**2**

**Collecting Business Requirements**

This chapter discusses how enterprises can accelerate business results by implementing a multi-cloud strategy. Typically, this is a task for enterprise or business architects, but in digital transformation, enterprise architecture and cloud architecture are tightly connected. We have to collect business requirements as a first step before we can think of the actual cloud strategy.

Every cloud platform/technology has its own benefits and by analyzing business strategies and defining what cloud technology fits best, enterprises can really take advantage of multi-cloud. A strategy should not be “cloud-first” but “cloud-fit.” But before we get into the technical strategy and the actual cloud planning, we must explore the business or enterprise strategy and the financial aspects that drive this strategy.

In this chapter, we’re going to cover the following main topics:

* Analyzing the enterprise strategy for the cloud
* Defining the cloud strategy from the enterprise architecture
* Fitting cloud technology to business requirements
* Applying the value streams of IT4IT
* Keeping track of cloud developments—focusing on the business strategy
* Creating a comprehensive business roadmap
* Mapping the business roadmap to a cloud-fit strategy

**Analyzing the enterprise strategy for the cloud**

Before we get into a cloud strategy, we need to understand what an enterprise strategy is and how businesses define such a strategy. As we learned in the previous chapter, every business should have the goal of generating revenue and earning money. That’s not really a strategy. The strategy is defined by how it generates money with the products the business makes or the services that it delivers.

A good strategy comprises a well-thought-out balance between timing, access to and use of data, and something that has to do with braveness—daring to make decisions at a certain point in time. That decision has to be based on—you guessed it—proper timing, planning, and the right interpretation of data that you have access to. If a business does this well, it will be able to accelerate growth and, indeed, increase revenue. The overall strategy should be translated into use cases. Use cases can be:

* Delivering products in new business models, such as SaaS
* Achieving more resilience in business, for instance, by implementing disaster recovery using the cloud
* Faster time to market in product development through the quick deployment of development environments
* Analysis of big data using data lakes in the cloud

These use cases must be reflected by the strategy of the enterprise. They will drive the decisions in designing and implementing cloud solutions and even in the choice of cloud platforms.

The success of a business is obviously not only measured in terms of revenue. There are a lot of parameters that define success as a whole and for that matter, these are not limited to just financial indicators. Nowadays, companies rightfully also have social indicators to report on. Think of sustainability and social return. However, a company that does not earn money, one way or the other, will likely not last long.

What are the drivers for business strategy? Typically, these are categorized into four areas:

* Financial objectives
* Customer objectives
* Product objectives
* Internal objectives

In the first chapter, the customer requirements were discussed alongside the methodologies to capture these, the use of **Quality Function Deployment**(**QFD**),**the House of Quality**(**HOQ**),**and the Voice of the Customer**(**VOC**). By understanding the customer needs, the enterprise is able to design, develop, and deploy products that customers are willing to buy—if the price is right. So, there’s another challenge for the enterprise: it needs to deliver products at a price that is acceptable to customers. The price needs to cover the costs and preferably with enough margin to make some profit. However, there’s one more aspect that is crucial to enterprise strategy: timing.

Time is one of the most important factors to consider when planning for business acceleration. Having said that, it’s also one of the most difficult things to grasp. No one plans for a virus outbreak and yet it happened in 2020, leading to a worldwide pandemic. It was a reason for businesses not to push for growth at that time. The strategy for a lot of companies probably changed from pushing for growth to staying in business by trying to drive costs down. It proved that modern businesses have to be extremely agile.

Business agility has become one of the most important strategic drivers to go to the cloud, but what is business agility? It’s the capability of a business to respond quickly to events in rapidly changing markets. This comes with a different enterprise architecture. The enterprise has to become adaptive in every layer. That means that the organization has to change: smaller, agile working teams, interacting closely with customers and focusing on smaller tasks that can be executed in sprints of a couple of weeks.

But it also means that systems must become agile. Changing customer demands must be met quickly, resulting in new features in systems. A lot of enterprises still have a massive technical debt, with big, monolithic systems that are hard to update and upgrade without the need to change the entire system. Modern architecture is about microservices: applications are compiled as a collection of services that are loosely coupled. Teams will work on specific services, independently from other teams. The services will interact with each other through protocols such as HTTP, TCP, and AMQP.

**Shifting to a subscription-based economy**

In the past decade, the markets for most businesses have changed dramatically. Markets have shifted from privately owned to platform economies and eventually a subscription-based economy. Products and services are used as a subscription, “as a service.” Consumers can have services on any device, at any time, at any place they desire. They simply pay a fee to get access to a service; they don’t want to own the product or service.

The challenge for businesses is that services are now consumed in a completely different manner. It’s not a one-off sale anymore, and customers can actually do something with the subscription: it can be paused, changed, suspended, restated, or stopped completely. Subscriptions are very fluid. Hence, enterprises must have architectures that are capable of addressing this flexibility, and systems must be agile and scalable.

