the role can be distilled into a few key skills and behaviors.

For an excellent presentation on essential arch characteristics, see So you're an architect. Now what? by Nathaniel Shutta.

Some time ago, Martin Fowler published a great article entitled "Who Needs an Architect?". In it, he describes two species of architects – *Architectus Reloadus* and *Architectus Oryzus*. Fun monikers that represent the ivory tower law giver and the intense collaborator respectively.

It is my opinion, my beloved reader, that an architect must play both roles depending on the type of organization. Product development organizations, ones in which there main source of revenue is software, the law giver is needed to maintain conceptual integrity.

In a DevOps shop, the architect functions as a change manager. In order to manage change, he must collaborate with his colleagues.

The TL;DR is that a “good” architect must exhibit all of the traits to design, manage, lead, and mentor. Read on for my opinion.

Designer

First, an arch must be able to think through the structure and function of a solution. He or she must embrace all the requirements, account for all the constraints, address all the desired ‘illities, and arrive at the simplest possible design that still works. The arch must be fluent in notations that convey ideas succinctly while not showing off obscure techy jargon.

On a small project, this is easy. Whip out the domain model or process flow and truck on. On larger projects, the arch decomposes the problem into smaller ones, aligned perhaps along organizational boundaries (see Conway’s Law, because that’s the thing the kids reference these days. There might even be a meme by now).

On live, running systems, the arch but be able to change the oil with the engine running. By knowing as much about the system state as possible, by deputizing others to own parts of the system, the arch manages change. He or she must ensure that nothing is introduced into the system that could negatively affect stability or performance.

Leader

Software development is a human endeavor. The challenge for the architect us to model the way. He or she must be credible in the eyes of the teams and be someone people are willing to follow. The architect should never expect anything from anyone that he or she is not willing to do himself. The arch should provide a framework that stretches the team but not so much so that the objective is unattainable. The architect must also work to resolve conflict. Conflict resolution must be done in a helpful and constructive manner.

All of the material in this article refers to skills, behaviors, and responsibilities that are attributed to the arch. A good leader will work to delegate and develop others to fulfill many of activities. One person cannot do it all.

Time Manager

The architect is a busy person. The day is filled with whiteboard discussions, meetings, status reports and head's down work. In addition there's an endless procession of interruptions. The architect must be good at keeping the main thing the main thing, while making sure no one is blocked are no task is ignored. Note the good

DevOps

DevOps means different things to different people. A person is in DevOps when they understand and are responsible for delivering functionality but also concerned about it working day in and day out. There are two main flavors of activity – The Dev and The Ops.

The Dev

The Dev in DevOps means to set up the processes and create the tooling to produce software on a regular basis. The process must work for solving business problems and delivering solutions while at the same time adhering to policies, regulations, and other constraints. Think SOX compliance.

The Ops

The ops (in my definition) is the deployment of solutions and the monitoring of the production system. The boundary between Dev and Ops is continuous integration / continuous delivery in which code is always being prepared for release. The arch must work with the team to develop and evolve the CI/CD process. The arch must also have a good understanding of the system underpinnings. Like Scotty, he or she must know the health of the system just be feeling the deck plates. In reality, the arch must know and monitor the system KPIs to keep the system operational.

Communicator

I tend to struggle the most with communication (as you might be able to tell from my writing). It's important to know your audience. Developers and other technical types are fine with details, but can be easily sidetracked into shiny object discussions. Upper leadership need to understand, but do not need to know all the details. If you share too much detail with leaders, then they may feel responsible for the information and try to manage on those terms.

Diplomat

Anytime you get a group of humans together, a struggle of personalities will eventually emerge. The arch is in a perfect position to mediate many such struggles. Now, an arch may not necessarily be a trained counselor, but they should expect to be the toxic avenger from time to time. The arch must be a voice of reason and calm in difficult situations.

Project Manager

What? Shouldn't the Project Manager manage the project? Of course. But, project success is the result of countless micro decisions. The architect realizes this and, in the interest of quality, may find himself at odds with the Project Manager. However, the arch balances the need for quality with the drive to deliver, making scope or schedule tradeoffs along the way.

Related to project management is estimation. In between the solution and the plan is the estimate of what it will take to deliver. For this reason, the arch must understand the requirements, the complete technology stack, and resourcing constraints. Additionally, the arch must have a methodology in his tool belt to build a defensible estimate.

Business Analyst

In order to conjure up a design, the arch must grasp the requirements to a great level of detail. A byproduct is that the arch may become familiar with the visits m business to at least the same level as a business analyst. The arch should leverage this knowledge to build relationships with the users and stakeholders as well as assist the developers with building the solution.

Quality Assurance

By now, it must seem like the arch has to do every job on the project. That's not entirely accurate. Instead, the arch must be *able* to just about every aspect of the project, including quality assurance. The arch guides the team in unit test case development, helps the business analysts with requirements-driven test case development, and likely owns nonfunctional testing such as performance and disaster recovery.

Teacher

Finally, the arch must be able and willing to pass it on. Being a good teacher helps grow others and in turn raises the caliber of the team. Also, teaching others avoids key-man dependency.

Conclusion

The architect must wear a lot of hats. It is a paradox of low definition and high responsibility. I am not writing this because I do this well, or even at all. These traits however are meant to set the boundary and the bar to do well in terms of your career and your team.