**19**

**Learning Soft Skills to Become a Better Solution Architect**

In the previous chapters, you learned how a solution architect needs to accommodate all stakeholders' needs. Even if the solution architect's role is technical, they need to work across the organization, from senior management to the development team. To be a successful solution architect, soft skills are essential and critical factors.

Solution architects should keep themselves up to date with current technology trends, keep evolving their knowledge, and always be curious to learn new things. You can become a better solution architect by applying continuous learning. In this chapter, you will learn about methods to learn new technologies and how to share and contribute back to the technical community.

Solution architects need to define and present an overall technical strategy to address business concerns. They need to work across business and technical teams to negotiate the best solution, which requires excellent communication skills. In this chapter, you will learn the soft skills a solution architect must have, including the following:

* Acquiring pre-sales skills
* Presenting to C-level executives
* Taking ownership and accountability
* Defining strategy execution and objectives and key results
* Thinking big
* Being flexible and adaptable
* Design thinking
* Being a builder by engaging in coding hands-on
* Becoming better with continuous learning
* Being a mentor to others
* Becoming a technology evangelist and thought leader

By the end of this chapter, you will know about the various soft skills required for a solution architect to succeed in the role. You will learn about methods to acquire strategic skills (such as pre-sales and executive communication) and develop design thinking and personal leadership skills (such as thinking big and ownership). You will learn about techniques to establish yourself as a leader and continue improving your skill set.

**Acquiring pre-sales skills**

Pre-sales is a critical phase for complex technology procurement, whereby the customer collects detailed information to make a buying decision. In the customer organization, a solution architect is involved in the pre-sales cycle to validate technology and infrastructure resources from various vendors. In the vendor organization, the solution architect needs to respond to customers' **requests for proposals** (**RFPs**) and present a potential solution to acquire new business for an organization. Pre-sales requires a unique skill set that combines strong technical knowledge with soft skills, including the following:

* **Communication and negotiation skills**: Solution architects need to have excellent communication skills to engage the customer with the correct and latest details. Presenting precise details of the solution and industry relevance helps customers understand how your solution can address their business concerns. Solution architects work as a bridge between the sales and technical teams, which makes communication and coordination a critical skill. Solution architects also need to create agreements by collaborating with customers and internal teams, which requires excellent negotiation skills. In particular, strategic-level decisions have a significant impact across multiple groups. Solution architects need to negotiate between the team, work on trade-offs, and develop an optimized solution.
* **Listening and problem-solving skills**: Solution architects need to have strong analytical skills to identify the right solution per customer need.

The first thing is listening to and understanding customer use cases by asking the right questions to create a good solution. You need to understand gaps and develop a solution to result in immediate business impact with long-term **return on investment** (**ROI**). For some customers, performance is more important, while others may be more focused on cost based on their application's user base. The solution architect needs to provide the right solution per their customer's primary **key performance indicator** (**KPI**) goal.

* **Customer-facing skills**: Often, the solution architect needs to work with both the internal team and the external customer's team. They influence stakeholders at all levels, from C-level executives to development engineers. They present solutions and demos to senior management, who look at your proposal more from a business perspective. C-level executive support and commitment to initiatives always result in the success of the adopted solution, which makes customer-facing skills very important. The C-level executive needs details of the solution in a defined time-bound meeting, and the solution architect needs to utilize the allotted time to their best advantage. You will learn more information about the executive conversation in the next section of this chapter—*Presenting to C-level executives*.
* **Working with teams**: The solution architect establishes a relationship with the business team and the product team. To prepare an optimal application, the solution architect needs to work with the business team and technical team at all levels. The solution architect needs to be a good team player and work with multiple teams, share ideas, and find a way of working.

The skills mentioned above are not only required for pre-sales but are also applicable to the solution architect's day-to-day job functions. Solution architects come from a technical background, and, being in such a role, they need to acquire critical skills to communicate at an executive level. Let's learn more about executive conversations in the next section.