Another aspect crucial to business strategy and agility is access to and the use of data. It looks like the use of data in business is something completely new, but of course, nothing could be further from the truth. Every business in every era can only exist through the use of data. We might not always consider something to be data since it isn’t always easy to identify, especially when it’s not stored in a central place.

Data is the key. It’s not only about raw data that a business (can) have access to, but also about analyzing that data. Where are my clients? What are their demands? Under what circumstances are these demands valid and what makes these demands change? How can I respond to these changes? How much time would I have to fulfill the initial demands and apply these changes if required? This all comes from data. Nowadays, we have a lot of data sources. The big trick is how we can make these sources available to a business—in a secure way, with respect to confidentiality, privacy, and other compliance regulations.

An example will make this clearer. The changes in the global healthcare market make an excellent example of business challenges, due to changing market circumstances such as a lack of skilled staff and a globally aging population, requiring more cure and care. The sustainability of the global health system is under high pressure. Hence, there’s a general understanding that there should be more focus on prevention, rather than on treatments. Treatment costs society a lot more than preventing people from getting sick in the first place. Governments and companies are therefore now trying to make sure that people start improving their lifestyles. Data is absolutely essential to start this improvement and then to develop services that will help people develop and maintain a better lifestyle.

Companies are investing in collecting health data. And no surprise, it’s big tech that’s heading the game, since the place to collect data is the cloud. It’s collected from medical devices and also from devices such as smartwatches and equipment in gyms connected to the internet. This data is analyzed so that trends can be made visible. Based on that data, individual health plans can be organized, but the data can also be utilized for commercial goals: selling running shoes, diet products, or health services, preferably on a subscription basis so that services can easily be adapted when customer demands change.

Challenges in this healthcare use case are primarily confidentiality and protection of data, and compliance with international privacy regulations. These are bigger challenges than the technology itself. The technology to collect data from various sources and make it available through data mining and analytics is generally available.

In the following sections, we will address business challenges such as business agility, security, data protection, and time to market and understand why cloud technology can help enterprises in dealing with these challenges.

**Considering cloud adoption from enterprise architecture**

In the previous section, the changes in the enterprise markets were discussed. The modern economy is changing from ownership to subscription-based models, leading to the need for flexibility, adaptability, agility, and scalability in all layers of the enterprise, including the organization and the systems. Subscriptions come with new payment models such as **pay-as-you-go**(**PAYG**) and freemium concepts, where a basic service is delivered at no charge, but customers can enhance service with paid options. Services must be interoperable, but also capable of interacting with other systems such as payment services.

The overarching enterprise architecture as a result of this digital transformation will inevitably change. The architecture will enable faster development, targeting microservices developed by smaller teams and using cloud-native technology. In the next sections, you will learn what an architect should take into account in defining this new digital, cloud-native strategy. This is typically the work of an enterprise architect but, as we already mentioned in the introduction to this chapter, with digital transformation, enterprise and strategic cloud architecture are getting closely related to each other. The enterprise architect must understand the cloud, and the cloud architect should have knowledge about enterprise architecture since cloud adoption is not purely a technological subject.

**Long-term planning**

When a business is clear on its position and its core competencies, it needs to set out a plan. Where does the company want to be in 5 years from now? This is the most difficult part. Based on data and data analytics, it has to determine how the market will develop and how the company can anticipate change. Again, data is absolutely key, but so is the swiftness with which companies can change course since market demands do change extremely rapidly.

**Financial structure**

A business needs a clear financial structure. How is the company financed and how are costs related to different business domains, company divisions, and its assets? As we will find out as part of financial operations, the cloud can be of great help in creating fine-grained insight into financial flows throughout a business. With correct and consistent naming and tagging, you can precisely pinpoint how much cost a business generates in terms of IT consumption. The best part of cloud models is that the foundation of cloud computing is *paying for what you use*. Cloud systems can *breathe* at the same frequency as the business itself. When business increases, IT consumption can increase. When business drops, cloud systems can be scaled down, and with that, they generate lower costs, whereas traditional IT is way more static.

So, nothing is holding us back from getting our business into the cloud. However, it does take quite some preparation in terms of (enterprise) architecture. In the following section, we will explore this further.

**Fitting cloud technology to business requirements**

We are moving business into the cloud because of the required agility and, of course, to control our costs.

The next two questions will be: with what and how? Before we explore the how and basically the roadmap, we will discuss the first question: with what? A cloud adoption plan starts with business planning, which covers business processes, operations, finance, and, lastly, the technical requirements. We will have to evaluate the business demands and the IT fulfillment of these requirements.

When outsourcing contracts, the company that takes over the services performs so-called due diligence. As per its definition, due diligence is *“a comprehensive appraisal of a business undertaken by a prospective buyer, especially to establish its assets and liabilities and evaluate its commercial potential”* (source: <https://www.lexico.com/en/definition/due_diligence>). This may sound way too heavy of a process to get a cloud migration asset started, yet it is strongly recommended as a business planning methodology. Just replace the words *prospective buyer* with the words *cloud provider* and you’ll immediately get the idea behind this.

