Chapter 1
Introduction

Learning Objectives

After completing this chapter, you will be able to . . .

- . . . explain the difference between Enterprise Architecture and Software Architecture
- . . . know typical concerns addressed by Enterprise Architecture Management
- . . . define Enterprise Architecture and Enterprise Architecture Management
- . . . know basic visualisations for Enterprise Architecture

The learning objectives for this section are pretty straightforward. First of all, you should be capable of describing the difference between enterprise architecture (EA) and the architecture of software systems. For those of you who already attended software engineering classes, you might already have a good understanding of what architecture means for software. We will now learn how the term architecture is interpreted on a broader level—at a corporate level. We will also take a look at typical concerns or problems that can be addressed with enterprise architecture methods and tools. We will do this by reviewing some of the literature, but also looking at recent reviews and surveys conducted with corporate representatives. The module will further contain definitions for the terms *enterprise architecture* and also *enterprise architecture management* forming the foundation for our common understanding. Last but not least, we present a couple of examples on how to visualise enterprise architecture so that it can be discussed with business people.

J. Jung, B. Fraunholz, *Masterclass Enterprise Architecture Management*,
https://doi.org/10.1007/978-3-030-78495-9_1

1.1 Motivation

The basic idea of Enterprise Architecture Management (EAM) is represented by the diagram in Fig. 1.1. It starts with a company which has a strategy and corporate objectives. These objectives usually refer to the provision of a product on a market as a result of executing business processes. The execution of business processes will require actors, resources and an organisation. In short, the objective of a company is earning money by providing the products. This is how the company works and this is what needs to drive information technology in the corporation. This might sound straightforward or trivial as all departments need to work together in order to achieve corporate objectives. Naturally the basic understanding should be that whatever information technology is there to support, should support the business with earning money and to be successful.

Nevertheless, when looking at large enterprises you can observe that IT departments developed a culture as being a department of their own. They are focusing on information technology as well as on their own aspects and concerns. IT people, consequently, get disconnected from what the business is doing. The reasons for this mismatch are manifold:

- IT departments are measured against implementing information technology instead of achieving business goals.
- IT is often recognised just as an additional department instead of being part of the value chain (*silo thinking*).
- IT people are very good in defining and implementing information systems but still lack an understanding of the value proposition of information technology for the overall company

At this stage we are looking at EAM for bridging the gap between business and IT. We do not only want to look at information technology and or processes but look at all the concepts together so that we can make sure information technology is achieving the right things—supporting all the business processes in a very high quality way.

1.1.1 Enterprise Architecture and Town Planning

Does this sound familiar to you? *Software Engineering* follows a similar approach: Understand business requirements first and develop a system based on these requirements. However, EA has a much broader view by looking at the whole company instead of a single IT system. Figure 1.2 explains EAM by differentiating it against software architecture by using the metaphor of a town planner. Each software system has an architecture that consists of modules. They are interrelating. We have several software systems that are working with each other. But what is the new aspect of EA?

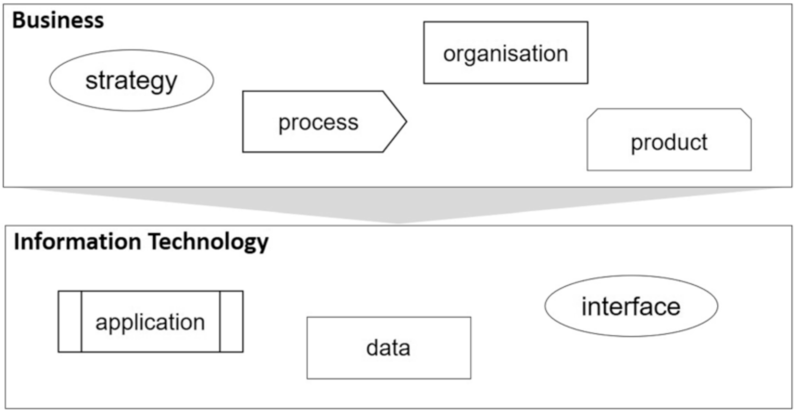
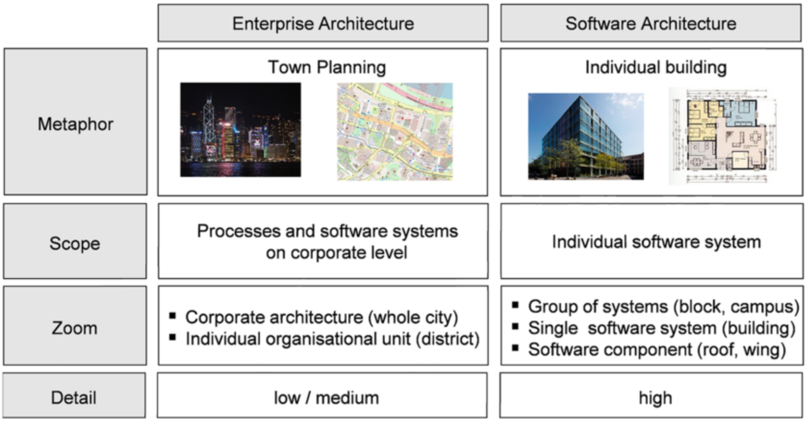


Fig. 1.1 Enterprise architecture management in a nutshell

This metaphor is very common in the literature (e.g. [1–3]). It explains the point of view of the software architect as being the one in charge of building a house or a group of houses. By contrast, Enterprise Architect is more like the town planner. The town planner is responsible for defining the infrastructure of a town, a big city (such as Burwood) or even a metropolitan area (e.g. Melbourne). He or she ensures that we have for example streets, public transport and water supplies. Town planning also includes the definition of rules for building houses—i.e. rules that the architect of an individual house needs to follow. Like in the real world, architects that build houses do not like town planners because they provide rules and restrictions. The town planner is telling them what to do and not to do. It is pretty much the same situation for an enterprise architect in a corporate environment. People developing software systems do not like the Enterprise Architect that much because they are acting in a very similar way.

The scope of the software architect is the one software system that he or she is supposed to implement. Hence, those people are busy with understanding the details of that system, how to implement it properly and managing the software development project. By contrast, the Enterprise Architect focuses on all the processes of a company and the entirety of all software systems that are working together in order to support corporate business processes. The house architect is dealing with details like the installations within a house, the colour of the walls, the placement of power outlets and switches and similar things. These are details that are not relevant for the town planner. The town planner is looking at infrastructure. The town planner is looking at making sure that we have a consistent view of our whole town. The town planner is, therefore, using maps to depict the whole town. This overview of the whole city similar to a helicopter view. He is not looking at the inside of the house but flying over the city having a view of all houses.

The town planner is sometimes supported by further organisational units that represent individual districts. A big city can consist of several districts, each of



	Enterprise Architecture	Software Architecture
Metaphor	Town Planning	Individual building
Scope	Processes and software systems on corporate level	Individual software system
Zoom	▪ Corporate architecture (whole city) ▪ Individual organisational unit (district)	▪ Group of systems (block, campus) ▪ Single software system (building) ▪ Software component (roof, wing)
Detail	low / medium	high

Fig. 1.2 Enterprise architecture vs. software architecture

them managed individually by a seperate mayor. In the same way, we may have the corporate IT also divided into several parts for distinct functional domains like for example customer relationship management IT, production IT supporting our manufacturing plants and finance IT. Each of them is specialised in managing a given functional domain.

The software architect's work is driven by the rules provided by corporate architecture. Building houses is focused on an individual house or also sometimes a group of houses like the university campus. A whole campus is managed by one architect (sometimes heading a team of architects). This is a further analogy to the system architect who can be accountable for a group of interrelated IT systems.

Not surprisingly, the level of detail for building a house is very high. One needs to consider where to place doors, windows, power outlets. These details are not relevant to the town planner. In the same way, the Enterprise Architect will have a high level view with only few details compared to what the software architect is looking at when defining a software system.

This metaphor of town planner compared to the house architect is quite popular in the EA textbooks. The reason for this is because there are a couple of commonalities between the notion of construction, town planning on the one hand and then building a software system or building a corporate IT landscape on the other hand.

