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?

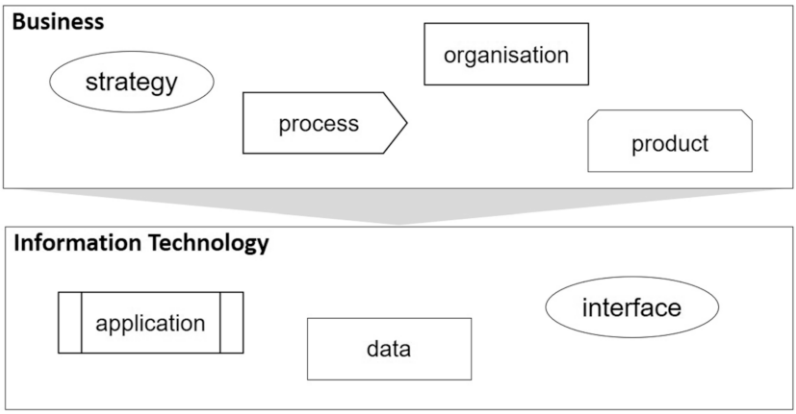

Fig. 1.1 Enterprise architecture management in a nutshell

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

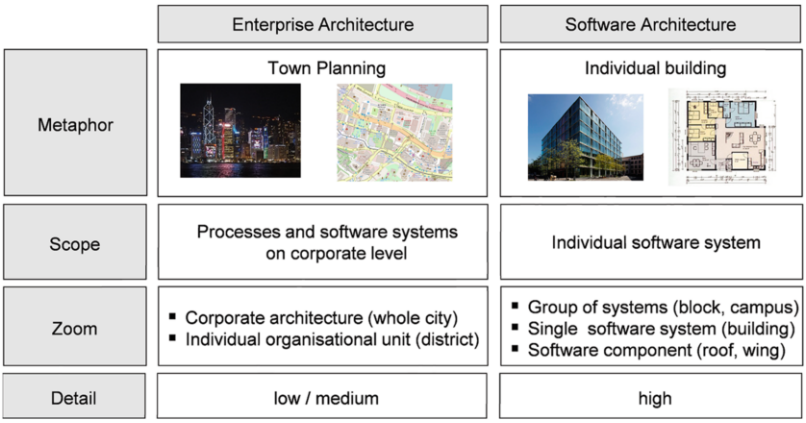

Fig. 1.2 Enterprise architecture vs. software architecture

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

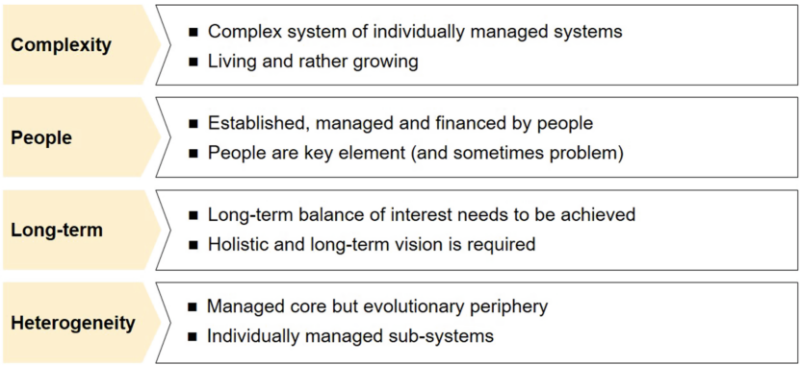

Fig. 1.3 Commonalities of town planning and EAM

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

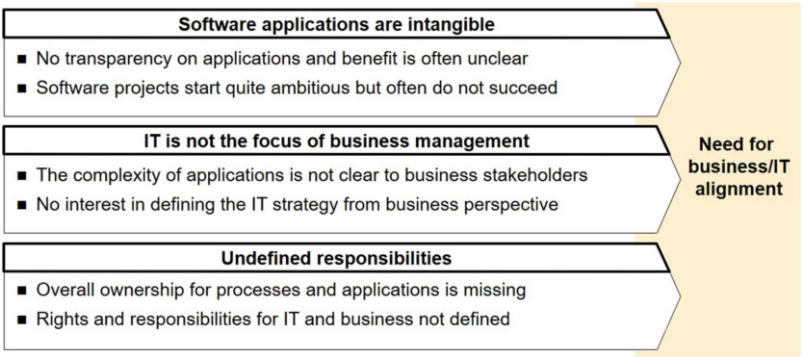

Fig. 1.4 Differences between town planning and EAM

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.

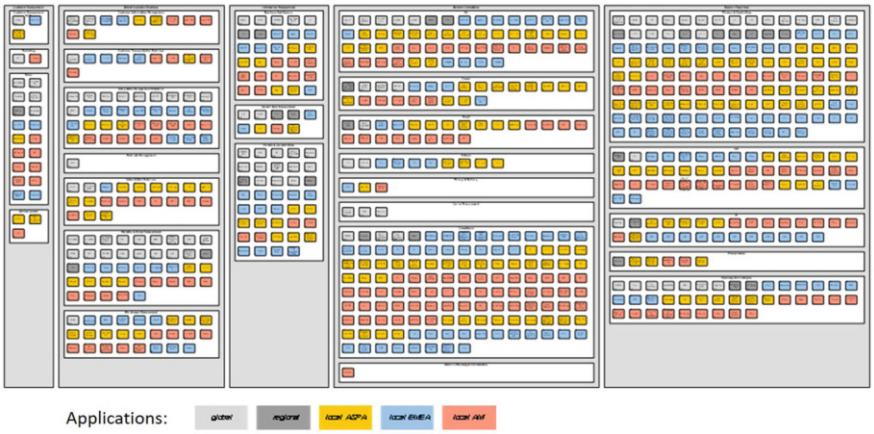

Fig. 1.5 Real world application landscape

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)

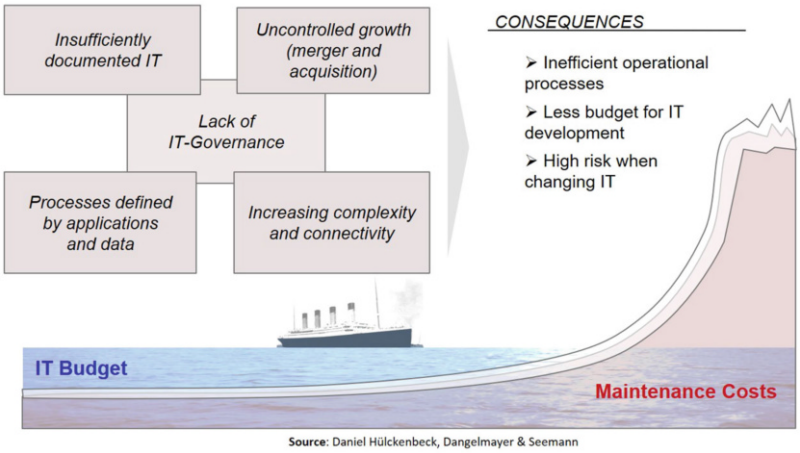

Fig. 1.6 Situation with unmanaged IT landscape

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.

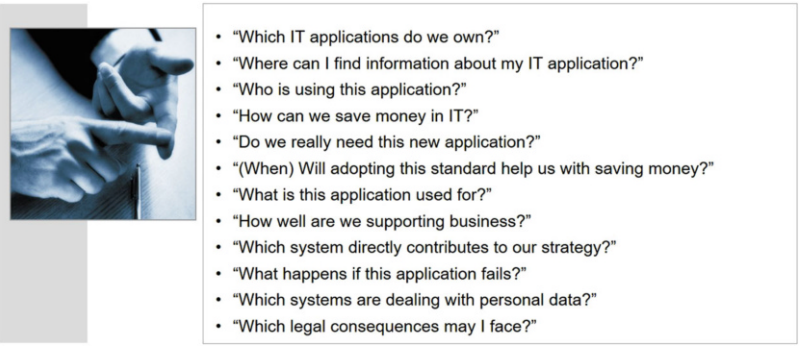

Fig. 1.8 Typical questions for enterprise architects

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].

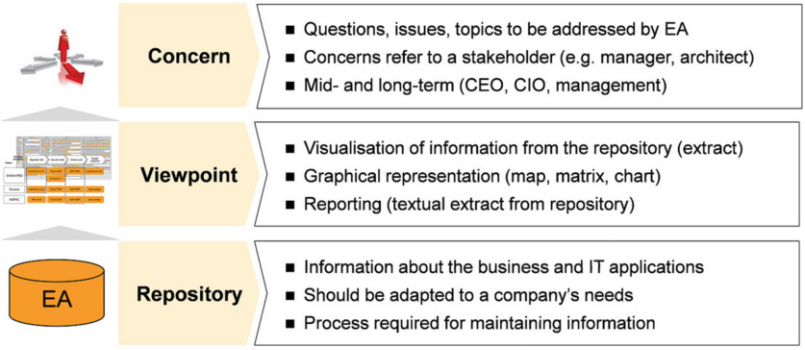

Fig. 1.9 Visualising business and IT in order to address concerns

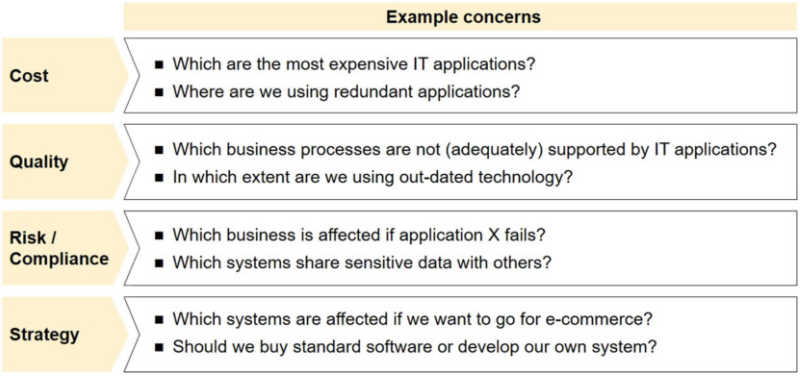

Fig. 1.10 Example concerns in EAM

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.

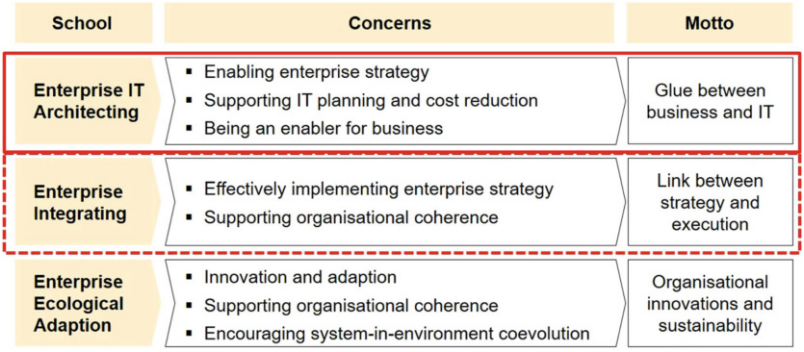

Fig. 1.11 Enterprise architecture—Three schools

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].

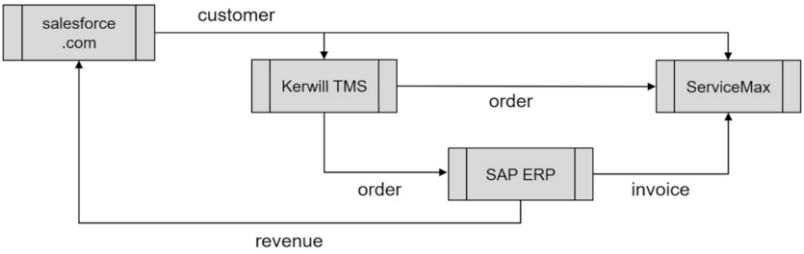

Fig. 1.12 Example: Applications and data flows

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?