**Business planning**

One really important step in the discovery phase that’s crucial to creating a good mapping to cloud services is evaluating the service levels and performance indicators of applications and IT systems. Service levels and **key performance indicators** (**KPIs**) will be applicable to applications and the underlying IT infrastructure, based on specific business requirements.

Think of indicators like metrics of availability, durability, and levels of backup, including RTO/RPO specifications, requirements for **business continuity** (**BC**), and **disaster recovery** (**DR**). Are systems monitored 24/7 and what are the support windows? These all need to be considered. As we will find out, service levels and derived service-level agreements might be completely different in cloud deployments, especially when looking at PaaS and SaaS, where the responsibility of platform (PaaS) and even application (SaaS) management is largely transferred to the solution provider.

If you ask a CFO what the core system is, they will probably answer that financial reporting is absolutely critical. It’s the role of the enterprise architect to challenge that. If the company is a meat factory, then the financial reporting system is not the most critical system. The company doesn’t come to a halt when financial reporting can’t be executed. The company does, however, come to a halt when the meat processing systems stop; that immediately impacts the business. What would that mean in planning the migration to cloud systems? Can these processing applications be hosted from cloud systems? And if so, how? Or maybe, when?

In business architecture, the architect defines the purpose, vital functions, critical processes, and interactions between various business components. It describes:

* Business processes.
* Products and services and their respective taxonomies.
* Business capabilities.
* Business architecture defines how the enterprise functions in order to deliver products and services. It also defines what data it needs to be able to function and, lastly, what systems are required.

Business planning involves the following items:

* Discovery of the entire IT landscape, including applications, servers, network connectivity, storage, APIs, and services from third-party providers.
* Mapping of IT landscape components to business-critical or business-important services.
* Identification of commodity services and shared services and components in the IT landscape.
* Evaluation of IT support processes with regard to commodity services, and critical and important business services. This includes the levels of automation in the delivery of these services.

These are the first topics to discuss when initiating a digital transformation that typically involves cloud migration or cloud development plans; from the enterprise strategy, it should be clear what the core competence and, therefore, the core business process is, supported by the company’s core systems.

**Financial planning**

After the business planning phase, we also need to perform financial analyses. After all, one of the main rationales for moving to cloud platforms is cost control. Be aware: moving to the cloud is not always a matter of lowering costs. It’s about making your costs *responsive* to actual business activity. Setting up a business case to decide whether cloud solutions are an option from a financial perspective is, therefore, not an easy task. Public cloud platforms offer **Total Cost of Ownership** (**TCO**) calculators.

TCO is indeed the total cost of owning a platform and it should include all direct and indirect costs. What do we mean by that? When calculating the TCO, we have to include the costs that are directly related to systems that we run: costs for storage, network components, compute, licenses for software, and so on. But we also need to consider the costs of the labor that is involved in managing systems for engineers, service managers, or even the accountant that evaluates the costs related to the systems. These are all costs; however, these indirect costs are often not taken into account in the full scope. Especially in the cloud, these should be taken into account. Think of this: what costs can be avoided by, for example, automating service management and financial reporting?

So, there’s a lot to cover when evaluating costs and the financial planning driving architecture. Think of the following:

* All direct costs related to IT infrastructure and applications. This also includes hosting and housing costs—for example, (the rental of) floor space and power.
* Costs associated with all staff working on IT infrastructure and applications. This includes contractors and staff from third-party vendors working on these systems.
* All licenses and costs associated with a vendor or third-party support for systems.
* Ideally, these costs can be allocated to a specific business process, division, or even user group so that it’s evident where IT operations costs come from.

Why is this all important in drafting architecture? A key financial driver to start a cloud journey is the shift from CapEx to OpEx. In essence, **CapEx**—**capital expenditure**—concerns upfront investments—for example, buying physical machines or software licenses. These are often one-off investments, of which the value is depreciated over an economic life cycle. **OpEx**—**operational expenditure**—is all about costs related to day-to-day operations and for that reason is much more granular. Usually, OpEx is divided into smaller budgets, which teams need to have to perform their daily tasks. In most cloud deployments, the client really only pays for what they’re using. If resources sit idle, they can be shut down and costs will stop. A single developer could—if mandated for this—decide to spin up an extra resource if required.

That’s true for a **PAYG** deployment, but we will discover that a lot of enterprises have environments for which it’s not feasible to run in full PAYG mode. You simply don’t shut down instances of large, critical ERP systems. So, for these systems, businesses will probably use more stateful resources, such as reserved instances that are fixed for a longer period. For cloud providers, this means a steady source of income for a longer time, and therefore, they offer reserved instances against lower tariffs or to apply discounts. The downside is that companies can be obliged to pay for these reserved resources upfront. Indeed, that’s CapEx. To cut a long story short, the cloud is not OpEx by default.