An overview on commonalities is provided in Fig. 1.3. One of the commonalities is that we are talking about **complex systems**. A town—imagine Burwood or even Melbourne as a huge metropolitan area—consists of a lot of houses and a lot of infrastructure elements that need to be planned and aligned. We need to make sure that we have enough housing areas and that the quality of living is as expected across the entire city. This is very similar when looking at corporate IT. We have a lot of software applications and IT systems that need to be connected with each other so that they work with each other. They might require the same infrastructure or being implemented based on the same standards. Furthermore we are not only talking

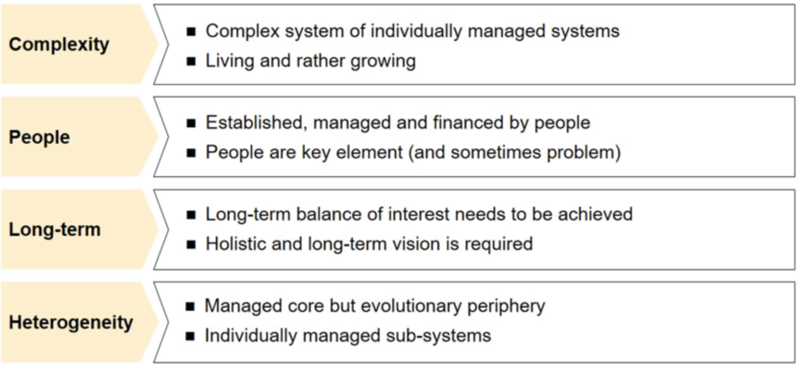

Complexity
Complex system of individually managed systems
Living and rather growing
People
Established, managed and financed by people
People are key element (and sometimes problem)
Long-term
Long-term balance of interest needs to be achieved
Holistic and long-term vision is required
Heterogeneity
Managed core but evolutionary periphery
Individually managed sub-systems

Fig. 1.3 Commonalities of town planning and EAM

about a static system. It never happens that we are planning once and then it is done forever. We are talking about a living organisation or living systems. Melbourne, for example, is expanding and—looking towards Germany—the metropolitan area of Frankfurt is growing as well. This also includes increasing demand in infrastructure. In the same way, a corporate IT landscape will never be fixed but *evolve* over time. This might be the case if the company decides to enter a new market, to develop a new product, or to add a new customer segment.

Perhaps the biggest factor to influence decisions on EA and town planning is, that we need to deal with **people**. Corporate IT is not only technology but also involves people's work. There are people using software applications and people maintaining the IT or making decisions about IT development. Looking at large organisations, it is interesting to see how people identify their role by the IT systems they own. It is similar to owning a small kingdom that defines my power. Hence, those people have an interest in making the system bigger and perhaps not shutting down a system even if it is no longer required. When planning to change EA, we have to incorporate people's attitudes, people's motivation, and people interests. We also frequently need to manage their expectations. (What will happen? Why it will happen?). Town planners need to incorporate the interest of people living in the town. Similar to the town planner, the enterprise architect is not just sitting in an ivory tower doing his or her own planning. There are many different people and many different groups of people that have various kinds of interests with respect to corporate IT.

Especially when looking at the kind of decisions that need to be taken during EAM it is rarely about short term decisions, like making decisions about an individual system change that needs to be done now. It is rather about making decisions about **long term** strategies. How do we want to establish our IT for the future? What will our IT look like in five, ten or even in fifteen years? Of course this needs to be aligned with the business strategy. We should not develop IT as a means of its own but align the development of the IT landscape together with the planned business developments. In order to do this, IT and business need a holistic

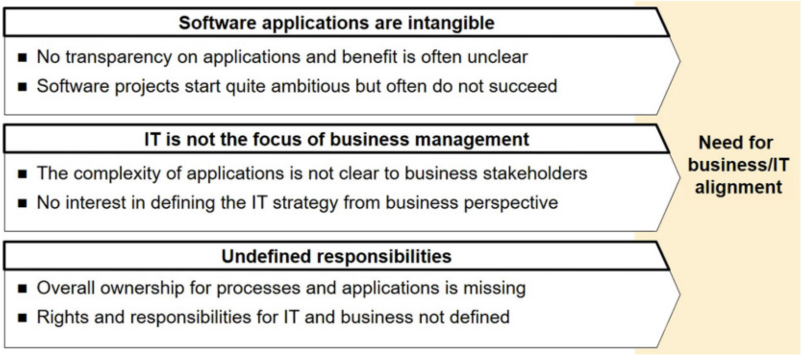

Software applications are intangible
■ No transparency on applications and benefit is often unclear
■ Software projects start quite ambitious but often do not succeed
IT is not the focus of business management
■ The complexity of applications is not clear to business stakeholders
■ No interest in defining the IT strategy from business perspective
Undefined responsibilities
■ Overall ownership for processes and applications is missing
■ Rights and responsibilities for IT and business not defined
Need for business/IT alignment

Fig. 1.4 Differences between town planning and EAM

and long term vision with a common understanding. Where do we want to be in the future? What will the company look like in the future? Which markets will we serve? Which products will we offer to the market? Answering these or similar questions will support making conscious decisions about what IT should look like in future.

Last but not least, the town planner and the enterprise architect have to deal with **heterogeneous** parts of the town and also heterogeneous groups of people. A town can consist of housing areas, industrial areas, shopping areas, infrastructure facilities like fire and police departments. We have a similar situation for the enterprise architect. IT landscapes do not only consist of the same kind of software applications but various types of systems for different purposes. This also includes systems supporting the management of IT infrastructure, allowing data flow between applications and workflow management. This situation is quite common in town planning. You can observe this in German towns very easily. There is something like a core town, the downtown, which is quite well managed. There is also the periphery consisting of suburbs around this town. There are smaller towns and villages serving as housing areas for people working in the city. They also have their own infrastructure, their own industry and commercial areas. The city and also the surrounding areas can be seen as subsystems that are managed by individual parties. A town in the Frankfurt metropolitan area has its own mayor, with its own by-laws and regulations. We have the same situation in a corporate environment. An Enterprise Architect might focus on the common core system, allowing certain functions to have their subsystems which they can manage on their own. Obviously, it is not independently managed. It needs to be aligned at the corporate level. But still we can delegate some responsibilities to suburbs or to surrounding subsystems.

Even though we have some commonalities between the town planner and the Enterprise Architect, there are a couple of differences that we need to keep in mind (cf. Fig. 1.4).

For many people, IT consists of computer hardware and infrastructure for networking. But most importantly it provides software applications that are supporting business processes. The issue with software applications is that they are **not**

tangible, hence, invisible to humans. We can see the user interface, but we cannot see the programming logic working in the background. We cannot see on the screen which other applications they are exchanging data with.

This is a common issue in plenty of software projects (software development, maintenance or shutting down applications). When building a house you can point at existing houses or parts of it. People have an elementary understanding about the physics behind it or you can explain relevant facts very easily. We can only build the first floor after finishing the ground floor, for example. Otherwise, the whole structure would collapse. Such kind of dependencies (e.g. one module requires the finalisation of another) and properties of software systems are not that visible. This is a risk for many software development projects, especially as people do not know how to express their requirements properly. The issue is even worse on a corporate level when you are talking about the systems, but cannot see what those systems are doing and how they are linked together because it is hidden behind the screen.

The second concern is a little bit subtle but still observable in many companies. The IT department is usually recognised as just another supporting department and **not considered as a focus area** for business management. Setting it up like this in the corporate organisation might have been reasonable when technology was rather simple in the past. However, we need to reconsider this as IT is getting more powerful and penetrating all functional areas of an organisation. This especially holds true when thinking of companies in which the whole business model is based on IT or innovative IT systems, even having IT services as a product to be sold on the market. IT is not only emerging in our daily life but also corporate processes. We, consequently, can observe a tighter integration between the business and the IT. But, this distinction is still part of stakeholders' perception, even though companies are already about to transform towards being a digital or an e-commerce company. In order to overcome this gap between business and IT, we need to have a discipline that is not only—like the town planner—produces nice maps, but one that changes the mindset of people involved in business decisions.

This last difference is visible in a lot of companies today. There are software systems available and used by business people, but **nobody on the business side feels responsible** for them. Responsibility would include maintenance (i.e. conducting projects to perform changes), support and funding (e.g. for licenses). It, furthermore, refers to making decisions on the future of individual software applications, like its decommissioning if it does no longer provide relevant value. Making decisions based on business relevance implies an assessment from the business perspective. In fact, many systems are just handed over to IT so that they keep them running but there is no business ownership. IT people then need to drive changes driven by business requirements. Ideally, these should be driven by business stakeholders. We still have to convince people in business and IT to understand the relevance of each IT system and then make sure the IT system has business relevance. We further need clear ownership within the business to take the responsibility for decisions on this IT system.

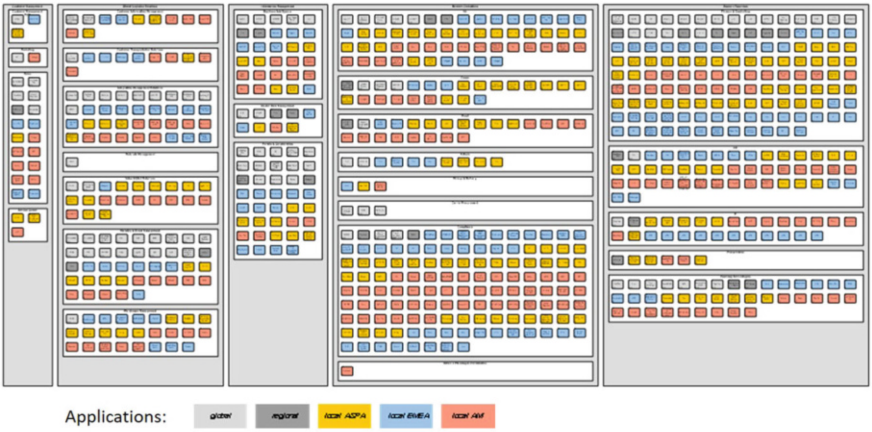


Fig. 1.5 Real world application landscape

1.1.2 *Examples*

Example 1.1 (Typical Application Landscape) So far, we are only talking about business IT, the town planner and the software architect. The complexity behind those decisions to be made by an Enterprise Architect might not be that clear. Just to give you an impression of the complexity and the size of a corporate IT, we present the application landscape of a division within an international forwarding company depicted in Fig. 1.5. The application landscape contains all software applications that are used during a certain period of time. The picture is quite dated and, obviously, it is not readable because we do not intend to reveal restricted material. Just bear in mind that each of those small boxes represents one IT system or software application.