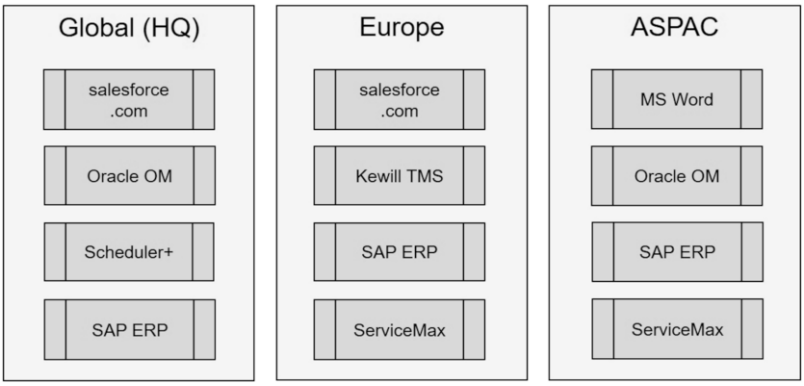

Fig. 1.13 Example: Applications in their business context

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

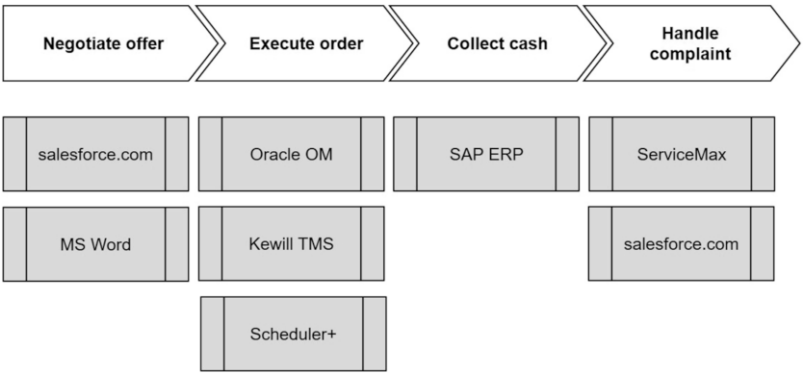

Fig. 1.14 Example: Applications supporting processes

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

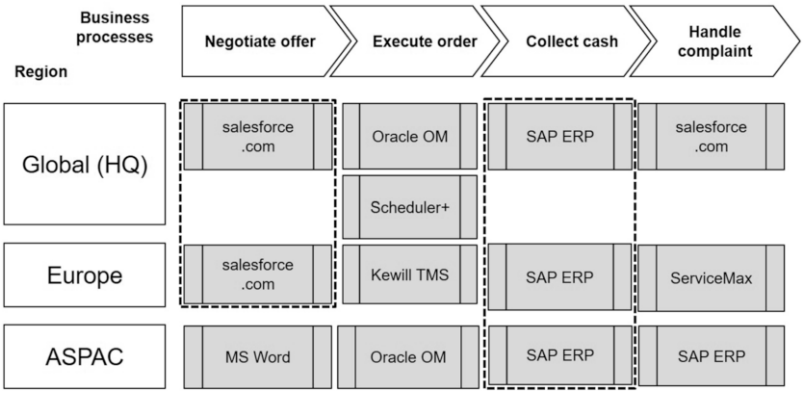

Fig. 1.15 Example: Combined viewpoint

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.

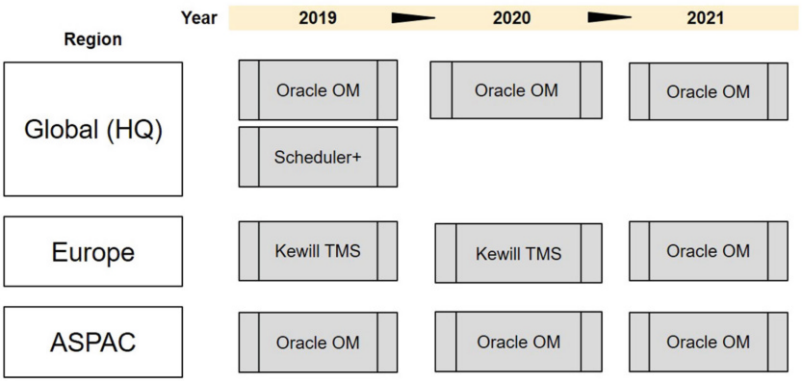

Fig. 1.16 Example: Roadmap

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.

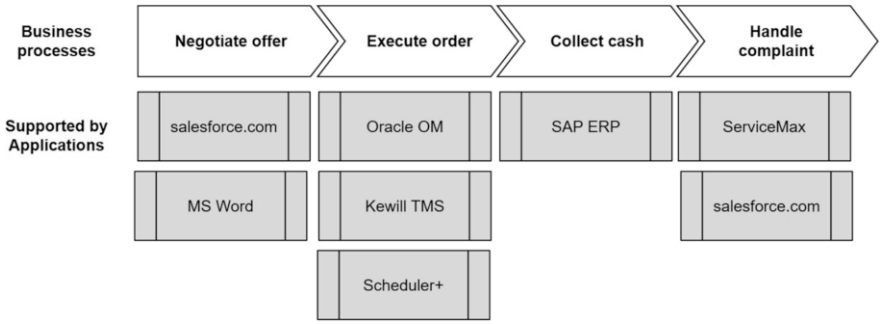

Fig. 1.17 Example: As-is as starting point for transition

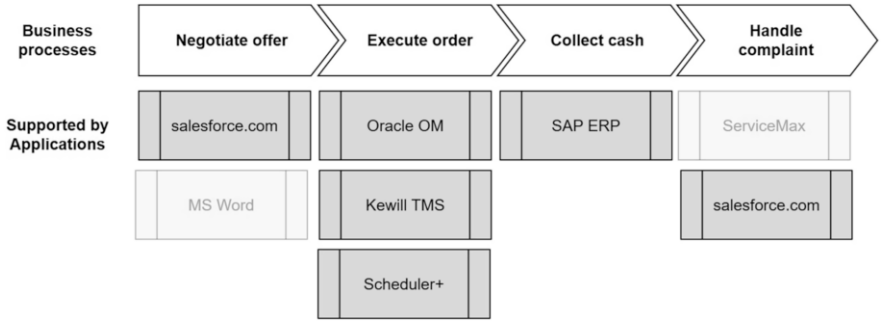

Fig. 1.18 Example: To-be as objective for transition

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

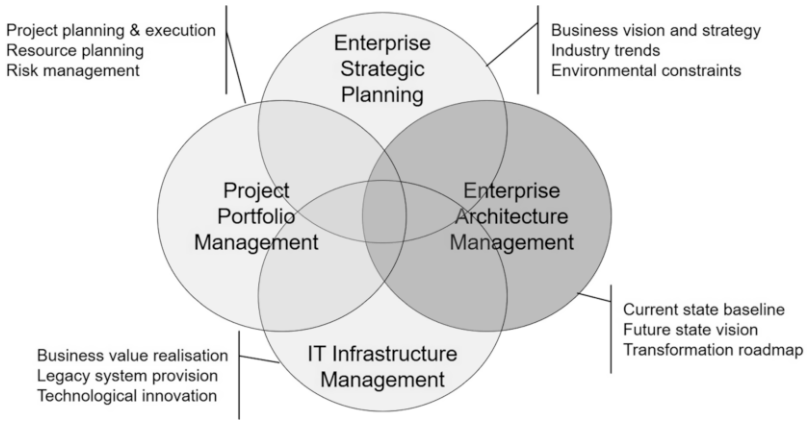

Fig. 1.19 EAM related to other disciplines (based on a figure in [12, p. 36])

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

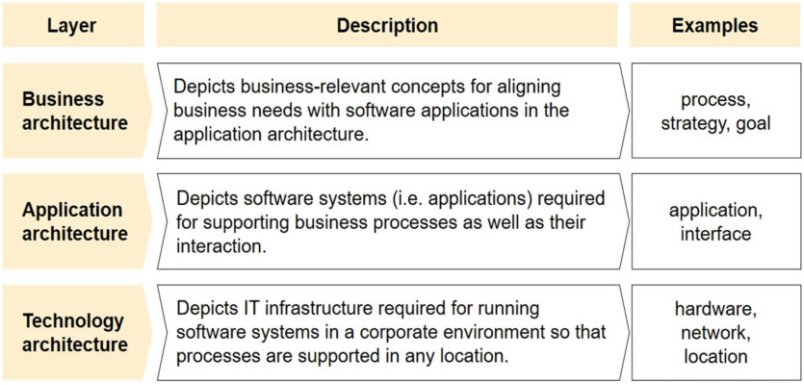

Fig. 1.20 Common enterprise architecture layers

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.

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?

1.7 Exercises

Exercise 1.1 (Legacy Systems) Please, answer the following questions using your own words:

1. What is a legacy system? You can look up the term in the literature.
2. Which are common problems associated with legacy systems? Try to find some in publications on legacy applications (i.e. software, system).
3. Why are legacy systems still in use? You can investigate for reasons in publications on the topic or also mention some from your own experience.

Exercise 1.2 (Your Experience) Consider a company or organisation you already know (like current/past employer, internship or business partner). Please, summarise your experience with respect to the following questions:

1. How well do IT people know business processes of that company?
2. How do business and IT people collaborate?
3. Is IT contributing to the business model of the company?

Exercise 1.3 (Purpose of EAM) Download the following document [5] from your library and read it carefully. Which of the topics listed there are relevant from your experience?

Exercise 1.4 (Digital Enterprise) Identify at least **three** companies who transformed their business model from classical products into a digital service offering. How does the separation between business and IT influence a company's capability with respect to a digital service offering?

Exercise 1.5 (Start-Up) Bob founded a company based on venture capital. He needs to set up IT for key corporate functions.

1. Which drivers are influencing the IT?
2. Which challenges might Bob need to be aware of?

Exercise 1.6 (Merger/Acquisition) A big company CA bought the main competitor CB. Information technology of CA and CB needs to be integrated.

1. Which challenges need to be addressed?
2. How should the IT look like ideally?

Exercise 1.7 (Business Context) How can we describe or visualize the following business concepts?

- Strategy, objectives and KPI
- Business processes
- Organisation and resources
- Software systems and data
- Cost an benefits
- Risk and mitigation actions
- Compliance rules

Choose at least three from the list above and provide examples on how to visualise or describe them.

References

1. Y. Namba, City planning approach for rebuilding enterprise information systems, Ph.D. Thesis, Tokyo Institute of Technology, Jan. 2005
2. M. Lankhorst, et al., *Enterprise Architecture at Work: Modelling, Communication and Analysis*, 3rd edn. ser. The Enterprise Engineering Series (Springer, Berlin, Heidelberg, New York, NY, 2013)
3. J. McKee, *Applying Principles from IT Architecture to Strategic Business Planning* (Business Science Reference, 2013)
4. M. Op't Land, E. Proper, M. Waage, J. Cloo, C. Steghuis, *Enterprise Architecture, Creating Value by Informed Governance*. ser. The Enterprise Engineering Series (Springer, Berlin, London, 2009)
5. J. Jung, Purpose of enterprise architecture management: Investigating tangible benefits in the german logistics industry, in *2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*, Oct. 2019, pp. 25–31
6. J. Jung, E. Proper, *Why Are Practitioners Doing EAM? Tangible Benefits from Enterprise Architecture Management*, Frankfurt, 2019. [Online]. Available: http://eamlab.frankfurt-university.de/wordpress/purpose-of-eam/
7. P.A. Khosroshahi, M. Hauder, A.W. Schneider, F. Matthes, Enterprise architecture management pattern catalog, Version 2.0, Software Engineering for Business Information Systems (sebis), Technische Universität München, München, Tech. Rep., 2015
8. J. Lapalme, Three schools of thought on enterprise architecture. IT Professional **14**(6), 37–43 (2012)
9. P. Saint-Louis, M.C. Morency, J. Lapalme, Defining enterprise architecture: A systematic literature review, in *2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW)* (IEEE, 2017), pp. 41–49
10. D. Greefhorst, E. Proper, *Architecture Principles: The Cornerstones of Enterprise Architecture*, ser. The enterprise engineering series. (Springer, Heidelberg and New York, 2011
11. S. Kotusev, Enterprise architecture: What did we study? Int. J. Coop. Inf. Syst. **26**(04), 1730002:1–1730002:84 (2017)
12. S. Bente, U. Bombosch, S. Langade, *Collaborative Enterprise Architecture, Enriching EA with Lean, Agile, and Enterprise 2.0 Practices* (Elsevier/Morgan Kaufmann, Amsterdam, 2012)
13. Å. Lindström, P. Johnson, E. Johansson, M. Ekstedt, M. Simonsson, A survey on CIO concerns: Do enterprise architecture frameworks support them? Inf. Syst. Front. **8**(2), 81–90 (2006)
14. S. Kotusev, *The Practice of Enterprise Architecture: A Modern Approach to Business and IT Alignment* (SK Publishing, Carlton, VIC, 2018)

Chapter 2
Understanding Business Architecture

The previous Chap. 1 introduced the definitions for *EA* (Definition 1.1 on page 19) as well as *EAM* (Definition 1.2 on page 24). It also discussed why such a discipline like EA is required (Sect. 1.2) and how it relates to other established disciplines (an overview is provided in Fig. 1.19 on page 27).

We already had talked about that EA is usually divided into several layers (depicted in Fig. 1.20 on page 28). The top one is called *business architecture* and will be subject of the current section. Subsequent chapters will then deal with further architectural layers, like the application architecture. We will talk about analysing EA throughout the book, and how to manage an EA. But now, let's have a look at business architecture.