**Understanding the cost of delay**

We have our foundation or reference architecture set out, but now, our business gets confronted with new technologies that have been evaluated: what will they bring to the business? As we mentioned previously, there’s no point in adopting every single new piece of tech that is released. The magic words here are *business case*.

A business case determines whether the consumption of resources supports a specific business need. A simple example is as follows: a business consumes a certain bandwidth on the internet. It can upgrade the bandwidth, but that will take an investment. That is an out-of-pocket cost, meaning that the company will have to pay for that extra bandwidth. However, it may help workers to get their job done much faster. If workers can pick up more tasks just by the mere fact that the company invests in a faster internet connection, the business case will, in the end, be positive, despite the investment.

If market demands occur and businesses do not adapt to this fast enough, a company might lose part of the market share and thereby lose revenue. Adopting new technology or speeding up the development of applications to cater to changing demands will lead to costs. These costs correspond to missing specific timing and getting services or products to the market in time. This is what we call the cost of delay.

Cost of delay, as a piece of terminology, was introduced by Donald Reinertsen in his book *The Principles of Product Development Flow*, published by *Celeritas Publishing*, *2009*:

*We need cost of delay to evaluate the cost of queues, the value of excess capacity, the benefit of smaller batch sizes, and the value of variability reduction. Cost of Delay is the golden key that unlocks many doors. It has an astonishing power to totally transform the mindset of a development organization.*

Although it’s mainly used as a financial parameter, it’s clear that the cost of delay can be a good driver to evaluate the business case for adopting cloud technology. Using and adopting consumption of cloud resources that are more or less agile by default can mitigate the financial risk of cost of delay.

**Moving to the benefit of opportunity**

If there’s something that we could call the cost of delay, there should also be something that we could call the benefit of opportunity. Where the cost of delay is the risk of missing momentum because changes have not been adopted timely enough, the benefit of opportunity is really about accelerating the business by exploring future developments and related technology. It can be very broad. As an example, let’s say a retailer is moving into banking by offering banking services using the same app that customers use to order goods. Alternatively, think of a car manufacturer, such as Tesla, moving into the insurance business.

The accessibility and even the ease of using cloud services enable these shifts. In marketing terms, this is often referred to as blurring. In the traditional world, that same retailer would really have had much more trouble offering banking services to its customers, but with the launch of SaaS in financial apps, from a technological point of view, it’s not that hard to integrate this into other apps. Of course, this is not considering things such as the requirement of a banking license from a central financial governing institution and having to adhere to different financial compliance frameworks. The simple message is that with cloud technologies, it has become easier for businesses to explore other domains and enable a fast entrance into them, from a purely technological perspective.

The best example? AWS. Don’t forget that Amazon was originally an online bookstore. Because of the robustness of its ordering and delivery platform, Amazon figured out that it also could offer storage systems “for rent” to other parties. After all, they had the infrastructure, so why not fully capitalize on that? Hence, S3 storage was launched as the first AWS cloud service and by all means, it got AWS to become a leading cloud provider, next to the core business of retailing. That was truly a benefit of opportunity.

**Technical planning**

Finally, we’ve reached technical planning, which starts with foundation architecture. You can plan to build an extra room on top of your house, but you’ll need to build the house first—assuming that it doesn’t exist. The house needs a firm foundation that can hold it in the first place but also *carry* the extra room in the future. The room needs to be integrated with the house since it would really be completely useless to have it *stand alone* from the rest of the house. It all takes good planning. Good planning requires information—data, if you like.

In the world of multi-cloud, you don’t have to figure it out all by yourself. The major cloud providers all have reference architectures, best practices, and use cases that will help you plan and build the foundation, making sure that you can fit in new components and solutions almost all the time. We will discuss this in detail in *Chapter 3*, *Starting the Multi-Cloud Journey*.

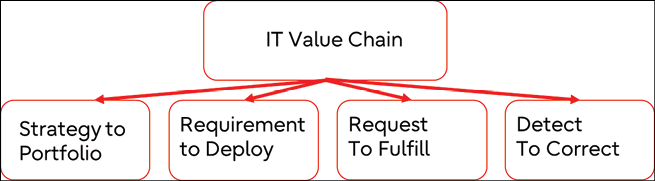
That’s exactly what we are going to do in this book: plan, design, and manage a future-proof multi-cloud environment. In the next section, we will take our first look at the foundation architecture.