**Presenting to C-level executives**

A solution architect needs to handle various challenges from a technical and business perspective. However, one of the most challenging tasks could be to get executive buy-in. Senior executives such as the **Chief Executive Officer** (**CEO**), **Chief Technology Officer** (**CTO**), **Chief Financial Officer** (**CFO**), and **Chief Information Officer** (**CIO**) are regarded as C-level as they have a tight schedule and need to make lots of high-impact decisions. As a solution architect, you may have lots of details to present, but your C-level meetings are very time-bound. Here, a solution architect needs to get the maximum value out of their meeting in the allotted time slot.

The primary question is: *How do we get senior executives' attention and support in a limited time?* Often, during any presentation, people tend to put a summary slide at the end, while, in the case of executive meetings, your time may further reduce as per their priority and agenda. The key to an executive presentation is to summarize the primary points upfront in the first 5 minutes. You should prepare in such a way that if your 30-minute slot reduces to 5 minutes, you should still be able to convey your points and get buy-in for the next step.

Explain your agenda and meeting structure even before the summary. Executives ask lots of questions to utilize their time properly, and your agenda should convey that they will get the chance to ask a clarification question. Support your summary with facts and data that align with their industry and organization. Keep the details with you in case they want to dive deep into a particular area; you should be able to pull up and show all the data.

Don't try to present everything in detail by stating information that may seem relevant from your perspective, but maybe doesn't make much sense for an executive audience. For example, as a solution architect, you may focus more on the benefits from the technical implementation. However, senior management focuses more on ROI by reducing operational overhead and increasing productivity. You should be ready to answer the following questions that concern executives more:

* **How will the proposed solution benefit our customers?**: Business revolves around the customer. While executives are looking at their company's growth, that is only possible if their customers are satisfied. Make sure to do your research on their customer base and their needs. Be ready to present benefits backed by reliable data.
* **What assumption did you make to baseline the solution?**: Often, these meetings are at the initial phase when you may not have enough details. Solution architects always need to make some assumptions to baseline the solution. List down your hypothesis in bullet points, and have a mitigation plan associated with it, in case things don't work as per assumption.
* **What will be my ROI?**: Executives are always looking for ROI by determining the **total cost of ownership** (**TCO**). Be ready with data to provide an estimated cost of ownership, solution maintenance costs, training costs, overall cost savings, and so on.
* **What happens if we continue as it is today and do nothing?**: Senior management may go into extreme vetting mode to identify ROI. They want to understand if the investment is worth it. You need to be ready with your market research—for example, technology trends, customer trends, and the competitive situation.
* **What will be our competitor's reaction regarding your solution?**: Competition is everywhere, and often, the executive worries more about it. They want to understand if your solution is innovative to beat the competition and give their organization the edge. It's better to do some upfront research and add competitiveness data relating to their industry and customer base.
* **What is your suggestion, and how can I help?**: You should always have a crisp list of action items as the next step while providing your suggestion. You need to get buy-in from executives and make them feel involved by asking for help. For example, you can ask the CIO to connect you with the engineering team or product team to take an overall solution to the next step.

Till now, in this chapter, we have talked about various soft skills, such as communication, presentation, and listening. Let's now look more at the leadership skills a solution architect should have as a technical leader for the organization.

**Taking ownership and accountability**

Taking ownership and positioning yourself as a leader helps you to win trust with accountability. Ownership doesn't mean that you need to execute things alone; it is more about taking new initiatives and holding on to them as it is your organization. You can have ideas that can benefit your organization in terms of productivity, agility, cost savings, and increasing the customer base. Sometimes, you may not have the time or resources to execute your idea, but you should always try to bring it forward as a new initiative and engage others for execution.

Accountability is about taking responsibility to drive the outcome. Ownership and accountability go hand in hand, where you are creating initiative and working on getting the result. People can trust you to execute any job and drive results. Accountability helps you build trust with your customers and team, which ultimately results in a better work environment and achieving a goal.

As a solution architect, when you take ownership, it helps you see things from the customer's and sponsor's perspectives. You feel motivated and a part of something meaningful that you enjoy doing. Make sure to define and create key successes and the objective key result. The goal/objective should be measurable using specific key results, and they must be time-bound. Let's learn more about **Objectives and Key Results** (**OKRs**).