Learning Objectives

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

- . . . explain the difference between business processes and capabilities
- . . . create a business capability map
- . . . identify relevant business objects
- . . . provide an overview on business architecture concept

The learning objectives for these sections are as follows. After carefully studying this chapter, you should be capable of explaining the difference between business processes and business capabilities. The concept of **business processes** is briefly explained in Sect. 2.1 together with its limitations with respect to EAM.

The notion of **business capabilities** is explained in Sect. 2.2. You will learn about their differences to business processes based on hands-on examples. You will be capable of explaining the difference to other people like, in the future, your boss or your peers. You will also learn how to create a business capability map for a given

J. Jung, B. Fraunholz, *Masterclass Enterprise Architecture Management*,
https://doi.org/10.1007/978-3-030-78495-9_2

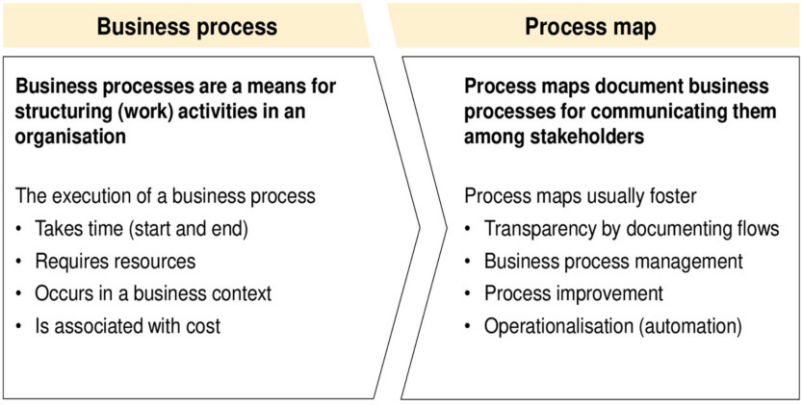

Fig. 2.1 Business process and process map

organisation. Imagine being a consultant that defines the capability map for a client. You can even imagine creating it for your current or future employer.

Next learning objective deals with identifying **business objects** for a given enterprise (cf. Sect. 2.3). Last but not least, there are many more concepts in **business architecture**, more than we will discuss in Sect. 2.4. We will provide an overview with a brief explanation of each, and you should be capable of summarising them to anybody else who will ask you in your future.

2.1 Business Process

Let us start with the concept of *business processes*. This textbook is not supposed to substitute a dedicated course on business process management, but provides a brief introduction to the notion of business processes. Later on, there will be an overview on how business processes relate to further concepts being used in business architecture.

A business process represents a series of activities that happen in an organisation or in a company (cf. Fig. 2.1). It takes time for execution, hence, it has a given start and end time. It is usually requiring resources for executing, i.e. people doing some piece of work, or machines used by people. Fully automated processes just requires machines (including computers) and no human actor. We can find processes anywhere, even outside companies in our personal environment. We will have a narrow view on business processes within this textbook: business processes are happening in a business context. Therefore, the execution of business processes is associated with costs. Whenever people are talking about business processes, and especially business process optimization, they are thinking of, how to save money.

2.1.1 Business Process Maps

We would like to have a clear distinction between a *business process* and its *description*. When using the term *business process*, we are referring to the actual work that is done in a company. Besides of this, the term *process map* stands for the (sometimes graphical) description of a process. You might have worked with process maps already or seen them in text books on business process modelling.[1] You might have seen them during an internship or any other work experience. They are also part of the definition how people should do their work in some organisations.

There is a clear distinction between the business process and the process map. The business process only refers to the actual work and is usually not visible. While looking at a shop floor in a company, you can see machines, busy people or material lying around. But you cannot see the processes themselves while they are happening there. Therefore, people are making drawings describing the process and this is what we call the process map. The purpose of the process map is its description so that other people—who don't know the process—can understand it. They can follow it like a step-by-step instruction about what to do.

Business process maps are also used for a corporate business process management, as you can only manage what you understand. And for understanding, you need some documentation which is provided on the map. One special aspect in business process management is, process improvement. Managing processes does not only mean making sure they are running, but also making sure that we are getting better with our processes over time. Process maps can also be used as a prescription for process automation. This will be supported by so called business process management systems (BPMS) or also referred to as *workflow management system*(WfMS). They require a formal description of the process and can then manage and control the execution of business processes automatically.

Business processes can be characterised by the properties provided in Fig. 2.2. First of all, the business process usually represents a **transformation**. It is not some work that is done without any reason or outcome. It always focuses on creating a result, and this result should provide a value for customers. It also requires an input. When creating a car, for example, we have the car as an output. Which also provides some value add for the customer, the person buying the car. And for producing it, the company requires materials and prefabricated parts that will then be assembled.

When dealing with business processes, and especially when keeping in mind that we want to improve processes, we are not looking at projects that only happen once but rather at activities that will be **repeated** over time very much in the same way. You can imagine drawing a business process map does not make much sense for a process that has been executed once and will never be executed again. It only makes sense if we are describing something that will also be useful in the future.

[1] Business process modelling is for example explained in [1]. We will use the term *business process map* and *business process model* as synonyms throughout this book.

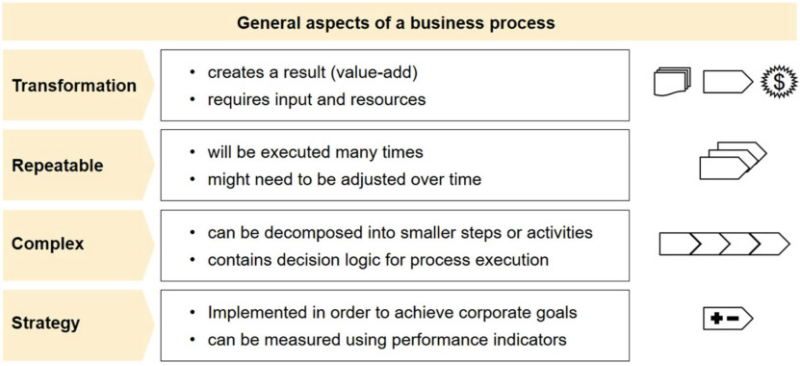

Fig. 2.2 Properties of a business process

Real life business processes usually be of some **complexity**. The complexity is often not visible, because there are many decisions to be made within the process.

Example 2.1 (Decision in Process Flow) An e-commerce company receives orders from their customers. Each order needs to be validated whether it is complete and can be executed. This can result in either *accepting* or *rejecting* the order.

Many decisions within a business process might lead to a complex control flow. Depending on decisions, one step might be done or the other. We have resources being involved. We have people being involved. We have customers and other stakeholders with certain expectations, which increases the complexity of a business process.

A business process is not just a means on its own but needs to fit to the corporate *strategy*. The car manufacturing business process only makes sense if this is part of our corporate strategy. Producing cars is not that relevant for a logistic company, because their strategy deals with managing and delivering parcels to its customers (i.e. it is different from the one of a car manufacturer). They are using cars, so therefore, they will focus on business process that will support their parcel strategy.

Those, who are familiar with business processes should already know some visualisations (i.e. business process models). Around the world, there are many different ways how to describe a business process. You can describe it with a text. You can describe it with a dedicated tool. There are standardised languages for describing. But you can also use presentation slides for putting process symbols one after the other.

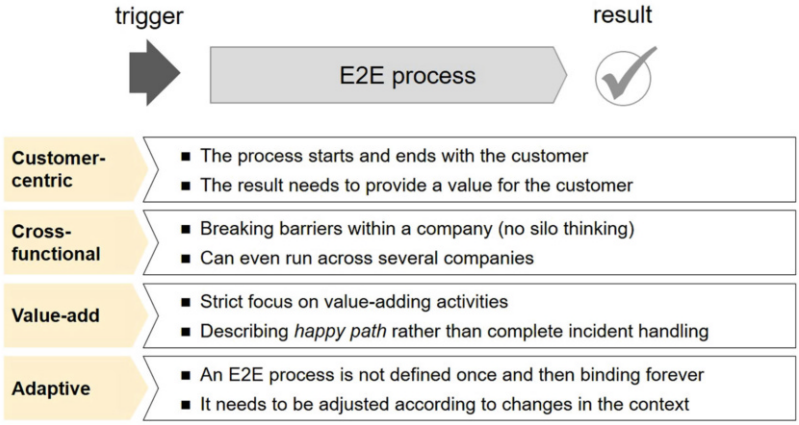

Fig. 2.3 End-to-end (E2E) process

2.1.2 End-to-End Business Process

There is a special notion of business processes and also business process maps that got quite popular, the so-called *end-to-end processes* (short: E2E process) or the end-to-end process view. Figure 2.3 shows an abstract example of an end-to-end process. An E2E process is always started by a trigger. It does not only have a time when it is started but also some event that is relevant for triggering the execution of the process. Each end-to-end process description covers any step that is required for achieving the result. In general, when creating such process maps, the basic idea is to have a **customer-centric** view. The execution is triggered by a customer—for example by placing an order. The customer is interested in getting the final result (i.e. the goods ordered). The process then covers all steps from taking the order, the order being executed, the product being delivered, and the order being paid by the customer. He does not have an interest in paying but this is part of the contract.

Whenever people are dealing with end-to-end processes, they will ask the following questions first:

1. Who is the customer?
2. What is the result expected by the customer?
3. How will the process be triggered (by the customer)?
4. What are the major steps to be executed from trigger to result?

The basic idea sounds easy, but many people still struggle with implementing processes end-to-end. Many companies have processes in place which have not been implemented by E2E principles. For sure, companies are providing products to customers. But especially large organisations have a lot of legacy processes that evolved over time following different or even no consistent principles. They appear as fragments without a clear customer perspective. There might be processes just

for individual departments or functions, but not optimising the whole value-adding chain from trigger to result. The execution of an order then requires the involvement of different parties and usually also several departments working together.

Example 2.2 (Departments in Order Processing) Executing a customer order might involve the following organisation units:

- *Customer service* for taking and validating a new order
- *Operations* for planning and scheduling the order execution
- *Manufacturing* for creating the final product
- *Logistics* for shipping the product to the customer
- *Finance* for doing invoicing and cash collection

These would be individual processes in a classical organisation (focus on division of labour). The E2E view fosters focussing on the value-add across the processes listed above.

Boundaries between departments usually hamper the execution of such a **cross-functional** business process. Individual departments just focus on their share of the work. That kind of division of labour also leads to local optimisation of individual process steps (owned by a department) instead of aiming at an overall optimisation. There is a shift in mindset with the end-to-end processes as people recognise that it is not only about your individual piece of work. It is all about the whole end-to-end process for the customer. And we need to make sure that we aim at optimising the whole process—even across departments.

As with any business process, the **value-add** is usually associated with the final result. Consequently, each steps of an E2E process needs to be value-adding by providing some value towards the final result. Non-value adding steps are not productive and, therefore, can be regarded as waste.[2] It can be observed in many process workshops during the past decades that people are not aware of the inefficiencies of their own work. We are talking about operations people who are executing the processes. Their heads are full with all the exceptions, with all the exception handling, with workarounds, what needs to be done if something goes wrong. This happens because there is no end-to-end process design and people are only focusing on their little piece of work. Many processes exist because they have been established many years ago and people are still executing them in the old-fashioned way.

[2]Eliminating waste is one of the key principles in Lean process management (cf. [2]). It aims at optimising processes by eliminating any non-value-adding activity.

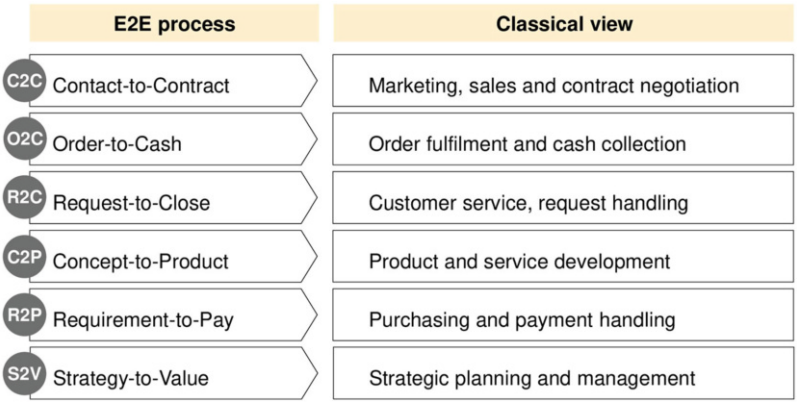

Fig. 2.4 Typical end-to-end processes

It is important to focus on the value-add during end-to-end process design. We call this the *happy path*, because it is omitting exceptions in the beginning, making sure that the whole process is stable. Examples for those exceptions are:

- missing information
- defective part
- resource unavailable

People in today's processes often spend a lot of time with handling those issues—sometimes using inefficient workarounds. Starting with a clear end-to-end process design can also help with avoiding those exceptions instead of handling them (e.g. improving data quality from the beginning). This will prevent the team for defining and executing failure handling procedures but having a process with built-in quality assurance.