**Applying the value streams of IT4IT**

The problem that many organizations face is controlling the architecture of businesses at large. IT4IT is a framework that helps organizations with that. It’s complementary to TOGAF and, for this reason, is also issued as a standard by The Open Group. IT4IT is also complementary to **IT Infrastructure Library**(**ITIL**), where ITIL provides best practices for IT service management. IT4IT provides the foundation to enable IT service management processes with ITIL. It is meant to align and manage a digital enterprise. It deals with the challenges that these enterprises have, such as the ever-speeding push for embracing and adopting new technology. The base concept of IT4IT consists of four value streams:

* **Strategy to portfolio**: The portfolio contains technology standards, plans, and policies. It deals with the IT demands of the business and maps these demands to IT delivery. An important aspect of the portfolio is project management to align business and IT.
* **Requirements to deploy**: This stream focuses on creating and implementing new services or adapting existing services, in order to reach a higher standard of quality or to obtain a lower cost level. According to the documentation of The Open Group, this is complementary to methods such as Agile Scrum and DevOps.
* **Request to fulfill**: To put it very simply, this value stream is all about making life easy for the end customers of the deployed services. As companies and their IT departments adopt structures such as IaaS, PaaS, and SaaS, this stream enables service brokering by offering and managing a catalog and, with that, speeds up the fulfillment of new requests made by end users.
* **Detect to correct**: Services will change. This stream enables monitoring, management, remediation, and other operational aspects that drive these changes.

The following diagram shows the four streams of IT4IT:



*Figure 2.1: IT4IT value streams (The Open Group)*

Frameworks such as IT4IT are very valuable in successfully executing this mapping. In the next section, we will focus on how cloud technology fits into the business strategy.

With that, we have not only defined a strategy; we have also learned how to apply the value streams of IT4IT, and how to realistically keep track of the ever-increasing cloud developments. We know where we’re coming from, and we know where we’re heading to. Now, let’s bring everything together and make it more tangible by creating the business roadmap and finally mapping that roadmap to our cloud strategy, thereby evaluating the different deployment models and cloud development stages.

**Keeping track of cloud developments—focusing on the business strategy**

Any cloud architect or engineer will tell you that it’s hard to keep up with developments. Just for reference, AWS and Azure issued over 2,000 features in their respective cloud platforms in just over 1 year. These can be big releases or just some minor tweaks.

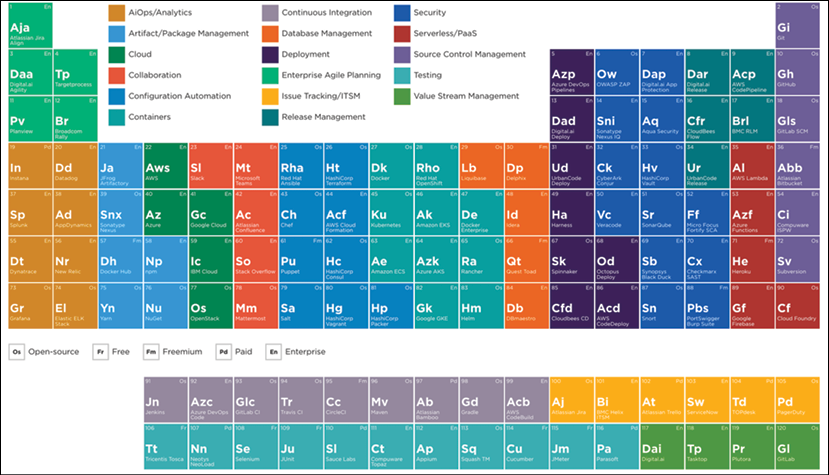
The major clouds Azure, AWS, Google Cloud Platform, and Alibaba release thousands of new features every year (refer to <https://www.gartner.com/doc/reprints?id=1-2AOZQAQL&ct=220728&st=sb>). These can be simple additions to existing services such as virtual machines or storage, but also new services with pre-packaged code that enable the fast deployment of web services.

But there’s much more. Think of the possibilities of **Virtual Reality** (**VR**), **Augmented Reality** (**AR**), and **Artificial Intelligence** (**AI**) from cloud platforms or building digital twins in the cloud. Cloud providers already also offer quantum simulation and it’s expected that this will grow significantly in the near future. However, there are innovations that even go beyond this. Think of blockchain, Web 3.0, and the Metaverse.

Web 3.0 is based on blockchain technology and promises to hand back control over data to the user. The other promise is the Metaverse: a 3D version of the internet, something that is still hard to really understand since definitions of the Metaverse differ quite a lot. In most definitions, the Metaverse is presented as a sort of parallel universe, where we have digital twins moving in that digital world. Predictions by Bloomberg are that by 2024, the Metaverse will hold over 80 billion USD worth of business (refer to <https://www.bloomberg.com/professional/blog/metaverses-80-billion-etf-assets-by-2024-virtually-a-reality/>). Google, Microsoft, Meta, and Apple are heavily investing in the technology.

It is a constant stream of innovations: big, medium, and small. We have to keep one thing in mind: whether it’s AR, VR, blockchain, or the Metaverse, somewhere there’s technology involved at a low level. What do we mean by that? Someone has to develop, program, test, and deploy the technology. Anyway, trying to keep up with all these innovations, releases, and new features is hard, if not impossible.