As stated, it only depicts the applications that have been recognised at this point of time. Can you guess the number of IT applications shown here? How many do you think there are? Is it 200, 300, or what number, what would you guess? There were around 850 known applications when the first iteration of this map was created. I counted the boxes in Fig. 1.5 and there are around 750 different applications. It does not look like a lot, but it still means a lot when considering the fact that each box can represent for example one of the following:

- Enterprise Resource Planning (ERP) system like SAP
- Transport Management System (TMS) for planning and monitoring all logistics operations

(continued)

Example 1.1 (continued)

- payroll management system
- finance system for generating and distributing invoices

Each of these can represent a very large IT system with a lot of complexity—and we are still not seeing the relationships between these. Those systems are not used in isolation but are working together. There is data in one application that will also be required by another. Furthermore, EA is looking at corporate business processes fulfilling an entire order for a customer. We are not only using one application but the whole workflow is distributed across various software systems. They encompass activities like accepting and validating the order, planning its execution, manufacturing the product, creating the invoice, managing dunning, collecting cash and later on completing corporate finance tasks. Each of them might be supported by a different system. Nevertheless, they need to work together so that we can fulfil a whole order for a customer.

I already emphasised the term *known applications*. The issue in many organisations today is, that IT has grown over time and people lost the overview of all software systems. You cannot simply go to the shop floor and take an inventory of all machines you recognise. You cannot see software and, therefore, rely on an existing inventory of software applications. Such an inventory can be completed by asking people in various organisational units about software systems they they are using. They will usually provide an answer, but there is no way of checking whether the information is accurate. Did they forget a system or is there a system they do not want to mention? This might happen because stakeholders do not want central IT departments to know about a particular software system.

In summary, having a picture with a lot of bubbles is only indicating the complexity with around 750 software applications. Each of them has a reasonable size, there will be connections between applications. This might only be the tip of the iceberg of the real corporate IT we currently have.

Example 1.2 (Shadow IT and Reoccurring Issues) When looking at IT in a company, we can only provide details about the systems we know. But it is a very common phenomenon that there is something called **shadow IT**. This term refers to IT which exists, but is hidden from central (IT) departments at the corporate level. This happens because individual departments decided to develop their own IT ignoring that there is a centralised corporate IT mandatory for everybody. Decision makers in subsidiaries might prefer control over their own IT instead of depending on a centralised offering.

(continued)

Example 1.2 (continued)
They implement their own infrastructure and their own software systems. And these are not communicated within the company for several reasons, like for example:

- Shadow IT is not following corporate (IT) standards.
- Hidden software applications are often redundant as there are already corporate standard applications in place.
- Redundant software systems cause additional cost to the whole organisation.
- Local software systems might violate global compliance rules.
- Unrecognised IT systems impose security risks (especially if they are not following corporate security guidelines).

Let's keep in mind that many companies are getting cost sensitive with respect to their IT. Hence, companies are aiming to switch off systems to save money. And if shadow IT is creating redundancies those systems will be decommissioned in order to reduce expenses for corporate IT.

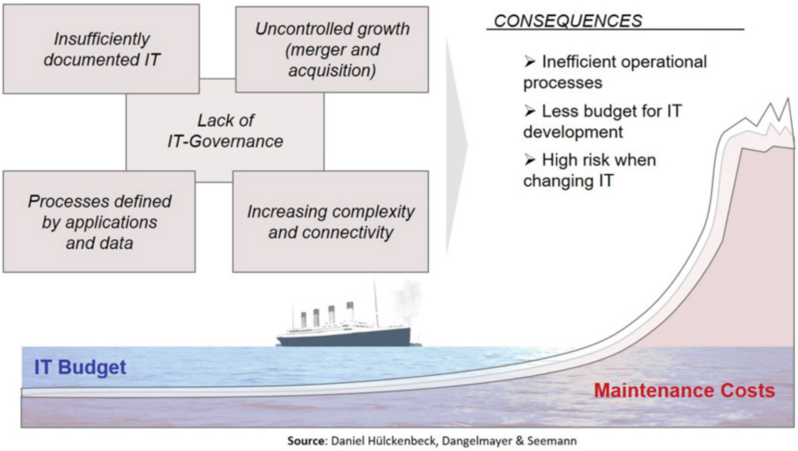
Furthermore, there are systems in use that nobody feels responsible for. Hence, there is **no clear ownership** with respect to maintenance and support.

There will also be systems that we are aware of but **details** are unknown. Required details include the kind of data being processed in the application. This can even be quite risky for a company, especially against the background of data privacy laws that have been established in Europe recently. Companies need to know about personal data of individuals (related to customers or to human beings) and which systems are dealing with them. They need to have an overview of data and applications so that they can ensure that data of individuals is protected properly. But how can you know this if you do not know all the systems? How can you do this if for you do not know details of many systems (including the kind of data processed)? This is one of the challenges that we can address with EA methods and tools presented in Chap. 3.

There are further challenges. Sometimes, the issue is not missing details of the systems, but a missing understanding of its *purpose*. There, for example, are systems that are used to maintain data about warehouses even though the system was originally developed for a completely different purpose. But now the system is misused to maintain the locations of your warehouses.

There might also be systems with functionality that is **obscure** or not known by people. Some old software systems have been developed for a special purpose but have been found useful for another purpose. Such kind of application can confuse IT people as they are supporting a different purpose than they were intended to be used for. Example: An inventory management system that is no longer used for warehousing but is still required to create shipping labels for outgoing deliveries.

(continued)


Insufficiently documented IT
Uncontrolled growth (merger and acquisition)
Lack of IT-Governance
Processes defined by applications and data
Increasing complexity and connectivity
CONSEQUENCES
➢ Inefficient operational processes
➢ Less budget for IT development
➢ High risk when changing IT
IT Budget
Maintenance Costs

Source: Daniel Hülckenbeck, Dangelmayer & Seemann

Fig. 1.6 Situation with unmanaged IT landscape

Example 1.2 (continued)

And last but not least, maintaining a number of **legacy** applications, which are outdated without an adequate user interface. Such applications often have a monolithic architecture and are, therefore, hampering maintenance (e.g. implementing new requirements or improving the user interface).

Example 1.3 (Merger and Acquisition) Having lost control over corporate IT was a key motivator for companies to introduce EA in recent years. You can imagine a large company, which did not grow organically, but rather by merger and acquisition. Let's take a large logistics company as an example. This company did not exist 20 years ago in its current form. It started with the CEO of a postal company making the decision:

> We want to transform this mail and parcel company into a global logistics group. It started with the acquisition of various logistics companies from the postal industry but also businesses from the express, freight forwarding and supply chain industry. The company is now a global player consisting of several divisions, each of them focussing on specific logistics services for different industry sectors. It is not an individual company, but several divisions within a huge corporation consisting of a lot of companies that have been acquired over the past and integrated in a more or less effective way. Also the IT of all the acquired and integrated companies needed to

(continued)

Example 1.3 (continued)

be integrated. With this scenario in mind, we can observe typical issues as mentioned in Fig. 1.6.

First of all, you don't see the software systems, perhaps you can see your data centre (sites hosting server computers) and you can see computers. But you don't see the software applications. Consequently, you rely on having **documentation** including an inventory of all software systems. But in most cases, there is no documentation about your IT system. Other if it is available its quality is rather poor. Documentation is lacking of details and all the different documents look different. You cannot get the whole picture just based on existing documentation.

Because of mergers and acquisitions we do not only have the effect that we have a complex landscape. We also have the effect that we maintain a lot of **redundant systems**. If the integration between various companies is not done properly, we end up in a situation like the logistics company, having . . .

- . . . around 40 systems just for managing payroll
- . . . seven different customer relationship management systems
- . . . more than 30 booking systems

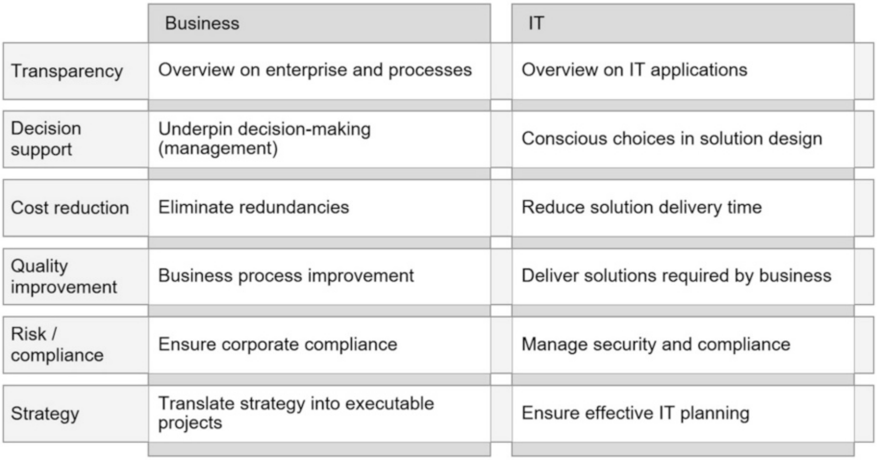
This creates additional complexity by distributing the same kind of data over different systems, not having one single representation or one single source of truth for all data.