Business processes and also end-to-end processes are not carved in stone. You do not just define them once, and then they will exist forever. Business processes need to be **adapted** continuously as there will be constant change. Customer expectations are changing. Competition is changing. Markets are changing. Even though we describe our business processes and business process maps for repeatable processes, we always need to keep in mind that an end-to-end process will be subject to changes over time. This will be done by business process management.

Figure 2.4 provides examples for common end-to-end processes. There are a couple of books available on the notion of end-to-end processes, including examples.[3] Hammer and Champy already presented the notion of end-to-end processes in the 1990s as part of their book on *Business Process Reenginering* [4]. Beside promoting their approach, they also describe a lot of examples from their

[3]Whittle and Myrick include some example E2E processes in a case study in their book *Enterprise Business Architecture* [3].

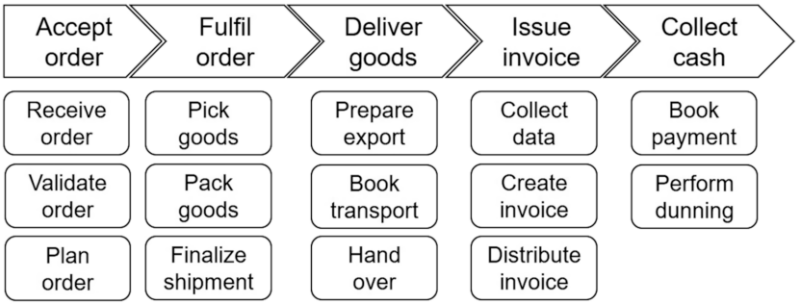

Fig. 2.5 Order-to-cash E2E process

consultancy experience. They found out-dated, complex or unproductive processes in large companies and showed their optimisation by implementing end-to-end processes. This notion has been adopted by other consultants and also researchers since then. There are some common end-to-end processes available in the literature and also in the consulting industry.

Let us have a look at Fig. 2.5, showing the **Order-to-Cash** (O2C) process (which is also listed in Fig. 2.4). It is implemented in SAP and also available in other *Enterprise Resource Planning* (ERP) systems. It is a common process that covers the execution of a customer order from the start to the end (i.e. trigger-to-result). The *Order-to-Cash* process is triggered by receiving a customer order. It then covers all the steps necessary for making sure that the order is completed. The result consists of a completed order as well as its payment by the customer. It ends with receiving the payment from the customer and then the whole end-to-end process is done. With the final result, the customer having the ordered product and also fulfilled his or her obligations with respect to paying for the product.

There are some other established end-to-end processes listed in Fig. 2.4 and also mapped to the corresponding functional view. **Contact-to-Contract** incorporates the classical marketing, sales and contract management functions. **Request-to-Close** refers to a classical customer service process which supports for example *complaint handling*, *incident management* or handling customer inquiries. The customer has a request (e.g. question or inquiry for sending information) and we need to provide the answer as a result and provide it to the customer. There are also rather long-lasting processes like **Concept-to-Product** for product development or *Concept-to-Service* in case of the service industriy. There is **Requirement-to-Pay** for covering purchasing activities and performing outgoing payments. Last but not least, **Strategy-to-Value** deals with strategies: Defining the strategy, implementing and executing it so that it can provide value for the organisation (and shareholders).

All E2E processes share the same naming convention. They start with a noun indicating the trigger (or the first object being involved), then followed by the word *to* and then the last component showing the result or the final object. The concrete

naming might defer in various organisations or text books. But they all share the same naming pattern *noun to noun* (i.e. *trigger-to-result*).

Each E2E process can be cut into several steps, as shown by the *Order-to-Cash* process in Fig. 2.5. It consists of five main steps in our case. First, we are getting the order from the customer. After receiving it, we can fulfil the order. Which means either producing something or providing another service. Our example shows a typical order fulfilment process: taking goods from the warehouse, packing them into parcels, and then finalising them for being delivered to the end-customer. The next step consists of delivering the goods (as listed in the customer order) to the customer and issuing the invoice. At the end, we need to make sure that we can receive the payment from the customer. You might have recognised already that the example captures an ordering process (so-called fulfilment process in logistics) where we have goods stored in a warehouse. A customer can order these goods. We pack them into a parcel and deliver the parcel to the customer.

Each of the main steps listed on top can be decomposed into smaller steps:

- *Accept order* starts with receiving the order from the customer. It is checked with respect to feasibility. We can check if all goods are on stock or if the order itself is complete and correct. Then we can plan how to ship the order.
- *Fulfil order* consists of somebody going into the warehouse, picking the goods out of the shelves, packing them into the parcel, printing the shipment labels or the address label, and then putting it onto the parcel.
- *Deliver goods* starts with handing over the parcel to delivery department that will then deliver all the data to the logistics company, will book a truck for the shipment, and then hand over the goods to the logistics partner.
- *Issue invoice* is done by the finance department. It collects all the data for all the orders, creates invoices for individual customers, and distributes them to the customer.
- *Collect cash* will be performed after the customer received the invoice. He or she is expected to pay for the delivery. What could happen is the customer is happy and pays, so we can recognise there has been a bank transfer. And we can book the payment as being done, so the order has been paid. Or in case the customer is not paying, we need to perform some dunning activities.

The small example is aiming at showing the following

1. Activities of a typical Order-to-Cash process for a fulfillment order. It shares some similarities with order management processes in other industries (especially the financial aspects).
2. Common way of drawing an end-to-end process by showing main steps listed from left to right and then adding detailed activities below.

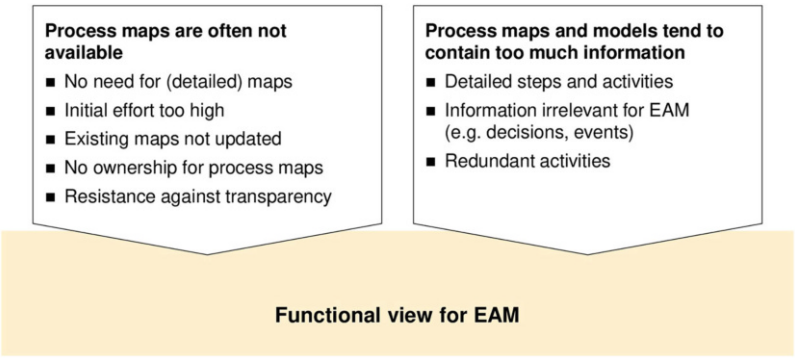

Fig. 2.6 Challenges with business processes

2.1.3 Business Processes and EAM

Business processes are a very prominent method for describing what the company is doing. Many of companies already introduced Business Process Management (BPM) by providing business process maps as a standard for managing and improving business processes. This seems to be an excellent starting point for developing the application landscape. When I started working in logistics, I got hired for introducing and supporting business process modelling. However, when starting doing my work, I was told that I am member of the business architecture team now. The job was expected to support process modelling so that the models can be handed over to the enterprise architects. The company did not have extensive business process maps and we were expected to support business experts with describing their processes in a standardised way. We also expected business people to see any value in having process maps for communication purposes.

Nevertheless, this was quite a challenging task. There were a couple of reasons why the detailed maps have not been made already before (cf. Fig. 2.6). First of all, documenting processes is a big **effort**. It is not just going around for a week and making some drawings with business experts. It rather takes a couple of months or even years having interviews with all departments, documenting the processes, and validating them.

A big challenge is then keeping process maps up-to-date. After creating a process map, you need to make sure that changes in the process are also **updated** and incorporated in the process map. Processes can change based on external factors or due to a business process optimisation. In contrast to this, the process map is only updated if somebody actively takes care of it. If you documented your process a while ago by a process map, the process will most probably work differently as of today. But if nobody updated the process maps, then it is not relevant and you cannot use it any more. This happens, especially if people do not feel responsible for maintaining the process maps.

There is always some **resistance** to change by people and they even try to avoid transparency.[4] They are afraid of showing how they are working, because other people might think that it is easier than expected. People might be afraid of being obsolete if others can do their work as well (as described in process maps). There is still some kind of silo thinking (being trapped in the own functional area) or kingdom mentality. Managers can interpret their department as their own kingdom and do not allow others to intervene. Providing process maps would allow others to understand their work and even start telling them how to do things differently. Consequently, you might not get process maps from these departments—even no support for creating them.

Even if you have a complete set of process maps, they usually contain a lot of **details** like for example:

- textual description for each individual activity
- processes decomposed into tiny activities
- formal logic for specifying decisions (e.g. conditions for evaluating a job application)
- resources required for performing a task
- information required for processing
- exception handling

Companies who documented their processes, will hand over a big book to you. This book consists of many pages with a diagrams and text. Business process maps also tend to include a lot of **redundancies** as the same kind of activity can be performed in different business processes. This might also lead to different names for the same activity, and it can be hard to identify similarities.

A less detailed (i.e. appropriate) representation of the business is required for a holistic EAM. In fact, we need a common understanding about what the company is doing and not that much how processes are executed. This also refers back to the notion of the town planner as introduced in Sect. 1.1 starting from page 2. The town planner is not interested in how the building looks like inside. He/she does not care for the colour of the walls inside or the location of the power outlets, or the switches, or for anything else inside. He or she is only looking from a high level. It is very much the same here. Business process models are deep into details and not helpful for somebody with a high level view.

Process redundancies are mentioned in Fig. 2.6 and this might look quite abstract. We prepared a small example in Fig. 2.7 in order to show typical redundancies across different processes.

[4] We will discuss the topic of change management and resistance to change in Sect. 5.1.

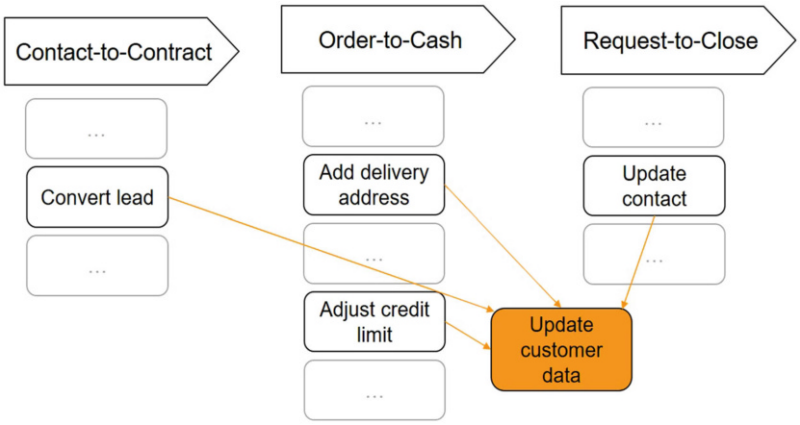

Fig. 2.7 Example: Redundancies in business processes

Example 2.3 (Redundant Activities in Processes) There is a company with various business processes. The **Contact-to-Contract** process encompasses any sales activity. One of them is a process step called **Convert lead**. Lead conversion usually refers to activities gathering and consolidating information about a prospective customer (i.e. sales lead). A company can get contact details and then enriches them with further information for future sales contacts. It also gets some idea about which kind of products to be sold to this lead. A combination of a lead and a product is called *opportunity*.

There is another process for order execution within our fictional company. We already talked about **Order-to-Cash** in more detail starting from page 41. It encompasses several process steps only two of which are shown in Fig. 2.7. There is one activity for adding the delivery address for the order (i.e. the shipping address of the parcel)–**Add delivery address**. This address might be stored in our order management system already but it sometimes needs to be updated (i.e. if the order at hand is not delivered to the standard address or the customer relocated recently).

Adjust credit limit is also done during order execution. A customer can usually place several orders, but sometimes companies put a limit for order volume or accumulated value. The company only accepts orders from one single customer until the order values sum up to 1000 Euros. If the limit is reached, then it will not accept any further order unless previous ones are paid. In other words, outstanding payments of one customer are always below 1000 Euros. Such a limit (like the 1000 Euros) is called the *credit limit*. Credit limits can be adjusted in a finance system or an order management system.

(continued)

Example 2.3 (continued)

The third process is the customer service process, called **Request-to-Close**. The process handles any inquiry made by a customer. Such an inquiry can be a complaint (because the order was not delivered completely or it was delivered to the wrong address), just a question, or a request for updating the customer account. The last example is done by the **Update contact** step shown in Fig. 2.7. The activity will update the customer account whenever the company gets aware of any changes (by phone or message). Updates can lead to a new delivery address, contact details or payment information.

Did you recognise the redundancies already? In fact, all the highlighted activities refer to the same kind of activity. Let us start on the right-hand side. Updating contact details refers to an activity with respect to updating customer data. The same holds true for **Add delivery address**. It is always an update operation on customer data. By providing a new address or an additional address, we are updating the customer data. Also, adjusting the credit limit is not done somewhere apart from the customer data. The credit limit for a customer is stored in the customer data or the data record for one specific customer. Adjusting the credit limit is updating customer data as well. Even converting leads is just an operation on a customer data record. Some CRM or ERP systems store lead data already in a customer data record. This means that adding a lead is the same as adding a new customer data record (even if it is only a prospective customer).