And it gets worse. It’s not only the target cloud platforms but also a lot of tools that we need to execute, in order to migrate or develop applications and prepare our businesses for the next big thing. Just have a look at the Periodic Table of DevOps Tools, which is currently managed by Digital.ai and is continuously refreshed with new technology and tools. It’s just a matter of time before blockchain technology, Web 3.0, and Metaverse toolkits are added to this table.



*Figure 2.2: Periodic Table of DevOps Tools by Digital.ai*

An interactive version of the Periodic Table of DevOps Tools can be found at <https://digital.ai/devops-tools-periodic-table>. For a full overview of the cloud-native landscape, refer to the following web page, which contains the *Cloud Native Trail Map of the Cloud Native Computing Foundation*: <https://landscape.cncf.io/>.

There’s no way to keep up with all the developments. A business needs to have focus, and that should come from the strategy that we discussed in the previous sections. In other words, don’t get carried away with all the new technology that is being launched.

There are three important aspects that have to be considered when managing cloud architecture: the foundation architecture, the cost of delay, and the benefit of opportunity.

**Creating a comprehensive business roadmap**

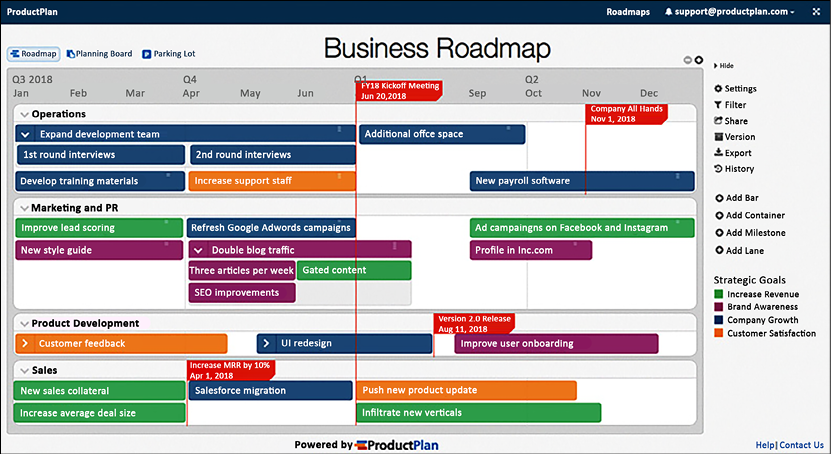
There are stores filled with books on how to create business strategies and roadmaps. This book absolutely doesn’t have any pretensions of condensing this all into just one paragraph. However, for an enterprise architect, it is important to understand how the business roadmap is evaluated:

* The mission and vision of the business, including the strategic planning of how the business will target the market and deliver its goods or services.
* Objectives, goals, and direction. Again, this includes planning, in which the business sets out when and how specific goals are met and what it will take to meet the objectives in terms of resources.
* **Strengths, weaknesses, opportunities, and threats** (**SWOT**). The SWOT analysis shows whether the business is doing the right things at the right time or that a change in terms of the strategy is required.
* Operational excellence. Every business has to review how it is performing on a regular basis. This is done through KPI measurements: is the delivery on time? Are the customers happy (**customer satisfaction**—**CSAT**)?

Drivers for a business roadmap can be very diverse, but the most common ones are as follows:

* Revenue
* Gross margin
* Sales volume
* Number of leads
* Time to market
* Customer satisfaction
* Brand recognition
* Return on investment

These are shared goals, meaning that every division or department should adhere to these goals and have their planning aligned with the business objectives. These goals end up on the business roadmap. These can be complex in the case of big enterprises but also rather straightforward, as shown in the following screenshot:



*Figure 2.3: Template for a business roadmap (by ProductPlan)*

IT is the engine that drives everything: development, resource planning, **Customer Relationship Management** (**CRM**) systems, websites for marketing, and customer apps. And these days, the demands get more challenging; the life cycle is getting shorter and the speed is getting faster. Where IT was the bottleneck for ages, it now has all the technology available to facilitate the business in every aspect. IT is no longer considered a cost center but a business enabler.

**Mapping the business roadmap to a cloud-fit strategy**

Most businesses start their cloud migrations from traditional IT environments, although a growing number of enterprises are already quite far into cloud-native development too. We don’t have to exclude either one; we can plan to migrate our traditional IT to the cloud, while already developing cloud-native applications in the cloud itself. Businesses can have separate cloud tracks, running at different speeds. It makes sense to execute the development of new applications in cloud environments using cloud-native tools. Next, the company can also plan to migrate its traditional systems to a cloud platform. There are a number of ways to do that. We will be exploring these, but also look at drivers that start these migrations. The key message is that it’s likely that we will not be working with one roadmap. Well, it might be one roadmap, but one that is comprised of several tracks with different levels of complexity and different approaches, at different speeds.