You can also observe in this logistics company that decisions on the business processes have not been made based on business facts, but rather based on the established software applications. Having an old large legacy system might not be very flexible with respect to changes. Even if you have new ideas for your IT system, it is usually hard to impossible to change legacy applications. People tend to adjust their behaviour (the way how they work in a corporate environment) to the IT system so that the new process also works with the legacy application. In effect, applications become the **real owners** of processes and data in a lot of organisations.

Furthermore, we had individual departments building their own IT applications, having their own standards, making their own decisions. This caused a rather **uncontrolled growth** of IT systems. Such a growth might not only be caused by merger and acquisition, but also driven by the fact that large organisations consist of many people with different interests just making decisions on their own.

Governance is a common discipline defining and monitoring measures in order to comply to given regulations. Inappropriate or missing **IT governance** will consequently lead to the symptoms mentioned above. This was the case in the past of our logistics company. It needed to learn that if you want to have a good IT to support your business, you need to have a good control on your IT to make sure that IT decisions are aligned with the business. The consequences that large organisations face are also present for any kind of company. We have a growing number of systems and even if the IT budget remains the same, the maintenance cost for all the legacy applications plus the new applications will grow over time. And it will grow significantly until we reach the point that the budget cannot cover the costs any more. This is the main driver for our logistics company to introduce EA.

In contrast to the initial motivation of this chapter, the logistics group does not only want to align business and IT. The biggest driver is cost reduction for IT. Of course not just cost reduction by switching off a lot of IT systems. It requires decisions about IT that still enable the business to continue. But at the same time it is aiming at reducing run and development costs in IT.



	Business	IT
Transparency	Overview on enterprise and processes	Overview on IT applications
Decision support	Underpin decision-making (management)	Conscious choices in solution design
Cost reduction	Eliminate redundancies	Reduce solution delivery time
Quality improvement	Business process improvement	Deliver solutions required by business
Risk / compliance	Ensure corporate compliance	Manage security and compliance
Strategy	Translate strategy into executable projects	Ensure effective IT planning

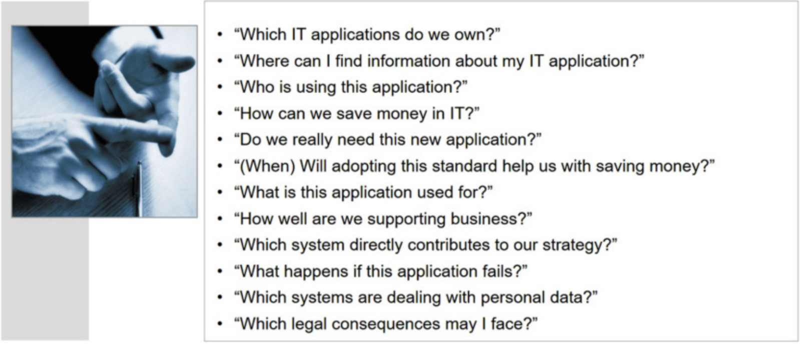
Fig. 1.7 Benefits of EAM (overview based on [4])

1.2 Purpose of Enterprise Architecture

The current section will provide an overview on the purpose of EA. There will be the text book view and also hands-on examples provided by practitioners.

1.2.1 Text Book View

Text books from the EA discipline usually provide an overview on typical purposes of EAM—also as a motivation. Their authors refer to EAM as a tool to provide transparency by showing processes and also IT applications. We can see an overview of typical purposes in Fig. 1.7. Tools provided by EAM can support decisions on the business side—including business process optimisation (left side in Fig. 1.7). They can also support decisions made in solution design, about new software systems or about changes to existing systems (right-hand side in Fig. 1.7). EA and methods promise to support cost reduction by optimising business processes, but also by eliminating redundancy in IT systems, and by supporting faster development of IT systems. If you have a global overview on your business, EA can provide visualisations that help you with improving your business. If you understand your processes and their inefficiencies, then you can improve your processes by performing business process optimisation. If you understood your processes, then you can also deliver software solutions that will perform as expected by business. Textbooks in EA also provide references so we can improve compliance and reduce risk. And last but not least, proper EAM will enable executing the strategy of the corporation.



- “Which IT applications do we own?”
- “Where can I find information about my IT application?”
- “Who is using this application?”
- “How can we save money in IT?”
- “Do we really need this new application?”
- “(When) Will adopting this standard help us with saving money?”
- “What is this application used for?”
- “How well are we supporting business?”
- “Which system directly contributes to our strategy?”
- “What happens if this application fails?”
- “Which systems are dealing with personal data?”
- “Which legal consequences may I face?”

Fig. 1.8 Typical questions for enterprise architects

Do you think that such a kind of list is really helpful in a corporate environment? Just keep in mind you are in a company and people are facing day to day issues with operations and with IT. They are doing a lot of overtime work just by solving current issues. Do you think that you can convince them by promising transparency? Imagine being a consultant and people are stuck with day-to-day routine and you tell them that introducing EA will solve issues by having the big picture. Will this be convincing? Do you think they will immediately tell you that you are hired as a consultant, provide EAM, and then we have transparency and all our problems are solved? Just think about it against the background of your own experience you may have had during internships in companies or by working in companies. Think of your peers, your colleagues or your boss. Which kind of problems do they face and which kind of questions would they expect to have answered by an enterprise architect?

1.2.2 *Practictioners’ Perspective*

In fact, there is a lot of criticism about the way EAM is currently implemented in organisations. Practitioners and also academics are challenging the high level objectives listed in Fig. 1.7. Is this really what we can tell companies to help solve issues? What we did during a couple of months in 2019 was conducting unstructured interviews with decision makers and architects from various companies. We asked them: *What are the real questions you have to answer in your organisation?* We know methods and tools to provide transparency on business and IT. But, which kind of questions are you faced with on a daily basis? Which kind of problems are you asked to solve by your colleagues? Which of them can you solve by applying EA methods and tools?

Results of these interviews have been presented at a conference (cf. [5]) and published as a white paper in [6]. A brief overview of the topics provided by practitioners given in Fig. 1.8. In many companies, it is really a challenge, to tell which applications do we own. In fact from our experience sometimes companies cannot provide a list showing all software applications. Therefore, EAM, beside all the holistic optimisation and so on, should at least be capable of providing an IT inventory including all applications we own. Architects should also be capable of telling people from where to get more information about individual IT applications. This might sound trivial, but keeping a software inventory is still a challenge for large organisations. Let us keep in mind that large companies (also as a result of merger and acquisitions) are growing over time with plenty of systems. Sometimes people do not know about each and every system. Hence, we need to start building a repository. In fact, many EA initiatives started during the past years, started with collecting data about software applications. Also collecting data about users and stakeholders of software applications.

After starting an EA initiative, or when having an overview, one of the most important questions asked by the CIO very often is:

> How can we save money? Architecture optimisation is fine. Supporting business is fine. But my objective is to save money with IT. Please tell me how can I save money with IT

Which also leads to questions like:

> There is this application and I did not understand what it is used for. Do we really need it?

And again, here is another question for the Enterprise Architect to come up with an answer by telling people what this application is doing, what it is supposed to do.

The answer sometimes leads to surprising questions like having a standard in mind. Decision makers sometimes talk to people in other companies or to external consultants. They go to conferences and they get information about new technologies or new standards. They then come back to the company and say

> I heard of all these fancy standards and everybody is telling you adopt this, and then you will have better business. If I adopt this standard, will this help me with saving money and by when? I don't want to wait for ages. I need to provide results within the next period—by the end of next year I need to show savings.

Another trivial question addresses the functionality implemented by an application. We do not only need to know who is using it, but also what it is doing. If we know what the application is doing we also need to understand how well it is supporting business processes. If we provided evidence that applications are required, we also need to document its degree of usefulness. In the situation of saving cost by eliminating redundant applications (i.e. implementing the same functionality), an enterprise architect needs to chose the one supporting the business in the best way. The other ones can be subject to being shut down.

There are also a few topics relating to the strategy and also to issues explained in Example 1.2 on page 9. An enterprise architect needs to know the corporate strategy and make sure that IT is contributing to it. This can only be done if the strategy and the contribution of applications to the strategy are documented in a proper way. Strategy, corporate objectives, Key Performance Indicators (KPI) need to be described and applications need to be linked to them. We will elaborate on this one in Chaps. 2 and 3.

There are also some questions concerning what-if-scenarios (Example: “What happens, if a certain application fails?”). It is a horror scenario in many companies if the central business operations system fails or crashes. Critical business processes cannot be supported any more. Imagine in a parcel logistics company a failure in the central sorting system and parcels cannot be sorted and delivered any more. Customers will not be very happy. The relevance of each application is determined by the business processes that are supported by it. If one has to make a decision which application is more important, an impact analysis is necessary.

According to data protection laws, companies have to protect data of individuals (i.e. protecting personal information). Before doing this, you need to know which systems are dealing with such kind of data. This seems to be a trivial question, especially in a small company with only a few systems. But let us keep in mind that we have a couple of hundred even more than 1000 systems and then answering this question is not that easy. Executives and stakeholders are usually driven by their own interests. People also need to consider legal consequences of violating compliance rules.