It is just a small example, indicating there are a couple of activities—converting leads, adding information, adjusting information, updating existing information, and then so on, which are not completely different process steps. Even though they are in different end-to-end processes, they all refer to the same kind of activity, updating customer data.

When planning a corporate application landscape for an organisation, we want to make sure that we are not buying too many software applications. Instead of having redundant applications doing the same thing we want to consolidate functionality in less systems. We, therefore, also want to avoid duplicates caused by redundant process steps. Furthermore, any kind of details provided by business processes might hamper EA as they represent unnecessary (or even misleading) input. The left hand side of Fig. 2.8 shows typical information of a business process map. It mentions detailed steps and in some companies, we even have something like work instructions or standard operating procedures. Work instructions are detailed descriptions for individuals, how to perform certain tasks in business processes. Business process maps also contain information about actors or human beings doing the work as well as their responsibilities. We have many lines showing in which order to execute process steps, how to make decisions, and how to handle loops.

However, when doing EAM, we are not the system architect and we are not implementing the system. We need to have information about systems on a high

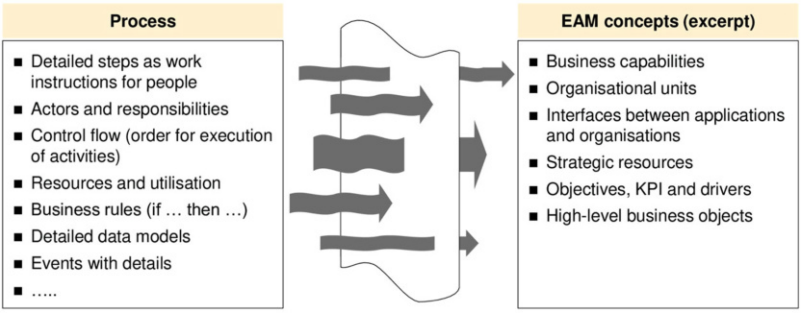

Fig. 2.8 Level of detail provided by process models compared requirements for EA

level and their mapping to business concepts. Consequently, we also need business concepts that help us with understanding the high level view on the business for then linking it to the IT landscape. An overview on those concepts is given on the right-hand side of Fig. 2.8. The list starts with the notion of **business capability** which will be further explained in the remainder of this section.

For managing software applications, we do not need individual information about users and what each user is supposed to do. We are rather thinking of organisational units or roles. We will have a look at interfaces between applications, including data exchange and service provisioning in Chap. 3. The notion of *resources* does not refer to individual resources (car, computer, machine, material) but to strategic resources. Also general business concepts are in scope for EAM. We will get an overview on *strategy*, *objective* and *key performance indicator* (KPI) for measuring if we achieved the objectives in Sect. 2.4. Section 2.3 will then introduce the concept of *business object* representing entities or concepts that are relevant for the organisation.

2.2 Business Capabilities

Let us first start with the notion of the *business capability* as defined by Definition 2.1. The definition will be key to EAM and characterises a business capability as a functional abstraction within the business architecture. It is located within the business architecture (not part of the application landscape) and it refers to **what** is the business doing. It is not an IT concept or an IT specific view on the business.

Compared to processes, it does not look that much at details describing **how** activities are performed but only look at **what** is done. A capability does not exist in isolation but has relationships to further business concepts. A capability supports the strategy and is required for achieving corporate objectives which can be measured by KPI. It always has a relation to the business model of the company and, of course, business processes.

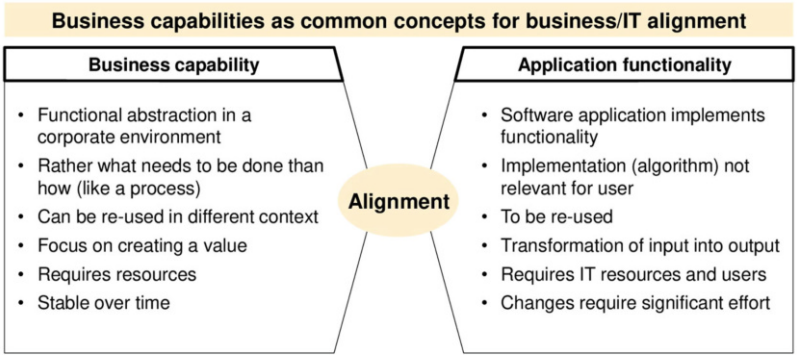

Fig. 2.9 Capabilities related to software applications

Definition 2.1 (Business Capability) A business capability (or just capability) is a functional abstraction within the business architecture and represents what the business is doing instead of providing details about how activities are performed. A capability (directly or indirectly) relates to the business model and supports achieving the business strategy.

Why are business capabilities so important when managing application landscapes for a company? Let us compare the notion of the business capability with software applications as shown in Fig. 2.9. A *business capability* is a functional abstraction in the business architecture. It presents a relevant function or some piece of work that needs to be executed in the business environment. It does not look at detail. It does not describe how activities are performed but rather represents that such an activity exists in the organisation. They are supposed to eliminate redundancies, meaning that business functions can be reused in different processes. Remember the example concerning updating customer data. This can be represented by one (small) capability and then be reused in various processes dealing with updating a customer account. Capabilities also provide a value by creating an outcome. They require resources for being executed, and they should be quite stable over time. Why stable over time? We will discuss this starting from page 50.

Comparing capabilities with software applications reveals a lot of commonalities (Fig. 2.9). Software applications are referring to some piece of **functionality** (implemented in the software). When looking at an application from an outside perspective or from a top-level perspective, it is not that important how it is implemented. We are not caring about the sorting algorithm that will sort the data to be shown in our user interface, because it is not relevant for the user and it is also not relevant for the enterprise architect as a town planner.

Applications are implemented with the objective of being **reused** in different contexts—at least in a limited way. A logistics planning software you cannot substitute an operating system (or being reused as such). Nevertheless, when creating a logistics management system, you try to implement current requirements

but also try to anticipate future use cases. A software application can offer generic features that are supposed to be reused or can have a flexible architecture[5] so that it can be extended easily in the future.

Similar to the business capability, applications **transform** input into output. Both can have input data and then store data (in a database) create new information (i.e. output calculated from the input). They can also generate required documents. Software applications require **resources** for being executed:computer hardware, other software and users interacting with the system. Software is developed for reuse because we do not want to change the application frequently. Each change takes time, requires significant **effort** and can also be a risk. The change might result in new errors and can increase the complexity of the system.

Let us take a minute and compare the list again. Try to imagine how the restriction of application functionality refers to something we can find in the business—called business capability. *Business capability* will be the core concept that we are using for aligning the business with the IT.

We would like to explain the concept of *business capabilities* by an example capability map shown in Fig. 2.10. It presents a capability map for a fictional logistics company including core business capabilities for **Logistics operations** and **Network management**. The latter one is covering managing a logistics network consisting of hubs and transportation routes. The **Customer support** capability deals with handling any claim, complaint or request made by customers. **Market development** encompasses for example gaining new markets, finding new customers and getting new customers from existing markets. Any kind of marketing activity can be reflected here.

Beside of this, we have some guiding capabilities on corporate level (Fig. 2.10, top). **Strategy development** obviously is about developing the strategy and then executing it. This is one of the capabilities associated with top executives in a company. Management people also do planning activities for the corporation (long- and short-term planning), represented by the **Corporate planning** capability. There is also a capability related to managing relationships with any kind of business partner: **Partner management**. Partners might be suppliers, strategic customers and strategic alliances. This is also located on a corporate level in this example.

Some common capabilities that you can find in any corporation are listed in the lower part of Fig. 2.10. Human resources (**HR**) consists of hiring, training, and supporting existing people working with a company. The **Finance** capability represents financial transaction management, budgeting, and year's end closing activities. Last but not least, it shows a capability for **Information management** that deals with data management, information processing, reporting and general IT functions.

This map is not supposed to provide a complete overview for one specific company. It just gives you some impression on how business capabilities can look

[5]The term *architecture* here relates to the software architecture (i.e. the house) and not the EA (town planning).

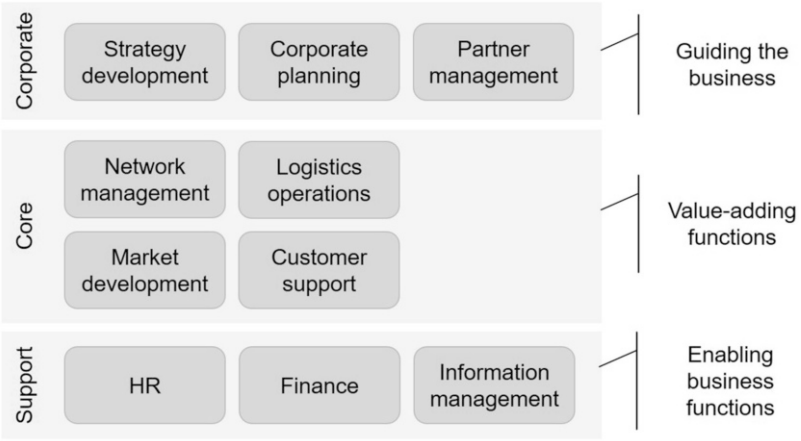

Fig. 2.10 Example capability map

like. You can see that they all have a name that refers to some kind of activity. They include a noun representing an activity like *development*, *planning* or *management*. The company is operating logistics (therefore *operations*, managing the network (*management*) and supporting the customer (*support*). Even though not all of them contain verbs, all of them have a noun that is derived by a verb for showing something is done here.[6]

The example is also using a certain kind of grouping for business capabilities.[7] This grouping is quite common for distinguishing core capabilities, from corporate and support capabilities. Core capabilities represent the core business and, therefore, are **value-adding**. They directly create value for the customers, the organisation or shareholders. They are immediately related to core processes or contribute to their value creation.

The ones on corporate level play a **guiding** role for the company. The strategy, for example, guides operations with respect to general aspects of their work. It does not provide the details, but most relevant expectations on how to execute work. Also partner management is not only about each and every business lunch or each and every event, but addressing long-term relationship also on a strategic level.

Support capabilities play an **enabling** role for the business. They do not directly create value, but are required by the value-adding functions so that they, in turn, can create value. Support capabilities enable core capabilities. The company needs personnel for doing logistics operations. It needs finance for collecting money

[6]The textbook at hand is following a consistent naming convention for processes and capabilities:
1. Process: *verb* + *noun* (example: *Check order*)
2. Capability: *noun* + *activity-noun* (example: *Order management*).

[7]The same grouping is also common for business processes.

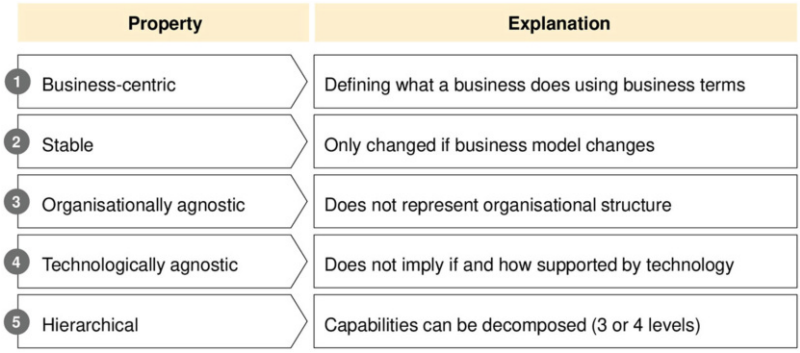

Fig. 2.11 Required properties of a capability (based on [5])

customers need to pay after delivering parcels. Information management (including IT) provides information processing systems used within the corporation.

Business capabilities are commonly used and people associate a couple of mandatory properties, as depicted in Fig. 2.11. First of all, business capabilities are **business-centric**. You might ask *Why is this mentioned in the beginning?* It seems to be obvious as we are in the context of business capabilities and business architecture. Capabilities are supposed to be a central concept for planning application landscapes with respect to business needs.

Capability maps often reflect an IT view on the company instead of the business perspective because capabilities have been evolved as a tool used by IT people for EAM. EA initiatives have been initiated by IT departments because IT wants to align IT with the business. As business process maps are not available in many organisations, creating capability maps seems to be a good starting point. It takes less effort than modelling all processes. However, IT people depend on business experts explaining the business capabilities. This support is often missing. Consequently, IT people start defining business capabilities the way they perceived the business.

This kind of map is not helpful for EAM, as we want to understand the business and then align the IT. Therefore, don't create them on your own as being an IT department. Discuss them with the business. Use business terminology, and not your terminology because its common in IT or because your software system is using this terminology. Business concepts can only be accepted by business people if you speak their language. If you start pushing your own IT or technical language onto them, they will almost immediately stop listening to you.

