



Technical Interviews & Assessments for Software Architects

Software architecture could be described as the process of converting software characteristics, such as flexibility, scalability, feasibility, reusability, and security, into a structured solution that meets the technical and business expectations. During your interview process, it is important to demonstrate that you are able to architect products that align with this description.

- Within the software industry, there are multiple different types of Architects such as (to name a few):
 - Solution Architect
 - Enterprise Architect
 - Cloud Architect
 - Data Architect
 - Application Architect
 - System/s Architect
 - Security Architect

Key Characteristics

In the general sense of a Software Architect, these are a few of the characteristics which will be advantageous to showcase during your interview process. Give examples of how you've showcased these characteristics in your role and how you would show them going forward:

1. Communication skills

You have to speak with customers in the language of the business, developers, business analysts, and managers of all levels. Focus on ensuring that your speech is eloquent, succinct, and competent and that this is conveyed strongly during the interview.

2. Broad and deep technical knowledge

Your expertise in several technological stacks at a decent level and others of which you have a good understanding will be integral for your role as an architect. You would be required to compose a large number of technical documents, reports, and diagrams. Be prepared to showcase what tech stacks you have experience with and how your experience/exposure ranges for them all.

3. Stress resistance

You will be required to make many decisions, work with different people from different areas, and you will be required to deal with rapidly changing demands, or even with changing business environments. It's important to be ready for stress and for healthy stress-coping mechanisms. Have examples ready for when in your work life you've had to deal with stressful situations and stress and how you went about that.

4. Management skills

Being able to showcase both organizational and leadership skills is essential in this role. Be able to showcase how you have shown and used these skills in previous positions and moments you showcased them competently and effectively.

5. Analytics skills

One of the most valuable tasks is the ability to represent an abstract problem in the form of some finite real object of the system, which developers are already evaluating, designing, and developing. So being able to analyse and communicate these tasks effectively to the members of the team and the customer is essential.

Common Practical Assessments/Case Studies

More often than not, the way for the company to assess your technical competency as an architect would be through a case study scenario. It could be a possibility that they give you a Senior Developer assessment to complete, company-specific, or something like an assessment from Hackerrank/Codility [see [Technical Interview prep PDF](#) for these assessments.]

With regards to the case study scenario, they could present you with a scenario (that you may need to use a whiteboard for) where you need to design a system that does x, y, and z. To make sure you have a good understanding of the initial scenario, be sure to ask them questions such as, what does the application look like (are there API endpoints, what kind of response would one get), what problems would you encounter. Ask for specificity and make sure you have a clear picture of what they want and need from the system you are to design.

Once you have sufficient information you can start designing the system. Vocalize your thoughts as you go about it and as a starting point, try to stick to a simple solution instead of overcomplicating it, as well as optimizing for a technology-agnostic solution. You should be able to comfortably explain not only the technological aspects of the problem (such as the back and frontend of the Web technology, databases, etc) but also to demonstrate strong communication skills.

The next part of the case study could be to scale the system you have just designed. So think through and vocalize how you would scale it, where do you think things would break and what changes you would make. Ask for feedback, so that if there are any points of failure, you would be able to showcase what you would do to architect them away, or if not possible to be architected away, how you would mitigate any problems that happened if (and when) it failed.

When looking at your design, you should be able to easily discuss the pros and cons of each design decision, as well as being able to see common problems and potential failure scenarios.

Two other potential case study scenarios could be:

- How would you refactor system X?
- How would you debug/profile and fix application X?

Q & A Examples

1. Tell me about a time where you created an architectural road map.
Highlight: A solid understanding of architectural road maps, critical thinking skills, and experience with different domains.
2. What is the most complex project you have worked on and was it difficult for good or bad reasons?
Highlight: Positive disposition toward challenges, honest account of working on hard projects, and learning something from the challenge and realization of any patterns and anti-patterns. You should explain this in-depth, ranging from the planning, problems, team-building with engineers, time spent, and the obstacles you faced.
3. What does SOLID stand for and what are the principles behind it?
Highlight: Awareness of who Robert C. Martin is, understanding of the five core concepts and appreciation for the fundamentals of object-oriented design.
 - The 5 principles: single-responsibility, open-closed, Liskov substitution, interface segregation, and dependency inversion.
4. What problems does Architecture Analysis solve?
Highlight: Your understanding of Architecture Analysis and how it's used.
 - Defects that lead to security problems come in 2 major forms; bugs in the implementation and flaws in the design. Highlight knowledge about how to identify and solve these problems:
 - Analyze fundamental design principles
 - Assess the attack surface
 - Enumerate various threat agents
 - Identify weaknesses and gaps in security controls
 - Know the three potential avenues you can take in finding and fixing design flaws; Architecture risk analysis (ARA), Threat Modeling, and Security Control Design Analysis (SCDA)
5. What is Dependency Injection?
Highlight: Your understanding of what this means, and be able to provide an example of this in your work.
 - What is it: It's a technique whereby one object supplies the dependencies of another object, it's made part of the client's state and it is also a creational design pattern.
6. What kind of design patterns do you know?
Highlight: Your understanding of the 3 basic types of design patterns (structural, creational, behavioural) as well as be able to provide definitions for each and why you would use them.
7. What kind of architectural patterns do you know?
Highlight: Your understanding of at least 3-5 software architecture patterns, and to be able to give the pros and cons of each.
 - Some of the patterns: Layered (n-tier) architecture, event-driven architecture, microkernel architecture, microservices architecture, space-based architecture, CQRS and event-sourcing.

Blog Posts

Check out these articles from our blog – they may be useful for you in preparing for your interview processes and beyond 🎉

- [How I got Started With Data Visualisation](#)
- [From Sedan to Sports Car - Designing Software for Performance](#)
- [A 3-Step Process to Communicate More Effectively as a Developer](#)

Resources:

- <https://medium.com/@nvashanin/the-path-to-becoming-a-software-architect-de53fcb310a>
- https://en.wikipedia.org/wiki/Software_architect
- <https://martinfowler.com/architecture/>
- <https://insights.dice.com/2018/07/18/solution-architect-interview-questions-soft-skills-vendor-options/>
- <https://codeburst.io/software-architecture-the-difference-between-architecture-and-design-7936abdd5830>
- <https://www.indeed.com/hire/interview-questions/enterprise-architect>
- <https://www.susanfowler.com/blog/2016/10/7/the-architecture-interview>
- <https://www.wisdomjobs.com/e-university/software-architecture-and-design-interview-questions.html>
- <http://www.7loops.com/software-architect-interview-questions/>
- <https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013>
- <https://techbeacon.com/app-dev-testing/top-5-software-architecture-patterns-how-make-right-choice>
- <https://dzone.com/articles/software-architecture-the-5-patterns-you-need-to-k>