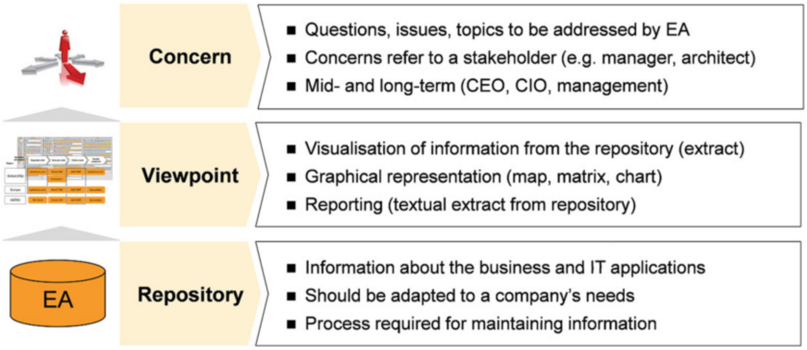
The list in Fig. 1.7 is only a fraction of the topics that have been provided by executives and enterprise architects as typical questions. The entire list is provided in [6] which is still a living document. It will be extended over time and we will also provide a survey getting more information about which of those questions are more relevant compared to others.

1.2.3 Relevance of **purpose**

Why is the notion of the benefit and the problems to be solved so important? Whenever we introduce a new methodology, it needs to provide benefits to the organisation and the same holds true for EA. If we introduce EA, we need people doing it (i.e. an EAM organisation). They would start collecting data and, therefore, involving further resources (e.g. other staff members).

The working mode of an enterprise architect[1] is depicted in Fig. 1.9. Work starts with collecting information about information technology (e.g. infrastructure and software applications) and business-related artefacts (e.g. processes and products). They store it in a dedicated database called a *repository* with EA data. Any kind of information will be collected here and updated over time. Enterprise architects will create visualisations for decision makers based on this repository—called *viewpoints*. You can think of a manager in a company, who might not be interested in doing queries in a database and then seeing a lot of cryptic data. They rather prefer to have (graphical) visualisations that help them addressing their concerns.

[1] The diagram has been created based on a similar discussion in [4, pp. 35] and [7, pp. 5].



Concern

- Questions, issues, topics to be addressed by EA
- Concerns refer to a stakeholder (e.g. manager, architect)
- Mid- and long-term (CEO, CIO, management)

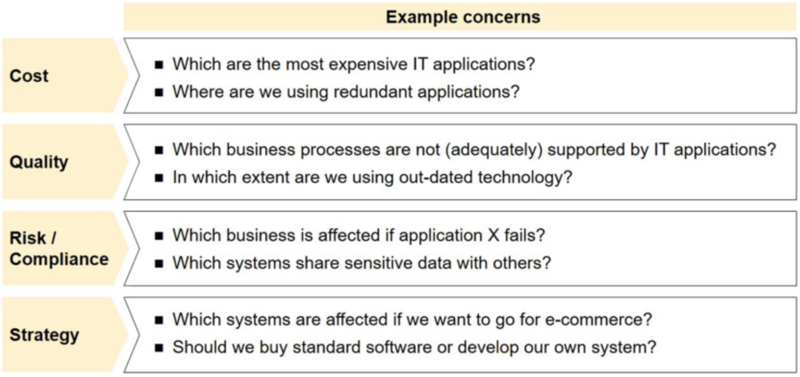
Viewpoint

- Visualisation of information from the repository (extract)
- Graphical representation (map, matrix, chart)
- Reporting (textual extract from repository)

Repository

- Information about the business and IT applications
- Should be adapted to a company's needs
- Process required for maintaining information

Fig. 1.9 Visualising business and IT in order to address concerns



Example concerns

Cost

- Which are the most expensive IT applications?
- Where are we using redundant applications?

Quality

- Which business processes are not (adequately) supported by IT applications?
- In which extent are we using out-dated technology?

Risk / Compliance

- Which business is affected if application X fails?
- Which systems share sensitive data with others?

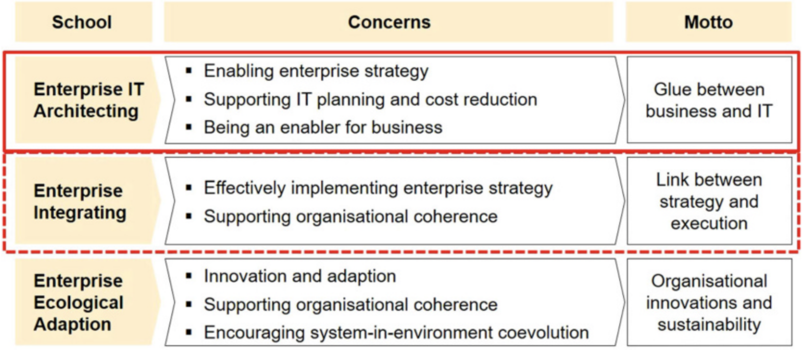
Strategy

- Which systems are affected if we want to go for e-commerce?
- Should we buy standard software or develop our own system?

Fig. 1.10 Example concerns in EAM

Stakeholder concerns are the driving factor for all work done by an enterprise architect. We already discussed those concerns in the context of the purpose of EA in the previous Sect. 1.2. Whatever question we had in Fig. 1.7 on page 13 also represents a stakeholder concern. The visualisation of the architecture of the IT and the processes of the organisation can be used to address those concerns. And this visualisation is usually created out of the repository all the data collected by the Enterprise Architect.Throughout the course we will get more details about which kind of data will be collected, which visualisations will have to be generated and how can it help a company or a stakeholder to make the company better.

A general overview on typical concerns from corporate stakeholders is shown in Fig. 1.10. They address effectiveness (quality of results), efficiency (cost reduction) and compliance at an operational level. They are complemented by strategic concerns from an IT perspective. These concerns will be discussed in a more detailed way in Sect. 4.1. The next section will introduce some common visualisations so that you get an impression of how architecture can be documented.



School	Concerns	Motto
Enterprise IT Architecting	▪ Enabling enterprise strategy ▪ Supporting IT planning and cost reduction ▪ Being an enabler for business	Glue between business and IT
Enterprise Integrating	▪ Effectively implementing enterprise strategy ▪ Supporting organisational coherence	Link between strategy and execution
Enterprise Ecological Adaption	▪ Innovation and adaption ▪ Supporting organisational coherence ▪ Encouraging system-in-environment coevolution	Organisational innovations and sustainability

Fig. 1.11 Enterprise architecture—Three schools

1.3 Enterprise Architecture and Visualisation

Previous sections are presenting the motivation for EA from a rather practical point of view. We were discussing aspects like typical problems to be solved with EA and issues to be addressed. This section will now shift our attention to the academic perspective by providing a definition and references to research. There will also be some example diagrams for showing how (enterprise) architecture can be visualised.

1.3.1 Three Schools

James LaPalme and co-authors performed a literature review to check how people interpret the notion of the EA discipline and EAM [8]. They identified three different directions which they call *school of enterprise architecture* as depicted in Fig. 1.11.

The first one, *enterprise IT architecting*, addresses aspects such as supporting IT planning and cost reduction. We already discussed these topics in previous sections. It includes enabling the enterprise strategy and business processes. They describe it by a metaphor that *enterprise IT architecting* provides the glue between the business and IT. This correlates very much with the notion of *business-IT alignment*.

A further interpretation is called *enterprise integrating* by LaPalme et al. It does not only have a focus on how IT can support business, but also, how EA methods can effectively implement the enterprise strategy and IT strategy. Enterprise integration supports setting up the organisation and implementing business processes. It aims to ensure that the whole organisation is achieving common objectives and a consistent implementation of measures on an operational level in order to follow the corporate strategy. This interpretation refers to being the link between strategy and execution. Please note that it is not only about IT operations. It is about any kind of execution, including workflows, business processes, and activities. Hence, this notion is quite broader than enterprise IT architecting.

The third school of EA is some kind of visionary interpretation. It refers to driving innovations within a company by EA methods and tools. Instead of just performing changes or adopting a given strategy, *Enterprise Ecological Adaption* represents the driver for evolving into a new company or business model in the future. This idea is quite appealing as the environment and markets will change over time. Adopting to a new situation and reacting to changing requirements should be one of the core competences of an organisation.

These three schools provide a summary of how the authors perceive EA based on existing publications during that time. It does not provide any methods or tools but just an overview on how the term can be interpreted. In the course of this book, we will remain with the first interpretation, enterprise IT architecting. We will focus on dedicated methods that can help us to describe the relationship between business and IT and also help us with the analysis of the quality of IT support for the business (i.e. business-IT alignment).

The other two schools, enterprise integrating and enterprise ecological adaptation, are not wrong, but they are more high-level, sometimes even visionary. They are looking at the organisation from a very high level and it will be hard to provide concrete methods and tools. Having a holistic scope will make them quite abstract. They also overlap with existing disciplines like business process management, enterprise modelling, and others that promise similar benefits. Nevertheless, we will sometimes also look at benefits provided by EA beyond business-IT alignment. IT does not only have the potential for being an enabler or providing a translation between business and IT. EA can also drive a coherent business (including IT) or even create new business models. When thinking of digital firms or company providing digital services, we will not even see the distinction between business and IT (at least in the organisation. Therefore, we will also partially address aspects from the enterprise integration meaning deriving a proper execution of the corporate strategy.