Second one, business capabilities need to be **stable** over time. Business capabilities are a description of what the business is doing so that we can use it as an input for creating the IT landscape for the application landscape. It should not happen that they are changing frequently as they are used as a baseline for planning the application landscape. Consequently, any change in the capability map will incur a change in plans concerning the application architecture. Planning and transforming

the application landscape is not a one-day task. It takes months, even years until it is completed. Imagine getting an overview on the business capabilities now and we plan our whole transformation roadmap. If the business suddenly informs us about new or removed capabilities, all our plans will be obsolete. And of course, if there are frequent changes, we will never achieve any objective. Therefore, for EAM, it is quite important to get the business capabilities from the business, but not as detailed so that we need to face changes frequently.

It also happens very often that people are not thinking about *what* are we doing, but rather about *who* is responsible for certain functions. We have, for example, an air freight department and we have an ocean freight department, each of them managing respective transportations. People tend to start with two kinds of capabilities as the bookings are handled differently. Considering the details, we can recognise several differences

- *Partner*: Airlines and ocean freight carriers are different kinds of business partners as they follow differing regulations.
- *Skill*: Planning ocean freight shipments requires slightly different skills compared to airlines.
- *Document*: Even though logistics documents share a similar structure across means of transport, there are specific differences
- *Process*: Processes are different due to industry regulations also by different paces of ocean vessels compared to air planes. This usually implies a different sequencing of steps.

However, we are getting distracted by the details! Let us consider the business processes for performing airfreight and ocean freight bookings. Do you think they are completely different? In fact, business experts from the industry will tell you a lot of differences because they have a solid knowledge. However, we need to focus on *what* people are doing instead of *who* and *how*.

Especially performing bookings is very much the same on a high level. Performing a booking means that we provide details about the shipment to the partner. We need to reserve capacity in a means of transport. We can then adjust the booking over time or cancel it. This can happen as cargo volumes are changing even after sending bookings to the carrier. These activities are the same on a high level of abstraction. Even though you have different organisational units, you should always check if the function is different. There might be different departments for a very good reason because they need different skills, different competencies, but we are looking at *what* they are doing. A business capability map should not recreate the organisational chart but be **organisationally agnostic**.

One of the most common mistakes we observed in our professional life is justifying the introduction of a capability because there is a corresponding IT application. This can also be a consequence of having an IT perspective on defining a capability map. This is not defining business capabilities because we want to derive the application landscape but the other way around. Such a reasoning will rather

recreate the application landscape as a capability map.[8] We need the capabilities for getting a clear picture of the business, and then derive the technology.

The business capability map needs to be **technologically agnostic**. We should create a business capability map together with business experts. It is not about recreating something we already know, but rather describing what the business is supposed to do. This will be the baseline for defining our application landscape. We can then analyse existing applications and identify typical issues like indicated by the following examples:[9]

1. An application is not supporting any of the business capabilities. This indicates that the application does not provide any business value. This happens quite often as applications are still maintained but not or only rarely used.
2. There might be redundant applications supporting the same business capability. Companies want to reduce the number of applications in order to reduce the IT budget.
3. Business capabilities that are not supported by IT applications are an indicator for missed automation potential.

The introductory example in Fig. 2.9 just showed a flat list of business capabilities. However, real life capabilities maps follow a hierarchical structure as capabilities can be decomposed in smaller capabilities. It can result in a tree having three or four levels. This will be illustrated by the following two examples.

Example 2.4 (Capability Map: Health Industry) Figure 2.12 shows an example for a business capability map from the health industry. The overall capability of the company in the health industry is **Health service management**. This is the big box around all the other ones. And then within this capability, we have **Hospital Management**, **Client Pathology**, **Client Triage**, **Patient Treatment**, **Patient Rehabilitation**, **Citizen Education Management** and **Medical Research Management**. All of them are direct sub-capabilities within the overall capability.

Each of them can be further cut into smaller business capabilities. Special kinds of patient treatments can consist of surgery (**Patient Surgery**), radiation (**Patient Radiation**), conducting exercises (**Patient Exercise**) or additional treatments (**Supplementary Treatments**). Any of them is represented by an individual sub capability within the capability patient treatment (rectangle

(continued)

[8] Recreating the organisational chart or the application landscape are very similar mistakes. They are driven by something you already know instead of focussing on a clear representation of business functions.

[9] Some more use cases for capability-based application landscape planning are provided in [6, pp. 4608].

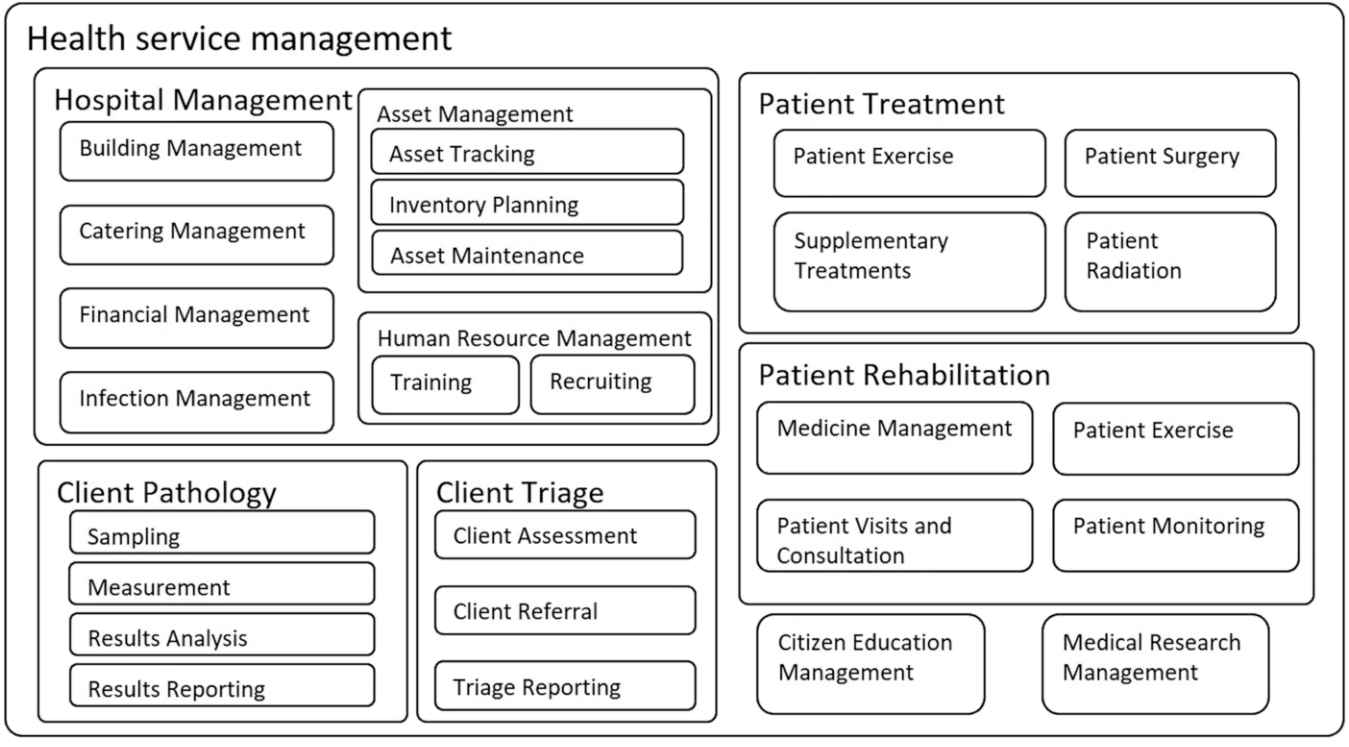

Fig. 2.12 Example capability map: health industry

Example 2.4 (continued)
within rectangle). Every capability can be decomposed into smaller capabilities, but does not necessarily has to. For example **Asset Management** consists of **Asset Tracking**, **Inventory Planning** and **Asset Maintenance**.

A few notes on creating such a capability map:

1. Each (sub-)capability needs to have a description (rule of thumb: at least two or three sentences).
2. Placing the rectangle representing one capability within the rectangle of another one, specifies that the first one is a sub-capability of another. The same holds true on any level.
3. A sub-capability can only be in one capability (i.e. it can not span several capabilities on the same level).
4. Decomposing a capability leads to disjoint sub-capabilities. There is no functional overlap between these sub-capabilities.
5. Decomposing a capability should lead to at least two sub-capabilities. **Patient Exercise** in Fig. 2.12 is violating this rule.
6. The number of levels should be limited to three or four. Otherwise, the number of capabilities gets very large and unhandy.
7. The number of levels can vary (i.e. does not need to be the same for all capabilities on the same level).

Example 2.5 (Capability Map: Logistics Industry) Another example for a hierarchical capability map is provided in Fig. 2.13, representing a parcel logistics company with the overall capability **Manage and execute parcel delivery**. This requires capabilities for **Logistics operations** as we have already seen before. It deals with managing the transports (**Transport management**) and warehousing (**Warehouse management**). Transport management, in turns, is decomposed into performing bookings, performing business transactions and offering functionality for tracking and tracing shipments.

All capabilities need to be explained. Here are a few examples:

- **Track & Trace** is one of the core capabilities of a logistics company. It helps with monitoring all shipments and is also offered as a service to the customer. You as somebody who is waiting for a parcel can check its status like the expected delivery date. Logistics companies need tracking and tracing data (i.e. track events) for controlling and managing logistics flows.

(continued)

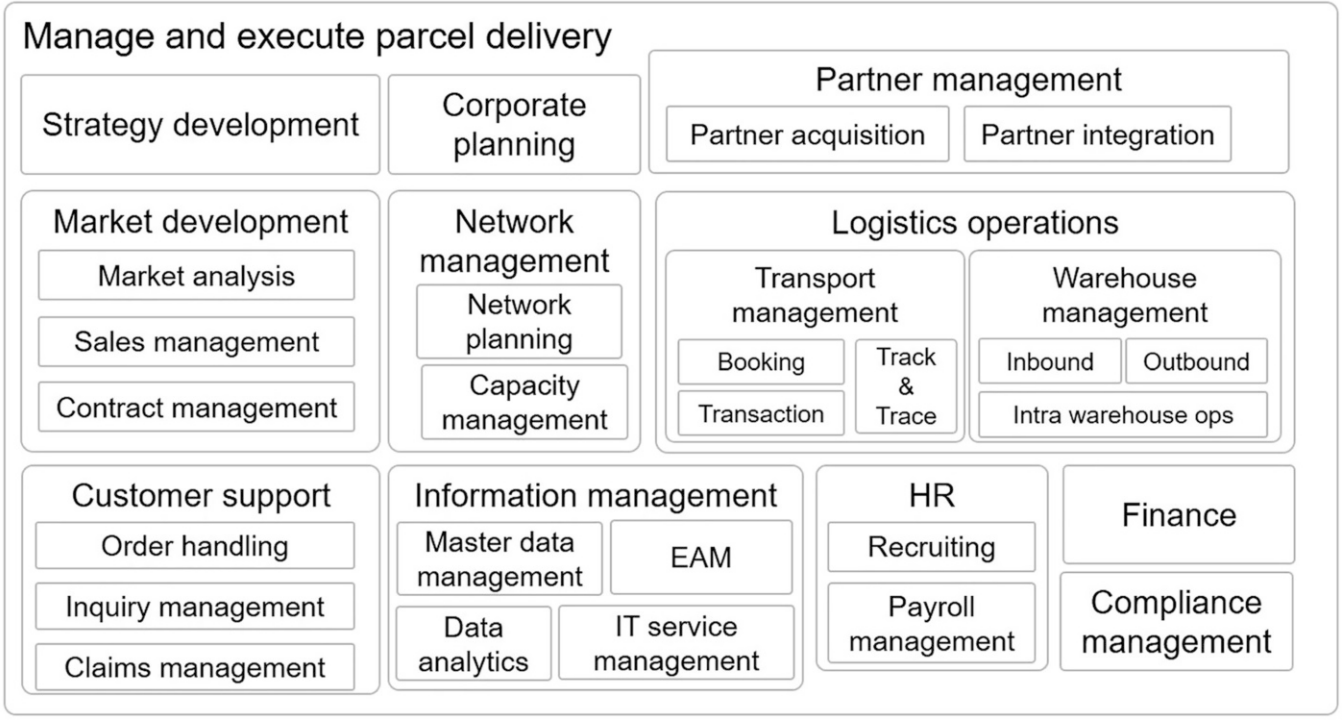

Fig. 2.13 Example capability map: parcel logistics company

Example 2.5 (continued)

- **Network planning** consists of managing the whole logistics network. A logistics network consists of hubs and transportation routes between them. Logistics panning will plan for required hubs and decide on how to set up transportation links between them. Transportation can be done by an external service provider so that this capability also includes contract negotiations with carriers.
- **Partner integration** beside the flow of physical objects, there is also a flow of information between logistics partners and government organisations. Hence, technical infrastructure for exchanging data with each individual partner needs to be set up. This includes agreements of data formats as well as protocols for data exchange.