**Defining strategy execution with objectives and key results**

Strategy execution is complex and challenging. Excelling in strategy execution is essential for realizing the organizational vision, mission, and goals. The idea needs to be converted into actionable elements to keep teams aligned and everyone moving in the same direction. Goal setting and managing goals are some of the best-established ways to get things done.

**Objective Key Results** (**OKRs**) are principles and practices (vision and execution) of goal setting. OKR is a strategy management system that focuses on strategy execution. It is a simple framework that lets you define the organization's primary strategy and its priorities. Objectives are the principles, and key results are the practice—it is a *what and how* of organizational vision. OKRs are based on four superpowers, as illustrated in the following diagram:

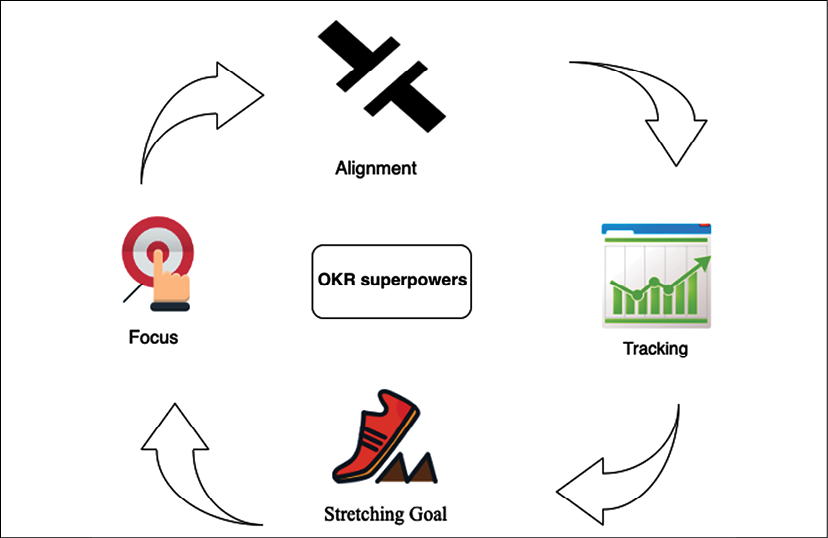


Figure 19.1: Superpowers of OKRs

OKRs' superpowers include the following:

* **Focus**: Start with the question: *What are our main priorities, and where should people concentrate their efforts?* Commit to what truly matters and provide clarity on what is essential.
* **Alignment**: Make goals public and transparent. Connect with the team and get cross-team, bottom-up, and sideways alignment.
* **Tracking**: Visually track the key results of each objective, down to the percentage point.
* **Stretching Goal**: Create ambitious goals to achieve something remarkable. Stretching goals allow people to reimagine and rethink.

OKRs provide visibility and a meaningful outcome to all stakeholders at various levels, from executive sponsors to teams. OKRs make the vision and mission of the organization clear. Team members that are working on day-to-day activities need visibility and clarity to the mission. They need to see how their everyday work has an impact on that organizational mission. The OKR framework allows you to define this link and provide visibility and meaning for everyone on the team.

**Thinking big**

Solution architects should have the ability to see the big picture and think ahead. A solution architect creates a foundation upon which the team puts building blocks and launches the product. Thinking big is one of the critical skills that solution architects should possess to think about the long-term sustainability of an application. Thinking big doesn't mean you need to make a very unrealistic goal. Your goal should be big enough to challenge you and bring you out of your comfort zone. Thinking big is critical for success at both a personal and an organizational level.

You should never doubt your capability while thinking big. Initially, it may seem challenging to achieve, but you will find the way as you start working toward the goal. Believe in yourself, and you will notice that others start supporting and believing in you. Thinking big helps to inspire people around you to become a part of your success. Set up long-term goals, such as *where you want to see yourself and your organization in the next decade*. Take one step at a time to gear a short-term goal to a long-term goal.

Once you set up the stretching goal by thinking big, it will help you take the initiative and explore new challenges. However, you need support from your peers and team to deliver the result, who can provide you with the right feedback and extend help as needed. Become a person that people want to help; of course, this is a two-way door. To get help, you need to be open to helping others. Adaptability is another critical skill for solution architects to work with others. Let's learn more about it.