1.3.2 Definition of Enterprise Architecture

Let's start with the definition of EA that will be relevant for this book (Definition 1.1). As we saw on the paper published by James LaPalme and others, there are different interpretations. When reading various books or papers about EAM, you might see different ones.[2] Definition 1.1 distils the essence of other definitions.

Definition 1.1 (Enterprise Architecture) Enterprise Architecture (EA) is the representation of an organisation's IT landscape (structure and behaviour) together with its business environment. EA incorporates the current use of IT in the

[2] A systematic literature review on definitions for EAM can be found in [9]. Further definitions are given for example in [10, p. 24] and [2, p. 3].

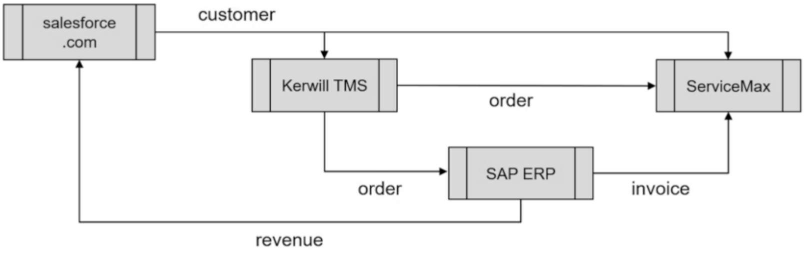


Fig. 1.12 Example: Applications and data flows

organisation (as-is), a vision for future IT support (to-be) as well as a roadmap describing the transition from as-is to to-be.

An EA is the representation of an organisation's IT landscape, together with a business environment. We will not only consider IT, but always incorporate IT in the context of its corporate environment. The notion of IT will encompass structure (e.g. systems and their relationships) and behaviour (functionality implemented by software applications). How do the various systems interact? How are they integrated? And how will all of this provide benefit for the business?

Established methods and initiatives aim at understanding, how does the IT landscape look today. They also support analysis and decision making concerning the future architecture. Where can we reduce costs? Where can we save money? Where do we need new systems for having a new product we want to serve? If, for example, a logistics company delivering letters and parcels also wants to deliver bulk freight, they might need a new system managing the transportation of bulk freight. This needs to be done differently from mail and parcels deliveries and will also require a new software application. In general, business strategy, new products and to-be processes will influence decision about the future IT landscape. When having the as-is picture as well as the to be architecture, we also need a plan for transforming from the as-is to the to-be. This plan will be further called *road map*. The road map is supposed to be a plan containing of certain steps, initiatives and activities that need to be conducted in order to achieve the to-be architecture in the future.

1.3.3 Example Visualisations for Enterprise Architecture

Leaving the abstract definition behind for a moment, let us now look at driving decisions for IT against the business background. We already discussed one of the most important tools of the architects for visualisations: *maps*. Let us get a grip on the notion of map by just looking at a few examples. Which kinds of maps are used in today's EA initiatives?

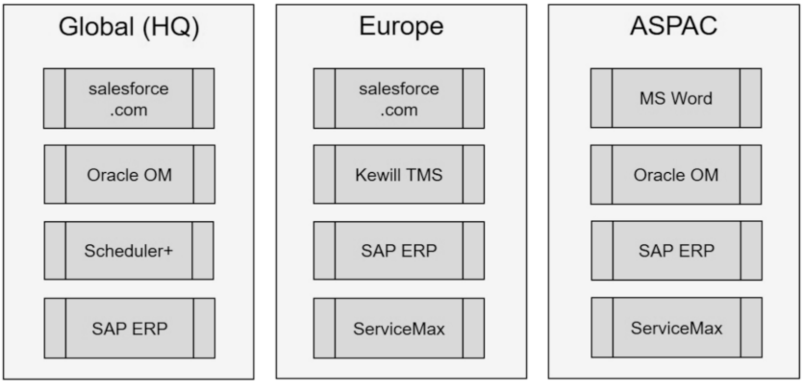


Fig. 1.13 Example: Applications in their business context

There is an example showing four software systems in Fig. 1.12. The arrows between those systems indicate data flowing from one system to the other. salesforce.com, for example is a Customer Relationship Management (CRM) software, maintaining customer data and providing it to Kewill TMS (a Transport Management System) and to our customer service system, called Service Max. Order data is transferred from the transport management system to the financial system—in this case, SAP ERP—and to the customer service system. SAP ERP will then generate and manage invoices based on order and transmit them to the customer service application. Customer service employees will then have all the information required for handling customer requests (e.g. invoice dispute). It is a very simple example. Obviously, it consists of only four systems, which is much less than the 750 or even more than 1000 systems that are common in large enterprises. However, the example should just provide an impression of which kind of information can be represented on an EA map.

Another popular visualisation in EA is assigning systems to organisational units as given in Fig. 1.13. It describes which organisational unit is using which system. In this case, we have, various software applications: salesforce.com again, Oracle Order Management (OM), Scheduler+ (a scheduling system) and others. The example is also showing three different organisational units of our companies, represented by grey columns. We have the global head quarter using four applications. It also shows applications used by a subsidiary in Europe using. And we have an organisation in the Asia-Pacific regions (short *AsPac*) also using several systems. We cannot see what those systems are used for. We can only describe which organisational unit is using which software system by just placing the application within the organisational unit.

Another frequently used visualisation is a map showing individual process steps together with software applications required for their execution (Fig. 1.14). In this

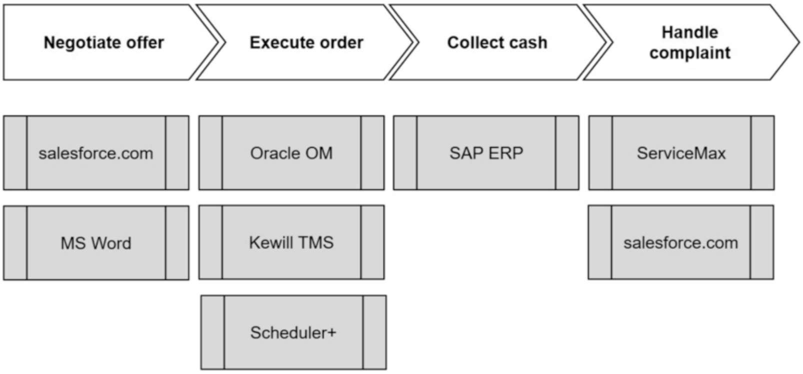


Fig. 1.14 Example: Applications supporting processes

case, we have a complete order management process, which starts with negotiating an offer with the customer. Such an offer can, for example, refer to a car having a certain configuration (e.g. engine, car, extras). This is some sales activity leading to executing the order after the customer made the final decision on accepting the offer. Data can then be provided to the manufacturing department, which will produce the car and deliver it to the customer (as part of *Execute order)*. After delivering the car to the customer, he/she will have to pay for it (an amount agreed upon during negotiating the offer). This will be performed in the *Collect cash* process step. And in case there is any issue with the car, he might come to our office and complain about it. It might, for example, not work or it is too noisy or it is not as fast as expected. In essence, there might be complaints by the customer that need to be handled in the last process step..

A similar kind of process may exist in any kind of industry. Another example would be the logistics industry: It starts with having a customer asking for the transportation of a huge amount of cars, or transportation of live horses that need to be delivered to a competition somewhere in another country. The logistics company can provide an offer for this transportation by using data from the CRM system and creating the offer using Microsoft Word (cf. Fig. 1.14). When the customer decides yes, I want you to move my horses from Germany to Australia for the race, then we can execute the order using the Oracle Order Management system. The company also uses the Kerwill transport management system for planning the transportation, for booking flights with airlines, which will then transport the horses. Additional timing and scheduling will be performed with Scheduler+ as shown in Fig. 1.14. Furthermore, the company can use other systems like SAP ERP system for all the financial aspects and as we can see here two different systems for customer service (e.g. handling complaints). Let us also keep in mind here that it is just a small example for illustration purposes—showing concepts and typical representation. Of course, a real map in a corporate environment is much larger. We should always keep in mind a typical corporation has a couple of hundred or even more than 1000

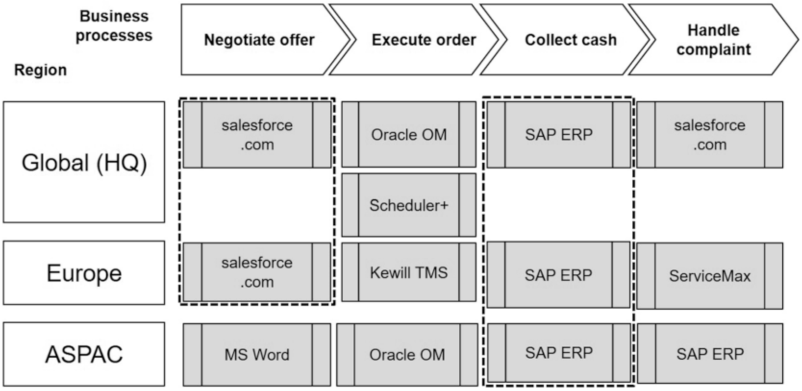


Fig. 1.15 Example: Combined viewpoint

applications (as shown in Fig. 1.5 on page 8), business processes are larger and have more detailed steps. We will usually have much more information. Drawing it in PowerPoint is good for showing the concept as we are doing here. But we will need a dedicated tool for analysing real world EA.