Capabilities gained a lot of attention in recent years (cf. [6–8]). They are an excellent mechanism for describing and understanding the business on a high level. They can be the basis for implementing a governance structure and managing the application landscape.

When you are joining a company (either as a new starter or hired as a consultant) you might not have the business knowledge or an overview on the core business. Imagine that you are tasked to create a capability map for this company without proper knowledge. Where can you start from? There are a couple of sources you can use for getting a first draft as shown in Fig. 2.14.

If **business processes** are documented (i.e. business process maps are available) then you can use them as a starting point. You can check the functions used there and use them as first candidates for business capabilities. It is always a good idea to start on a high level and do investigation on detailed activities (on lower levels) later. You need to check for redundancies in the process maps and consolidate similar

Source	Description	Examples
Business processes	• Consolidation of common steps in processes • But: Risk of replicating redundancies	Customer data management
Business objects	• Identify business-relevant concepts • Determine what needs to be done with them	Customer, order, invoice
Reference architecture	• Existing capability maps as reference • Process frameworks or reference processes	PCF by APQC

Fig. 2.14 Sources for capabilities

steps from a functional point-of-view. If you just replicate the processes, there is always the risk of replicating existing redundancies.

Some methodologies use **business objects** for deriving business capabilities. They should not be confused with capabilities as discussed below. A business object refers to a relevant entity or concept within the business domain. Typical examples for business objects in a corporate environment are

- customer
- order
- invoice
- parcel

Business objects can relate to documents, physical objects, actors, or data that is being processed or used within a company. They are a pivotal element for developing a capability map according to the Business Architecture Guild in the BIZBOK, [9]. Roger Burlton provides an example for creating a capability map in the banking industry in [5]. The concept of business objects will be explained in more detail in the following Sect. 2.3. There will be a definition as well as further requirements.

There are also already **reference architectures** containing capability maps. Some people started creating reference architectures not only for an individual company but for certain industries. This kind of architecture comprises the commonalities of all companies within that industry and leave out the details. Such a capability map can then be reused for similar companies of that industry. It is called a *reference* because it is valid for being reused by other companies. The concept of reference models already exists for business processes. There is a standard available called the *Process Classification Framework* (PCF) by the *American Productivity & Quality Center* (APQC). Please, just open your browser and enter the following URL www.apqc.org/pcf. The PCF is a hierarchy of processes that is for example used for comparing different companies (by mapping individual processes to the PCF). But when looking at them and the naming, it is always referring to some kind of activity. And in fact you can reuse parts of the PCF also for defining business capabilities. To be honest, I also did it in my professional career in the following ways:

1. Getting ideas for new business capabilities
2. Checking existing capability maps with respect to completeness
3. Standardising capability maps of different corporate divisions

A quality check against the PCF can help us in identifying gaps in our own work and also identifying issues that can be eliminated.

2.3 Business Objects

In the course of this chapter, we discussed business processes and their relevance for EA. We explained why business capabilities are better suited for architecture work. Business capabilities are introduced by a definition and then further explained by examples and general requirements. Let us now elaborate further on the second core concept[10] that we will use for EAM, it's called *business object*.

Definition 2.2 (Business Object) A business object (BO) is a static abstraction in the business architecture for representing an entity or concept of the company and relates to the business model. Even though they are less detailed than a data type, business objects can be used as the basis for developing a data model in system development.

While the capability is a *functional* abstraction (i.e. referring to some activity), a business object is a *static* abstraction in the business architecture. It represents an entity, or any kind of concept that is relevant for the organisation. It also has a relationship to the business model. In fact when identifying business objects on a high level together with business people in many cases the business object model can then be a starting point for describing a data model for the company. Business objects are on a high level while data types and models for system implementation are very detailed (i.e. low level). Nevertheless if you have the broad overview on the entities and the concepts you can use this for defining your data types, refining, then handing it over to software development activities which then will develop the database for the company.

A few examples for business objects in the logistics industry are listed on the left-hand side of Fig. 2.15. We can apply the same distinction as we already did with business capabilities: corporate, core, and support objects. Core objects within logistics companies are the customer, the customer order or the transport service provider. This is another logistics company performing a part of the logistics chain on behalf of us. A very important concept is the booking. This is the order between us and other logistics companies. If a logistics company wants to move a container with an airline, they need to provide a booking to the airline. And a very common business objects that you will find in most companies is the invoice.

Please note when looking at the names that it is always about entities (e.g. documents, roles, objects) or any kind of static thing. Business objects are not representing activities. The business object *customer* only refers to our customer. It does not say what we are doing with the customer. It is the same with the customer order. It just represents the document or the data set but it does not say what we are doing with a customer order. Business objects and business capabilities are complementary as

- *Business object* represents a concept from the business that we are using
- *Business capability* represents the function being performed on a business object or having a business object as a result

[10] We will mainly use business capabilities and business objects for EAM in subsequent chapters.

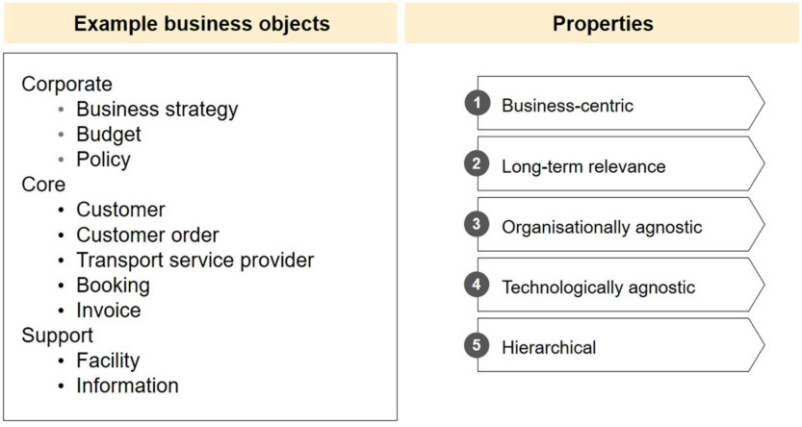

Fig. 2.15 Business objects: examples and properties

Figure 2.15 shows requirements for business objects on the right-hand side. Similar to the business capability, they need to be **business-centric**, they need to reflect the language of the business. If the business is calling the booking a *booking* then it is a booking, and it is not a transportation order or any other term. IT people tend to push their own wording towards the business as it seems to be more precise or even logical. Even though *transportation order* might be more specific as it states the fact as it is about an order, we should not use it as long it is not an established business term. We should not think like a database designer who clubs data types together for having less redundancies. In this case, we need to reflect the business language.

Business objects also need to have a **long-term relevance**—similar to the business capability. They should also be independent of any **organisation**. In the same way as the business capabilities, we should not introduce business objects just because we have **systems**. The business object should be the same no matter if the process later on will be manual or automated.

And last but not least, they can also be **hierarchical**. Each of those business objects you can cut into smaller objects. You can cut the customer into financial details, delivery addresses and sales agreements. You can have additional details for order execution. You can even have standard operating procedures just specific for customers.

Business objects can be listed and described in a text document—similar to a glossary. They can also be specified using a so-called *business object model* (BOM), which is similar to a map describing all of the business objects together with their relationships. In fact, a business object model looks like a high-level *Entity Relationship Model* (ERM). An example business object model is shown in Fig. 2.16. It contains familiar business objects that we already mentioned before. It shows, for example, the customer order (just named **Order**), **Customer**, **Booking** and the **Service providers** for transportation services.

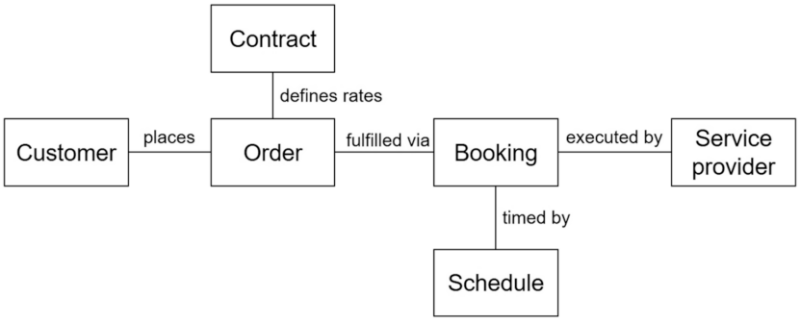

Fig. 2.16 Example business object model

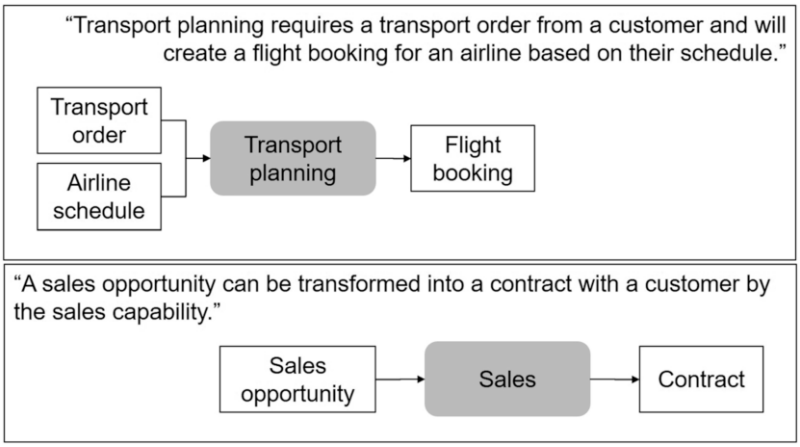

Fig. 2.17 Business objects related to capabilities

Showing business objects in such a model allows for showing relationships between them (which would not be possible in a glossary). We can, for example, specify that the rates used for the order (the price that the customer needs to pay) are defined within the contract: Relationship **defines rates** between **Order** and **Contract**. Further relationships describe that the customer who **places** the order and that orders are executed or **fulfilled via** bookings that we place with a service provider. Each booking is not just a document sent to an airline, but needs to fit to the airline's schedule: **timed by**. We can only book capacity on air planes that are really flying at the time and between the expected locations. It is just a small example for giving an impression of a business object model. You can find more examples in the literature, for example [10, pp.162] and [11, pp. 227].

Business objects and business capabilities are two different things. One is a static abstraction the other one is a functional abstraction. The first one describes the entity, the other one describes an activity. However, those are complementary, so

they perfectly fit to each other. We have two examples in Fig. 2.17 for showing the relationship between objects and capabilities. The business capability **Transport planning** requires some input and will create an output. The arrow heads at the end of the lines indicate the direction of the object flow. The output is represented by the business object **Flight booking** as this is the value created by the capability. For creating the booking, the capability requires two business objects as an input. The **Transport order** provides the details for the shipment as provided by the customer (e.g. information on cargo, origin and destination). It further requires information on available flights, represented by **Airline schedule**.

The **Sales** capability (Fig. 2.17, bottom) deals with establishing new contracts with (new or existing) customers. It starts with a **Sales opportunity** as an input. A sales opportunity is a combination of a (potential) customer and a product that a company is aiming at selling to this customer. The **Sales** can consist of smaller capabilities like customer communication, performing sales activities or contract negotiation (not shown in the example).[11] If sales is successful then there will be a **Contract** with the customer. The example is showing the transformation from opportunities to contracts by reusing existing concepts: capabilities and objects.

The examples show an important aspect of EA: The architecture does not consist of different diagrams but the information provided by them is also connected with each other. We can create a capability map and a business object model and then establish a bigger picture by linking objects with capabilities. We will follow this principle throughout this textbook. We will introduce new concepts and also set them into context by showing relationships to other concepts.

2.4 Business Architecture Concepts

Business capabilities and business objects will remain the key concepts for business architecture within the textbook at hand. This is why they have been introduced in such detail. Nevertheless, business architecture can consist of many more concepts that help with understanding the business. This section will provide a brief overview so that you will get some understanding of the possibilities of business architecture. Only a few of them will further be used in this textbook.

Figure 2.18 shows an overview on common business architecture concepts. The underlying structure is provided by an experienced EA consultant in a paper published in [13]. This book is covering various aspects on business architecture and also contains some case studies. The structure is divided into three parts.

One of them is the **Business Motivation** which is providing a strategic view on the organisation:

> Why are we doing business and what do we want to achieve?

[11] We are not considering how those activities are performed but only recognise that they exist as business capability.