There has been a good reason for discussing enterprise strategy, business requirements, and even financial planning. The composition of the roadmap with these different tracks is fully dependent on the outcome of our assessments and planning. And that is architecture too—let there be no mistake about that.

Here, we’re quoting technology leader Radhesh Balakrishnan of Red Hat:

*A multi-cloud strategy allows an organization to meet specific workload or application requirements—both technically and commercially—by consuming cloud services from several cloud providers.*

He adds the following:

*Not every department, team, business function, or application or workload will have similar requirements in terms of performance, privacy, security, or geographic reach for their cloud. Being able to use multiple cloud providers that meet their various application and data needs is critical as cloud computing has become more mature and mainstream.*

These business requirements will drive the cloud migration approach. We recognize the following technological strategies:

* **Rehost**: The application, data, and server are migrated as is to the target cloud platform. This is also referred to as **lift and shift**. The benefits are often quite low. This way, we’re not taking any advantage of cloud-native services.
* **Replatform**: The application and data are migrated to a different target technology platform, but the application architecture remains as is. For example, let’s say an application with a SQL database is moved to PaaS in Azure with Azure SQL Server. The architecture of the application itself is not changed.
* **Repurchase**: In this scenario, an existing application is replaced by SaaS functionality. Note that we are not really repurchasing the same application. We are replacing it with a different type of solution.
* **Refactor**: Internal redesign and optimization of the existing application. This can also be a partial refactoring where only parts of an application are modified to operate in an optimized way in the cloud. In the case of full refactoring, the whole application is modified for optimization in terms of performance and at lower costs. Refactoring is, however, a complicated process. Refactoring usually targets PaaS and SaaS.
* **Rearchitect**: This is one step further than refactoring and is where the architecture of an application, as such, is modified. This strategy does comprise an architectural redesign to leverage multi-cloud target environments.
* **Rebuild**: In this strategy, developers build a new cloud-native application from scratch, leveraging the latest tools and frameworks.
* **Retire**: This strategy is valid if an application is not strategically required for the business going forward. When an application is retired, data needs to be cleaned up and often archived, before the application and underlying infrastructure are decommissioned. Of course, an application can be retired as a follow-up strategy when the functionality in an application is refactored, rearchitected, rebuilt, or repurchased.
* **Retain**: Nothing changes. The existing applications remain on their current platform and are managed as-is they are.

It’s not easy to predict the outcomes of the business case as many parameters can play a significant role. Benchmarks conducted by institutes such as Gartner point out that total cost savings will vary between 20 and 40 percent in the case of rehosting, replatforming, and repurchasing. Savings may be higher when an application is completely rearchitected and rebuilt. Retiring an application will lead to the highest savings, but then again, assuming that the functionality provided by that application is still important for a business, it will need to fulfill that functionality in another way—generating costs for purchasing, implementation, adoption, and/or development.

This all has to be taken into account when drafting the business case and choosing the right strategy.

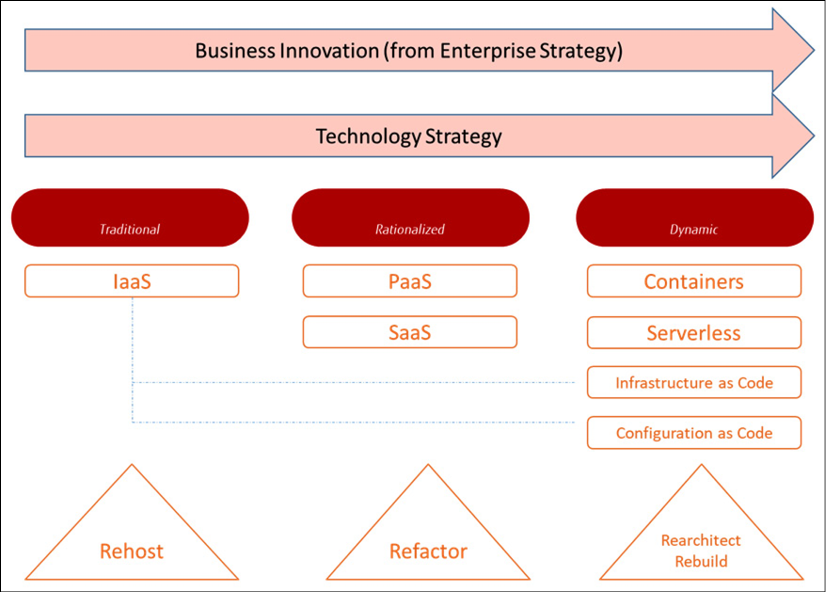
On top of that, multi-cloud will bring its own set of challenges. There might be good reasons to have a data lake in one cloud and have office applications hosted in another cloud. Rationales for this might be specific capabilities that cloud providers have developed in various domains. Think of the Microsoft 365 suite and the capabilities in data and AI that Google has developed. There’s a major business benefit in a strategy that is initiated by the best of breed: choosing the best solution and best provider for specific business use cases.