The examples presented so far are each focusing on a single aspect. We can also combine different views as shown in Fig. 1.15. Figure 1.14 is showing process steps for the corresponding IT applications. Figure 1.13 shows IT applications used by different organisations. Figure 1.7 combines the two viewpoints into a new visualisation: Showing systems per organisational unit, which represent regions here. And then showing which system is supporting which step in each organisational unit.

The examples provided in the section at hand are just meant to give an impression on simple visualisations used in EA. We will discuss further viewpoints in subsequent sections. There are also more examples in available publications like for example [11]. After now having some idea of what EA means, we will now have a look at what managing an EA is about.

1.4 Enterprise Architecture Management (EAM)

The previous sections provide a brief overview on EA and its visualisation. They addressed questions like: *How can we describe business and IT?* The examples are just meant to provide a short introduction and there will be more visualisations in subsequent chapters.

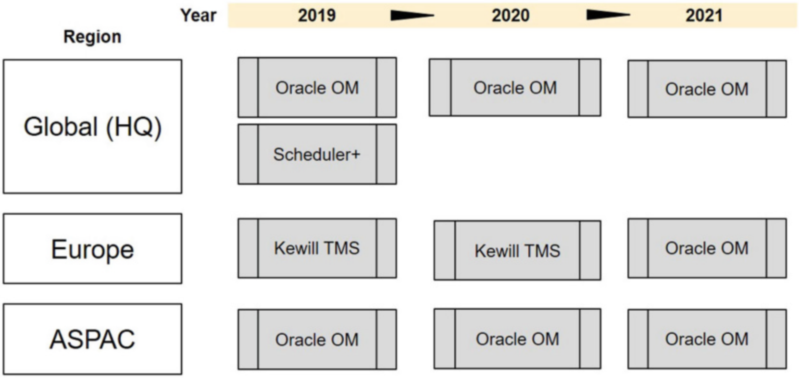


Fig. 1.16 Example: Roadmap

1.4.1 Definition of EAM

Before going into detail with method and tools, we also need to define the notion of EAM (cf. Definition 1.2). We characterised EA as a representation of the IT landscape and its business context, including time. We also need to consider changes in EA as we have an as-is state and want to develop it into an optimised architecture (to-be). This is exactly provided by EAM, which is a structured approach for establishing EA and maintaining the maps showing the business and applications landscapes. EAM is further aligning IT with corporate objectives defined by the business. EAM also provides methods and organisational structures and best practises that help us with introducing and performing EA activities in a corporate environment.

Definition 1.2 (Enterprise Architecture Management) Enterprise Architecture Management (EAM) is a structured approach for establishing, maintaining and using EA in order to align IT with corporate objectives. EAM defines methods and an organisational structure for enabling EA activities.

1.4.2 Roadmaps for Visualising Transformations

What does it mean, performing a change in the context of EAM? Again, we have a small example showing how we can describe the evolution of our application landscape over time in Fig. 1.16. Please note, we are using the same organisational units on the left-hand side as we did in previous examples in Sect. 1.3 (corporate headquarters, the European subsidiary, and also the Asia-Pacific organisation). Each of them is using its own set of software applications.

We can see a lot of different applications used perhaps for the same purpose in Fig. 1.16. Global HQ is using Oracle Order Management and Scheduler Plus. The Europe organisation is using a completely different system, called Kewill TMS, and the Asia-Pacific region is using Oracle Order Management (same as Global HQ). Let us now consider the fact that, IT people at EAM will focus on eliminating redundancies (i.e. saving money).

One might have made the decision that we need to reduce the amount of systems. And those two systems, Oracle Scheduler and Kewill TMS, are providing the same functionality. In addition to that, we want to get rid of Scheduler+ by end of next year. As the ASPAC organisation can only work with Oracle Order Management, it should also be possible to use only Oracle in the global headquarters. Consequently, Scheduler+ will be phased out (i.e. shut off). People might recognise during further analysis that Kewill TMS and Oracle are doing pretty much the same. We should, consequently, decommission Kewill TMS and replace it by Oracle Order Management in Europe. By the end of the year 2021, we will have only one system supporting order management globally. Oracle OMS will be the standard system within the organisation.

Why this example in Fig. 1.16? First of all, this is a typical representation showing how those kinds of changes are documented. It is providing a time frame for changes (years 2019 until 2021) and the changes that are planned to be conducted during that time. The example is also showing typical changes like

- eliminating a software application (Scheduler)
- replacing one application (Kewill TMS) by another (Oracle OMS)

There are more kinds of changes that may be triggered as a result of EA analysis. They will be explained in the course of EA analysis in Chap. 4 and when explaning the role of an enterprise architect in Chap. 5.

Another example for documenting changes on corporate IT is provided in Figs. 1.17 and 1.18. A roadmap showing the transition from as-is to to-be does not necessarily have to be documented by one map only but can also span several maps. We can have different maps documenting a change—so starting with the map we have already seen previously for showing which process step is supported by which IT application (Fig. 1.17). Based on this view, we might consider changes in order to optimise our landscape and derive an ideal to-be state as depicted in Fig. 1.18. Starting with a dedicated as-is map helps with focusing on understanding current inefficiencies and, therefore, optimisation potential.

Changes can then be documented by shifting the lens towards a to-be view as in Fig. 1.18. We should try to get rid of certain systems by 2020, like stating we don’t want to write each and every offer using Microsoft Word. We want it to be generated on Salesforce.com. We are also owning two software applications for complaint handling. Why are we providing redundant systems for customer relationship management? Salesforce.com can also be used for the same purpose. Somebody made the decision that, by 2020, we want to switch off MS Word and ServiceMax in our to-be architecture.

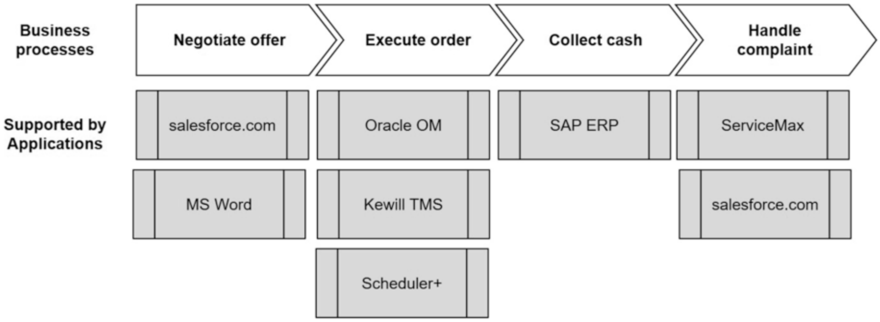


Fig. 1.17 Example: As-is as starting point for transition

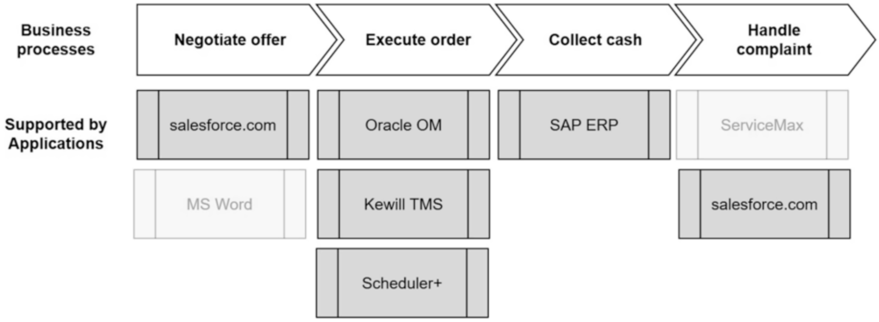


Fig. 1.18 Example: To-be as objective for transition

A roadmap can generally consist of at least two maps, showing the as-is and the to-be architecture as provided in Figs. 1.17 and 1.18. More complex roadmaps may span all changes over several maps, showing transformations over a period of time.

1.4.3 EAM and Related Disciplines

We are about to end this Chap. 1 and should have some basic understanding of EA and EAM. We should not forget about the fact that EAM is not a universal discipline for managing a company and solving all problems. We already know a couple of other management disciplines already being implemented in companies (Fig. 1.19) which are subject to academic research.