**Being flexible and adaptable**

Adaptability and flexibility go hand in hand, and you need to be flexible to adapt to the new environment, working culture, and technology. Adaptability means you are always open to new ideas and to working with the team. Teams may adopt a process and technology that is best suited for them. As a solution architect, you need to be flexible in accommodating team requirements during solution design.

For example, in a microservices architecture, each service communicates via a standard RESTful API over the HTTP protocol. Different teams may choose to write code in a different language or tool of their choice, such as Python, Java, Node.js, or C#. The only requirement is that teams need to expose their APIs securely so that the entire system can build upon utilizing them.

You need a different mindset and perspective to look into the problem to get a more innovative solution. Encouraging teams to fail fast and innovate helps an organization to be competitive. The personal traits of flexibility are demonstrated by the following:

* Thinking about various solutions to solve a problem with the team and take the best approach
* Helping team members to offload their work
* Volunteering to fill up a place if a team member needs to take time off for weeks due to personal work reasons
* Being able to collaborate effectively with teams across different locations and time zones

You need to be open-minded and adaptable to changes in technology and processes. You may face resistance when bringing change to your team or organization. You need to encourage others to be flexible and convey the importance of change. For example, when an organization wants to move its workload from on-premises to the cloud, they often face resistance, as people have to learn a new platform. You need to explain the value proposition of the cloud and how it will help them be more agile and innovate faster.

As a solution architect, you need to be adaptable to carrying out multiple assignments and setting the right execution priority. You should have the ability to adjust to the situation and work under pressure. A solution architect needs to have critical design thinking to create an innovative solution. Let's learn more about design thinking in the next section.

**Design thinking**

A solution architect has the primary role of system design, which makes design thinking an essential skill. Design thinking is one of the most successful approaches adopted across industries to solve a challenging and unclear problem. Design thinking helps you to look at problems and solutions from a different perspective, which you might not have considered in the first instance. Design thinking is more focused on delivering results by providing a solution-based approach to solve the problem. It helps to question the problem, solution, and associated risk, to come up with the most optimized strategy.

Design thinking helps you redefine problems in a more human-centric way by putting yourself in the place of end users and customers. The following diagram illustrates the primary principles of design thinking:

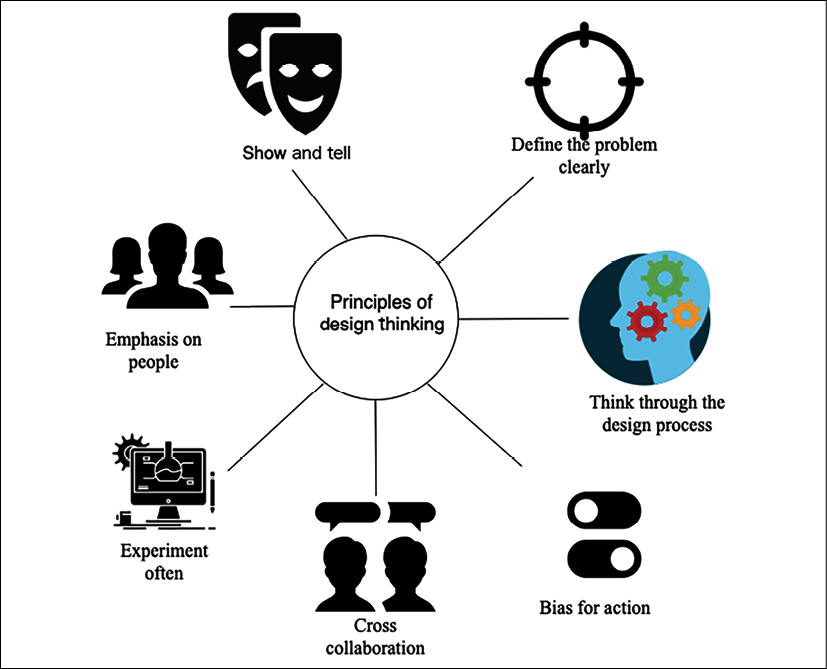


Figure 19.2: Principles of design thinking

The following points are some design-thinking principles:

* **Emphasis on people**: Collect feedback from various users and put yourself in their place to understand the problem from a different perspective.
* **Cross collaboration**: Bring in people from different backgrounds to look for problems in a diversified way, and make sure solutions accommodate everyone's needs.
* **Think through the design process**: Understand the overall design process, with clear goals and methods.
* **Show and tell**: Present your thoughts in visuals to be easy to grasp for everyone in the room.
* **Define the problem clearly**: Create a well-defined and clear vision for a given challenge, which can help others understand clearly and encourage them to contribute more.
* **Experiment often**: Create a prototype to understand the implementation of the idea in real-life situations. Adopt a fail-fast strategy and experiment more often.
* **Bias for action**: The ultimate design to deliver a solution rather than just thinking. Be proactive in pushing forward and coming up with activities that can result in a workable solution.

Design thinking has a solid foundation to apply empathy and create a holistic view of the given problem. To adopt design thinking, there is a five-phase model proposed by **d.school** (<https://dschool.stanford.edu/resources/getting-started-with-design-thinking>). They are pioneers in teaching and applying design thinking. The following diagram illustrates the five phases of design thinking:

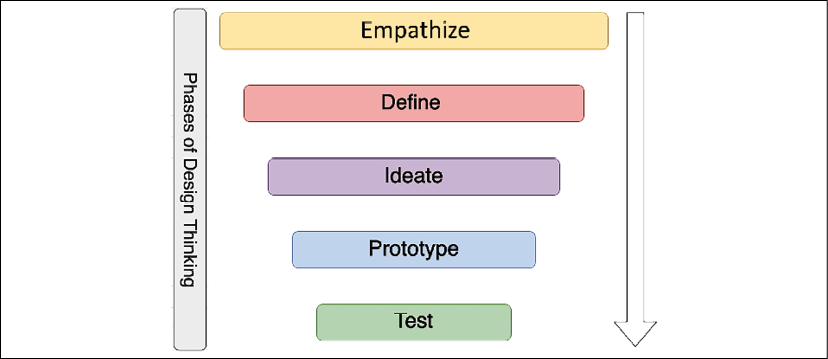


Figure 19.3: Five phases of design thinking

Design thinking is an iterative approach that needs to evolve continuously. The output from one phase can recursively be input to other phases until the solution gets solidified. A brief overview of the phases follows:

* **Empathize**: Empathy is the building block and foundation of design in the human context. To empathize, you should observe your user behaviors and engage with them to understand the actual issue. Try to immerse yourself in—and experience—the problem by putting yourself in the situation.
* **Define**: Empathizing helps define the problem as you experience the user's needs and the problem they face. In the define mode, you apply your insight and define the problem clearly, which can fuel brainstorming to find an innovative yet simple solution.
* **Ideate**: The ideation phase is about moving from problem to solution. You work with the team to find various alternative solutions by challenging assumptions. You need to get an obvious solution out of your head and work collaboratively to find all possible solutions, which allows for innovation.
* **Prototype**: The prototype phase helps to convert ideas into concrete solutions. Prototyping can provide lots of learning and help resolve disagreements by showing a **proof of concept** (**POC**). It helps you to find gaps and risks. You should build a quick prototype without lots of investment, which allows you to handle failure and increase learning.
* **Test**: The test phase is about getting feedback on your solution and reiterating accordingly. The test phase helps you to redefine the solution and learn more about your users.

Design thinking accommodates all the phases required to come up with a logical and practical solution. When designing application architecture, you can relate the phases and principles of design thinking to your real life. There is special stress on prototyping, as that is the only way to solidify your proposal and existing solutions with data and facts. A solution architect's primary job is to understand the business concern and create a technical solution design with a prototype that the team can implement. To build a prototype, the solution architect needs to get their hands dirty and engage in coding hands-on. Let's learn more about it.

**Being a builder by engaging in coding hands-on**

A solution architect is a builder who learns by doing. A prototype is worth a thousand pictures. It helps to reduce miscommunication and ideate solutions. Presenting a POC and prototyping is an integral part of the solution architect's role.

Prototyping is the pre-solution phase, which helps to deepen your understanding of the application design and user. It helps you to think and build multiple solution paths. With the testing of the prototype, you can refine your solution and inspire others, such as teams, customers, and investors, by demoing your vision.