Another reason might be that developers have a preference to work with certain tools in a specific cloud, or that applications must be hosted in different clouds for compliance reasons. If countries or regions have restrictions on using certain cloud providers or a provider isn’t available in a region, then companies are forced to use multiple providers. Think of companies that can use AWS and Azure in Europe but have to use Alibaba Cloud or Tencent in the Chinese region. Companies would still like to offer the same type of services, regardless of the cloud provider, but it will bring extra challenges in portability and the integration of services.

At a high level, we can plot these strategies into three stages:

* **Traditional**: Although organizations will allow developers to work with cloud-native tools directly on cloud platforms, most of them will still have traditional IT. When migrating to the cloud, they can opt for three scenarios:
  + Lift and shift systems “as is” to the cloud and start the application’s modernization in the cloud.
  + Modernize the applications before migrating them to the target cloud platform.
  + A third scenario would be a mix between the first two.
* **Rationalized**: This stage is all about modernizing the applications and optimizing them for usage in the target cloud platform. This is the stage where PaaS and SaaS are included to benefit from cloud-native technology.
* **Dynamic**: This is the final stage where applications are fully cloud-native and therefore completely *dynamic*; they’re managed through agile workstreams using **continuous improvement** and **continuous development** (**CI**/**CD**), fully scalable using containers and serverless solutions, and fully automated using the principle of everything as code, making IT as agile as the business requires.

It all comes together in the following model. This model suggests that the three stages are sequential, but as we have already explained, this doesn’t have to be the case:



*Figure 2.4: Technology strategy following business innovation*

This model shows three trends that will dominate the cloud strategy in the forthcoming years:

* **Software to services**: Businesses do not have to invest in software anymore, nor in infrastructure to host that software. Instead, they use software that is fully managed by external providers. The idea is that businesses can now focus on fulfilling business requirements using this software, instead of having to worry about the implementation details and hosting and managing the software itself. It’s based on the economic theory of endogenous growth, which states that economic growth can be achieved through developing new technologies and improvements in production efficiency. Using PaaS and SaaS, this efficiency can be sped up significantly.
* **VM to container**: Virtualization brought a lot of efficiency to data centers. Yet, containers are even more efficient in utilizing compute power and storage. VMs still use a lot of system resources with a guest operating system, whereas containers utilize the host operating system and only some supporting libraries on top of that. Containers tend to be more flexible and, due to that, have become increasingly popular for distributing software in a very efficient way. Even large software providers such as SAP already distribute and deploy components using containers. SAP Commerce is supported using Docker containers, running instances of Docker images. These images are built from special file structures mirroring the structures of the components of the SAP Commerce setup.
* **Serverless computing**: Serverless is about writing and deploying code without worrying about the underlying infrastructure. Developers only pay for what they use, such as processing power or storage. It usually works with triggers and events: an application registers an event (a request) and triggers an action in the backend of that application—for instance, retrieving a certain file. Public cloud platforms offer different serverless solutions: Azure Functions, AWS Lambda, and Google Knative. Serverless offers the maximum scalability with the greatest cost control. One remark has to be made at this point: although serverless concepts will become more and more important, it will not be technically possible to fit everything into serverless propositions.

In the next chapter, we will get into the more technical details, learning about the transformation from monolithic architectures into microservices, and the benefits of containers and serverless concepts, while keeping the infrastructure across systems consistent and easy to manage.

**Summary**

In this chapter, we explored the methodologies that are used to analyze enterprise or business strategies and mapped these to a cloud technology roadmap. We also learned that it is close to impossible to keep track of all the new releases and features that are launched by cloud and technology providers. We need to determine what our business goals and objectives are and define a clear architecture that is as future-proof as possible, yet agile enough to adopt new features if the business demands this.

Business agility must be the focus of a modern, digital enterprise. Business drivers are derived from financial, customer, product, and internal objectives, but these are rapidly changing in the current markets. One of the major trends is the upcoming subscription-based economy that forces businesses to create agile organizations, and systems that are able to respond to these changes quickly.

Enterprise architectures, using frameworks such as IT4IT, help us to design and manage a multi-cloud architecture that is robust but also scalable in every aspect. We have also seen how IT will shift, along with the business demands coming from traditional to rationalized and dynamic environments using SaaS, containers, and serverless concepts.

In the next chapter, we will learn how to translate the business requirements to cloud requirements, starting our multi-cloud journey.

**Questions**

1. How would you define business agility?
2. What would be the first thing to define if we created a business roadmap?
3. In this chapter, we discussed cloud transformation strategies. Rehost and replatform are two of them. Name two more.
4. In this chapter, we identified major developments in the cloud market. What is recognized as being the main change in business models that can be facilitated by using cloud-native technology?

**Further reading**

* *How Competitive Forces Shape Strategy*, by Michael E. Porter (Harvard Business Review)