We have the discipline of **project portfolio management**. Companies are not only conducting projects, but having many projects that have interdependencies, and we need to make a decision about which projects we want to execute now, and in which order. And also, the technical discipline of **IT infrastructure management**. Even though EAM always promises to be a holistic approach, it is not meant to

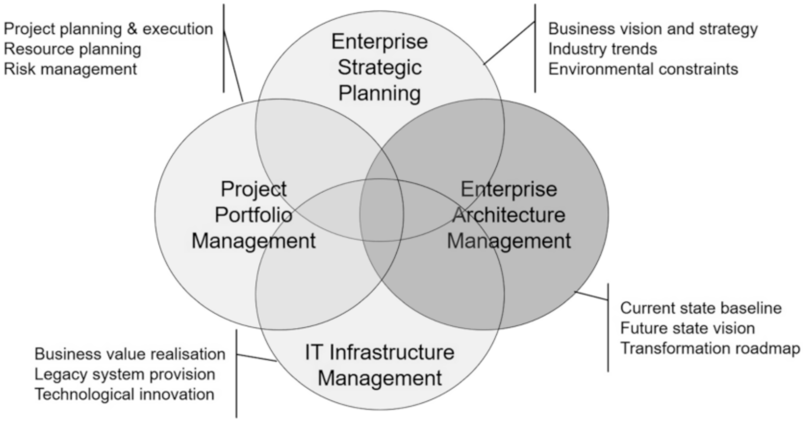

Project planning & execution
Resource planning
Risk management
Enterprise Strategic Planning
Business vision and strategy
Industry trends
Environmental constraints
Project Portfolio Management
Enterprise Architecture Management
Current state baseline
Future state vision
Transformation roadmap
Business value realisation
Legacy system provision
Technological innovation
IT Infrastructure Management

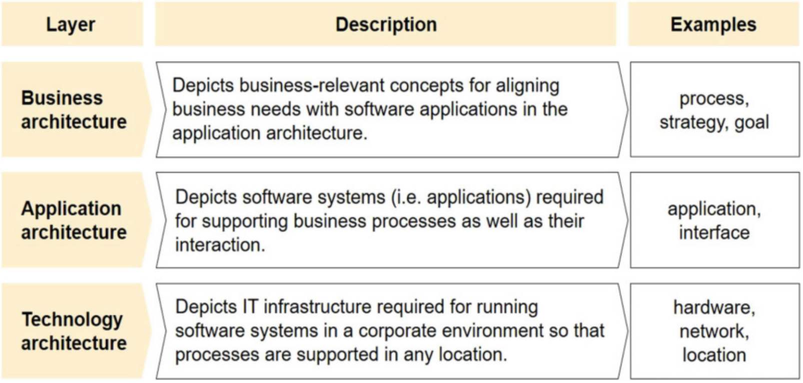
Fig. 1.19 EAM related to other disciplines (based on a figure in [12, p. 36])

replace all the other disciplines. To be more precise, for EAM, we need input from the **strategic planning**. Strategic planning still provides methods and tools for analysing the company, the context, markets, and so on, or making conscious decisions about, how do we want to act in the future, which will then be the input for EAM.

EAM is dealing with IT and software applications. This will have an impact on how to manage the IT infrastructure. IT infrastructure management is not part of EAM. It still has a couple of aspects in addition to what we will discuss in the course of EAM. Nevertheless, there is a huge overlap. And the same holds true for project portfolio management. We already mentioned potential changes that are performed on the corporate application landscape. We can introduce new software systems, switch off legacy systems that are not used anymore. We can change existing systems with respect to new requirements, or merge individual software applications into one system. Each change needs to be conducted within a project. And as EAM is rather having the holistic view, it is not only one project being triggered by EAM, but a portfolio of projects, which will use methods and tools also from project portfolio management. Therefore, it is not one toolset being used for managing a company. EAM coexists with others and will help with aligning business and IT properly as well as providing information and decisions that can help the other disciplines.

1.4.4 Architectural Layers

Defining architectural layers is a common concept that is part of each and every method, or part of each and every *EA framework*. It is driven by the principle divide



Layer	Description	Examples
Business architecture	Depicts business-relevant concepts for aligning business needs with software applications in the application architecture.	process, strategy, goal
Application architecture	Depicts software systems (i.e. applications) required for supporting business processes as well as their interaction.	application, interface
Technology architecture	Depicts IT infrastructure required for running software systems in a corporate environment so that processes are supported in any location.	hardware, network, location

Fig. 1.20 Common enterprise architecture layers

and conquer, not having the architecture as a whole. But rather cutting the entire view into several layers or into several points of views.

We will use a very simple distinction of layers in our text book and it is shown in Fig. 1.20. There is one layer called the *business architecture* (or also *business layer*). It contains all details and fragments that describe the business. We can use, for example, business processes, the strategy, KPI, objectives, goals for describing the business environment. We will cover this topic in the following Chap. 2. The term *business architecture* is often used by enterprise architects when they refer to their knowledge about the business.

Below the business architecture is the *application architecture* (or *application layer*). The application architecture consists of all software applications that are used and maintained by an organisation. It also includes data within the applications, data flows and interfaces between applications. Real world application landscapes tend to grow very large. All of this will be covered by the notion of application architecture in Chap. 3.

Separate from the top two layers is a third layer called *technology architecture* (*technology layer*). Technology architecture encapsulates all hardware and IT infrastructure (like networks, data centres, location where we need to establish and maintain IT). It is representing a very technical perspective.

Classical EA frameworks can have even more layers and a more complex way for arranging them. For our purpose such an easy layering making the distinction between business and applications and then also providing IT technology in addition is quite sufficient.

We will not elaborate on technology architecture, IT infrastructure and we will not discuss IT service management or any related topic. This book basically focuses on

- How can we describe the business architecture as the top layer of Fig. 1.20? (Chap. 2)
- How can we describe the application architecture? (Chap. 3)

- How do we link applications to business-related concepts (i.e. relationships between business and application architecture)? (Chap. 4)
- And how can we drive changes on the application architecture for the future? (Chap. 5)

1.5 Further Reading

We finish this chapter with an overview on existing publications that are worth reading. First of all, there is one textbook published by a Dutch guy with the last name Op't Land and co-authors (cf. [4]). Even though it is quite old they provide a good overview on frameworks. Also on the organisational aspect of and also the question, how can we create value by applying EA?

As we saw on the slides the value they are promising is rather high level. Like providing transparency, we can reduce risk, we would save costs. Real world enterprise architects have quite more hands on questions (cf. [5]). We collected them in documents that will be provided as supplementary material in [6]. There are further reviews like the one published by Lindström et al. in [13].

We also have to look at the theory behind EA—also looking at what researchers are publishing about this one. One prominent paper is published by James Lapalme and co-authors called *Three Schools of Thought on Enterprise Architecture* [8]. One of those schools is having a focus on *enterprise IT architecting* but there are also wider views up to how EA being the innovator or the driver for innovations within a company.

There are also a couple of publications on the definition of the term *enterprise architecture*. Saint-Louis and co-authors provide a systematic literature review in [9].

There are a couple of books available at the moment who are criticising the classical EAM approach with being on high level a lot of documentation also suggesting we need to incorporate more agile and lean techniques. One of them is published by Bente and two co-authors (cf. [12]). The three of them are consultants with Tata consultancy and developing their own methodology for the start for understanding what is EA being used for. It is a good introductory reading. We will reiterate the same book later on in the book when talking about setting up an organisation for EAM.

And last but not least, there is a book published and authored by an Australian guy with the last name Kotusev (cf. [14]). He is also located in Melbourne and provided some textbook with an overview on typical visualisations being used in EAM. This book is some kind of a recommendation for everybody who wants to see which different kinds of maps are being used. Of course, we will present and discuss further views in the following chapters of the book. We started already with some simple examples. But there is more to come …

1.6 Summary

That’s it so far for the first chapter. During the introduction to EA, we started with describing why is EA different from system or software architecture. We are not only thinking of individual software applications, but of the IT landscape as a whole. We had the metaphor of the town planner, compared to the architect creating an individual house. We then looked at how is EA beneficial for us and which kinds of problems can we solve with applying EA methods and tools by having some definitions—providing definitions on how we see the notion of EA and also having some typical maps of visualisations to be used by enterprise architects. It was not meant to be a complete overview, but just having some typical examples, just for making the theoretical part a little bit more visual.

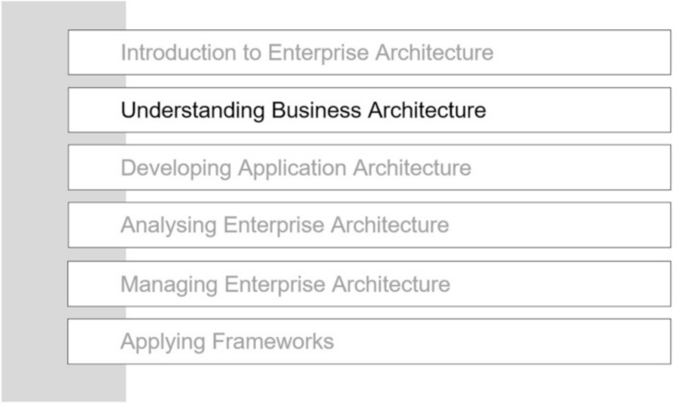

Introduction to Enterprise Architecture
Understanding Business Architecture
Developing Application Architecture
Analysing Enterprise Architecture
Managing Enterprise Architecture
Applying Frameworks

Fig. 1.21 Following next: Understanding business architecture

In the next section, we will have a closer look at *understanding business architecture* as depicted in Fig. 1.21. At the end of this section, we introduce the distinction between business architecture and application architecture. The two of them will be further looked at in the following Chaps. 2 and 3. The next one will focus on the business architecture, by covering the following topics:

- Are business processes the best way for aligning business and IT?
- What are business capabilities and how can they be used for deriving an application landscape?
- Which further concepts exist on business architecture?