A solution architect is a technical leader who works closely with the development team. In the empowered agile team of developers, a solution architect needs to show a piece of code as a POC, in addition to a PowerPoint presentation. A solution architect doesn't need to be part of the development team, but works collaboratively to convey the solution to the dev team in their language. Successful delivery is only possible if the solution architect can understand the deep technical aspect of a solution that comes with continuous coding, hands-on.

A solution architect is often seen as a mentor and player-coach; having some hands-on coding helps them establish credibility. A solution architect needs to decide which programming languages and tools the team should use. A hands-on approach helps identify gaps that may not fit your team or solution requirements—always learning new technology enables the solution architect to make a better decision on behalf of the organization. Let's learn more about the techniques of continuous learning.

**Becoming better with continuous learning**

Solution architects need to continually absorb new knowledge and enhance their skill set to help the organization make better decisions. Continuous learning keeps your skill set relevant and builds confidence. It opens up your mind and changes prospects. Learning could be challenging with a full-time job and a busy family life. Continuous learning is about developing the habit of always learning something new, whereby you have to be motivated and disciplined. You first need to set up learning goals and apply effective time management to achieve them. This often slips through the net when you get busy with regular daily work.

Everyone has their style of learning. Some people may like formal education; some may read books; others may want to listen to and watch tutorials. You need to find the learning style that is most effective for you and suited to your lifestyle. For example, you can choose to listen to audiobooks and tutorials when commuting to work. You can read books during a business-trip flight or watch video tutorials during exercise hours in the gym. Overall, you need to make some adjustments to put time aside from your busy work life for continuous learning. Here are some of the ways to engage yourself in constant learning:

* **Learning new technologies, frameworks, and languages by trying them out**: Solution architects are the builders and are ready to experiment hands-on. As a successful solution architect, you need to keep learning new technologies by building a small POC. Understanding modern programming languages and frameworks will help you provide the best advice on technology adoption for an organization and team.
* **Learning new skills by reading books and tutorials**: Online learning has brought a revolution and has made it easy to understand and dive deep into any area. You now have massive knowledge bases at your fingertips to learn anything. An online platform such as Udemy or Coursera provides thousands of video tutorial courses in all areas that you can watch online or download to your device for offline learning.

Similarly, there are millions of books available on Kindle to read anytime and anywhere. Audiobook platforms such as Audible and Google Play's audiobooks can help you listen to the book during your commute. There are so many convenient resources available that there is no excuse not to apply continuous learning.

* **Keeping up with technology news and developments by reading articles on websites and blogs**: The best way to keep yourself updated with technology trends is by subscribing to technical news and blogs. TechCrunch.com, Wired.com, and Cnet.com are some of the popular websites to get the latest technology trends. Major newspapers such as *CNBC* or *The New York Times*, and the BBC News and CNN channels have technology articles that provide a good insight into industry trends. You can subscribe to blogs for new learning in the respective technology area. For example, for cloud platform learning, you can subscribe to **Amazon Web Services** (**AWS**) blogs, which have thousands of articles and use cases in the area of the the AWS cloud, and similar blogs are available from other public clouds such as Azure and **Google Cloud Platform** (**GCP**).
* **Writing a blog, whitepaper, or book**: Sharing knowledge is the best way to learn as you think through use cases when trying to present to others. Publishing blogs and articles in popular blog-publishing platforms such as *Medium*, Blogger, and LinkedIn helps you share your learning and learn from others. Active participation in question-and-answer platforms enables you to find an alternative solution for any given problem. Some popular question/answer platforms are Quora, Reddit, Stack Overflow, and Stack Exchange.
* **Solidify your knowledge by teaching others**: Teaching others helps you collaborate and get a different perspective of your knowledge. Often, use cases proposed by participants give you different ways of finding a solution. Running a full-day workshop with a hands-on lab and concept building helps you solidify your learning and learn with others.
* **Taking online classes**: Sometimes, you want to go for formal learning to be more disciplined, and you want to be flexible. Online courses provide flexibility and help you to adjust to other priorities and save time. Online courses can offer you an organized way to learn new technologies and help to enhance knowledge.
* **Learning from teammates**: Teammates share the same working environment, and you spend most of the day with them. Learning with team members can help to speed up your learning. The team can adopt a divide-and-conquer strategy whereby each team member can share their topics and present deep-dive lunch-and-learn sessions. These sessions are a standard method used by many organizations to conduct regular learning sessions among team members. Each team member shares their new learning in a weekly learning session, and everyone quickly learns new topics.
* **Attending and participating in user groups and conferences**: All large vertical industry and technology organizations conduct conferences to provide insight into new technology trends and hands-on sessions. Participating in industry conferences and user group meetings helps to develop networking and understand technology trends. Some of the large technology conferences from industry leaders include AWS re:Invent, Google Cloud Next, Microsoft Ignite, SAP SAPPHIRE, and Strata Data Conference. You can create a local user group and conduct a meetup in your local area, which will help you to collaborate with professionals across industries and organizations.

A solution architect plays a technical leadership role, and good leadership warrants preparing more leaders like you, which is possible through mentorship. Solution architects should play a player-coach role and mentor others. Let's look at this in more detail.

**Being a mentor to others**

Mentoring is about helping others and setting them up for success based on your learning and experience. It is an effective way to develop leaders by having one-to-one mentor/mentee relationships. To be a good mentor, you need to establish an informal communication style where the mentee can develop a comfort zone. The mentee can seek advice in multiple areas such as career development, or personal aspects such as work-life balance. You should do an informal needs assessment and set up mutual goals and expectations.

Mentorship is more about listening. Sometimes, people need someone to listen to them and advise as required. You should listen carefully first and understand their point of view.

Help them to make their own decisions as this will make them feel more accomplished. As a good mentor, when advising for a career, you need to be open to advise what the best fit for the mentee is, even if it may not necessarily be the best fit for the company. Always provide honest, constructive feedback to help them identify gaps and overcome them.

The critical trait of a mentor is the ability to inspire people. Often, people may choose you as a mentor if they see a role model in you. Help your mentee realize their full potential without putting your view forward, and help them achieve what they never thought of earlier. There are always mutual benefits to being a mentor; you also learn from mentees about people's behavior and growth. Being a mentor to others will ultimately help you to become a better leader and person.

**Becoming a technology evangelist and thought leader**

Technology evangelism is about being an expert to advocate technology and your product. Some organizations with an extensive product base roll out a separate technology evangelist role. Still, often, a solution architect needs to assume the role of an evangelist as part of their job. As a technology evangelist, you need to be aware of current technology trends to understand real-world problems and advocate your technology to solve their business concerns.

Technology evangelism involves participating in an industry conference as a public speaker and promoting your respective platform. It allows you to become a thought leader and an influencer, which can help the organization increase its platform and product adoption. Public speaking is one of the critical skills required for a solution architect to interreact on various public platforms and present in front of a large audience.

An evangelist also creates and publishes content such as blog posts, whitepapers, and microblogs to advocate their product. They socialize the content to increase adoption and interact with the user to understand their feedback. An evangelist works backward from the customer and communicates feedback to the internal team to help to make the product better. With time, as an evangelist, you will refine the message that works in the organization's best interests.

Overall, a solution architect is a role with multiple responsibilities, and taking more ownership will help you to better succeed in your career.

**Summary**

In this chapter, you learned about the various soft skills required for a solution architect to be successful. A solution architect needs to have pre-sales skills such as negotiation, communication, problem solving, and listening, which help them support the organization's pre-sales cycle, such as with the RFP. You learned about the presentation skills required for executive conversations and buy-in.

You learned about the strategic understanding that a solution architect should define key objectives and results for an organization. To execute at various levels, solution architects should have the ability to think big and be flexible and adaptable. You learned details about solution architects taking ownership and being accountable for their actions.

A solution architect's role has the primary responsibility of architecture design. You learned about design thinking, with its principles and phases. You also learned about the importance of continuous learning and different techniques to carry on learning and keep yourself up to date with market trends. You also learned about the additional responsibilities of the solution architect—to work as a mentor and evangelist.

It has been a long journey through this book learning all about solution architects, from their roles and responsibilities to different aspects of solution design and architecture optimization. I hope you have learned a lot and that it will help you develop your career as a solution architect or help you succeed in your current role.

Happy learning!