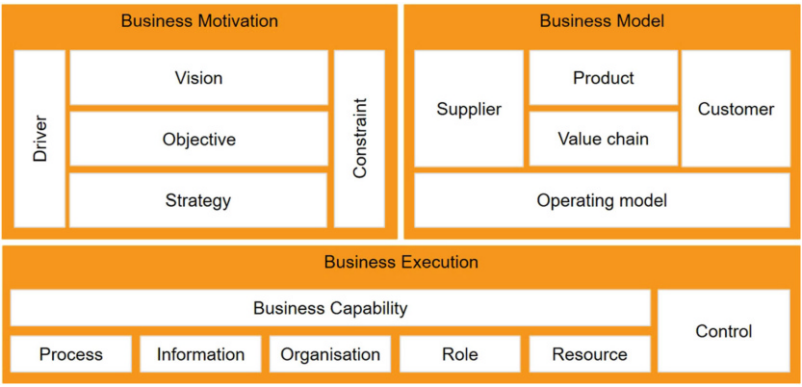

Fig. 2.18 Overview: business architecture concepts (based on [12])

Drivers cover the reasons for the motivation for starting and performing the business. They can be internal (e.g. product idea or personal motivation) or external (e.g. market opportunity or high demand). The **Vision** provides a high level description of the purpose and differentiators from other market participants. The vision is something like a leading principle for the organisation and can even be unrealistic. It can be written in an emotional way for making the company more attractive for shareholders and customers. The vision will then be the starting point for deriving **Objectives** that express concrete targets. Those targets need to be specific, measurable and time-bound so that we can determine whether they have been achieved.[12] *Key Performance Indicators* (KPI) will then be used for measuring the extent of the target achievement. The **Strategy** provides a high-level plan on how to achieve corporate objectives. It is like the company's roadmap for creating value.

Vision, objectives and the strategy reflect the internal view of our company. It is not operating in an isolated way but acting on a market so that we also need to incorporate external factors. There are certain drivers for making business, like markets, customers or the competition (especially lack of competition). A need for innovative products that are not available yet can be a special motivation. It will directly influence the creation of the vision. **Constraints** restrict the options for acting on the market. The market already has constraints but there are also legal and industry-specific regulations as well as physical restrictions. Company laws define how to perform business and industry regulations define business transactions between partners.

The **Business Model** provides the plan for transforming the strategy into value:

How do we provide value and which are our business partners?

[12]Objectives need to be SMART which stands for: specific, measurable, achievable, relevant and time-bound.

Concept	Description
Driver	Motivation for setting the vision and achieving objectives
Vision	Guiding image for the enterprise, providing direction
Objective	Desired result—specific, measurable, achievable, timed
Strategy	General course of actions and business priorities
Constraint	Political, economic, social, technological or internal

Fig. 2.19 Business motivation

It describes **Products** together with the **Value Chain** required for creating them. A value chain encompasses major process steps that will transform input resources into the final product for the customer. Products are supposed to have a value to the **Customer** (i.e. value proposition). Therefore, we also need to have a common understanding of the customers represented by customers segments. **Suppliers** provide resources (e.g. raw material, machines, services, information human resources) required in the value chain for creating the product. Strategic suppliers are key for the business model and require maintaining a long-term relationship. The **Operating Model** below describes restrictions for our business, for our infrastructure. The business model is derived from the business motivation and the elements are related to each other. The value chain, for example needs to consider the strategy and achieve corporate objectives.

Business Execution focuses on an operational view on the organisation. How do we execute our business? The business execution has a relationship to the value chain. The value chain is a high-level description of how to execute processes and the business execution refines processes so that they can be executed. **Process** performance relies on the availability of **Information**, **Roles** and **Resources**. Process execution is also embedded in an **Organisation**. **Control** mechanisms are implemented for monitoring the performance of business processes. Figure 2.18 also contains the notion of the business capability. Daniel Simon follows a slightly different interpretation of the term *Business Capability*. They see it as a bracket encapsulating the other concepts rather than restricting it to a functional view (as presented in Sect. 2.2).

Figure 2.19 summarises descriptions for business motivation concepts like the driver being the motivation for setting the vision and how to achieve the objects. We have the vision as some kind of guiding image for the company, providing directions for people within the company but also for communicating a good idea to shareholders that might fund the company in the future. We need SMART objectives that are specific, measurable, achievable, relevant and time-bound.

Concept	Description
Supplier	Partner providing resources or services
Product	Product or service with value proposition for customers
Customer	Business customer or consumer grouped by segment
Value chain	Main value adding stages from supplier to customer
Operating model	Business blueprint for implementing the value chain

Fig. 2.20 Business model

Concept	Description
Information	Information objects required for performing processes
Organisation	(Hierarchical) structure of executing business units
Role	Human actor or skill required for a specific activity
Resource	Any tool or material required for business processes
Control	Measures for controlling compliance and performance

Fig. 2.21 Business execution

We had the business model (Fig. 2.20) with suppliers referring to partners that help us with providing the service. We have the product as a description of what we are providing to the customers, including, very important, the value proposition. We need to represent the customer so that we can understand his or her needs. We can describe the high level value chains or what are the basic steps for creating the products for the customer. And we can also already have a link to end-to-end processes and then the operating model as some kind of a blueprint how to implement the value chain by using concrete resources and partners.

With business execution level (Fig. 2.21), we had concepts that you already know from IT-related classes. Information is representing data objects or any kind of information and knowledge that is required for performing processes. Organisational units build the structure of the company and execute business

processes. Roles represent skills or competencies needed for performing activities. We have resources, which can be tangible resources like physical resources, but also intangible ones like information. And last but not least, the control enables us for measuring the execution of our business, also controlling compliance, and especially the performance of our business.

2.5 Further Reading

There are some recommendations for further reading: If you want to learn more about end-to-end processes, there is a very good textbook provided by Alex Sharpe and Peter McDermott called *Workflow Modelling*. The second edition is available from 2008 [14]. Alex is a consultant in the process industry, also providing workshops and tutorials about processes. He offers training, so that people can document processes. But he also works as a consultant helping top executives with improving their processes, having a very strong focus on end-to-end process, together with the trigger and the result.

The book by Simon and Schmidt on *Business Architecture Management* is a collection of papers, all of them having some more details about business architecture and how business architecture can help with managing the IT that is within a corporation. It is a pretty good reading for getting more information [13].

The Business Architecture Guild is an organisation consisting of people interested in business architecture. They provide a handbook on business architecture as well as methods for developing it. The BIZBOK Guide [9] is available to members of the guild only and will be updated frequently. An overview on the work of the Guild is published in [15].

The Open Group published a white paper on the notion of business capabilities and how business capabilities can help with EAM[16]. It is available for free in the internet.

Reference [5] is a blog post written by Roger Burlton, a consultant specialised on business architecture. He is quite popular also in the business process community. Roger is conducting conferences and organising conferences with respect to business process management. He is a consultant for companies who want to improve their processes. A while ago, he started writing a weblog, providing his point of view on business architecture, business processes, and so on. He published a very good article on the notion of business capabilities—just a short reading. I recommend having a look at this one because Roger perfectly describes how capabilities can be linked to the business processes, what is the difference, and how do they relate to each other, and also introduces a method—how to identify your business capabilities from your business objects. He is doing it by using examples, but also has some additional hints.

	Business Object	Business Capability
Description	• Static abstraction • "There **is** something."	• Functional abstraction • "Something is **done**."
Similar concepts (business)	• Concept, entity • Examples: physical object, product document, information, resource	• Business function • Examples: transformation, activity, (process) step
Similar concepts (computer science)	• Data type (programming) • Entity type (ERM) • Class (UML)	• Function, operation, method (programming) • Use case, activity (UML)
Naming	Noun	Noun + activity (noun)

Fig. 2.22 Business capabilities and objects compared

2.6 Summary

That's it so far concerning the business architecture within this chapter. We just want to summarise the most important concepts:

- *Business processes* are a common concept for describing the business. However, business process maps might not be available or tend to be too high level.
- *Business capabilities* are a functional abstraction for describing what a company is doing. It is one of the most important concepts for EAM.
- *Business objects* are a static abstraction representing entities or concepts of the business. They are a starting point for a data model and can be linked to business capabilities.

It sometimes seems to be hard to distinguish between the two concepts *Business Object* and *Business Capability*. They tend to describe the business on a very high level of abstraction (i.e. they are abstract) and may even have a similar naming. The word *Booking* can refer to the booking document (i.e. business object) or the activity of performing a booking (i.e. capability). This also shows that a consistent and adequate naming is of paramount importance for mapping business objects and capabilities.

An overview on differences between the two concepts is provided in Fig. 2.22. Beside a short description, the figure also provides information on similar concepts in both, business and IT. Business concepts can be used as a starting point for identifying business objects or capabilities. Typical sources for business objects are documents or physical object. Business capabilities can be derived from activities or corporate functions. The rather IT-related concepts (*computer science*) already provide an outlook on which concepts objects and capabilities will be mapped to. Business objects can be a high-level representation of an enterprise-wide data model, mapped as an entity-relationship diagram. Business capabilities can be implemented as functionality in a software system or as a web service.

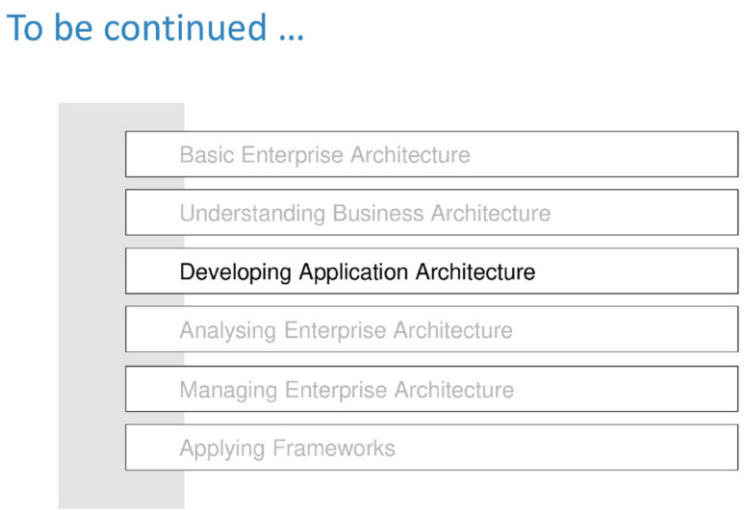

Fig. 2.23 Following next: Developing application architecture

There are many more business architecture related concepts as provided in Sect. 2.4. Most of these concepts are derived from business and management disciplines and aim at having a complete view on the company. The next Chap. 3 will continue from here and have a closer look at the application architecture (Fig. 2.23) by covering the following topics:

- How can applications and application architecture be described?
- How do applications relate to business capabilities?
- How should an ideal application architecture look like being derived from the business capabilities?

2.7 Exercises

Exercise 2.1 (Core Processes) Identify and describe at least *three* core processes of the following businesses:

1. bakery producing and selling pastries
2. postal company for international parcel logistics
3. university
4. hospital
5. restaurant offering home delivery
6. car manufacturer
7. farming company producing fruits

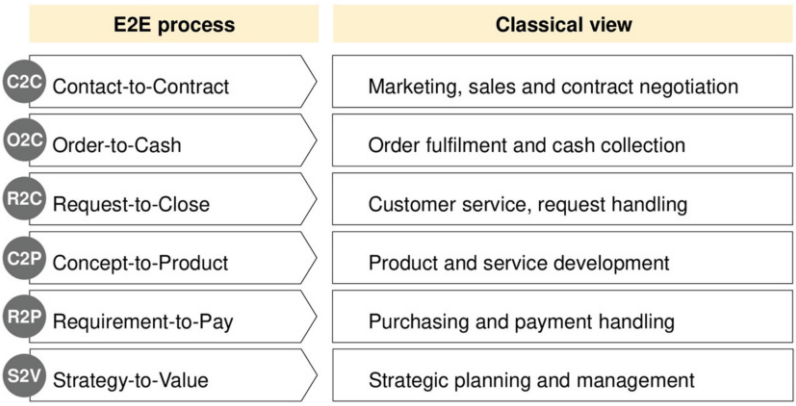

Fig. 2.24 Typical end-to-end processes

Each description should include the following information:

- result of the process
- major process steps or activities
- main resources and input

Exercise 2.2 (End-to-End Processes) Please, examine the end-to-end processes provided in Fig. 2.24 and determine for each of them:

1. *trigger* starting the process
2. *result* being the value add provided by each E2E process

Exercise 2.3 (Processes of a University) You are the Chief Process Officer (CPO) of the Deakin University and your first task is to create a high-level process map. Identify relevant end-to-end processes of the university and cover at least the following topics:

1. Defining and executing the strategy
2. Hiring staff (academic and administrative)
3. Developing new subjects for studying
4. Scheduling courses for a trimester
5. Executing a course within a trimester
6. Performing examinations

Describe each process by around five major steps.

Exercise 2.4 (Capability Map of a Logistics Company) Given is the business capability map of a parcel logistics company in Fig. 2.25.

1. Study it carefully and try to identify at least three missing capabilities.
2. Classify each capability on top level as cororate, core or support capability.

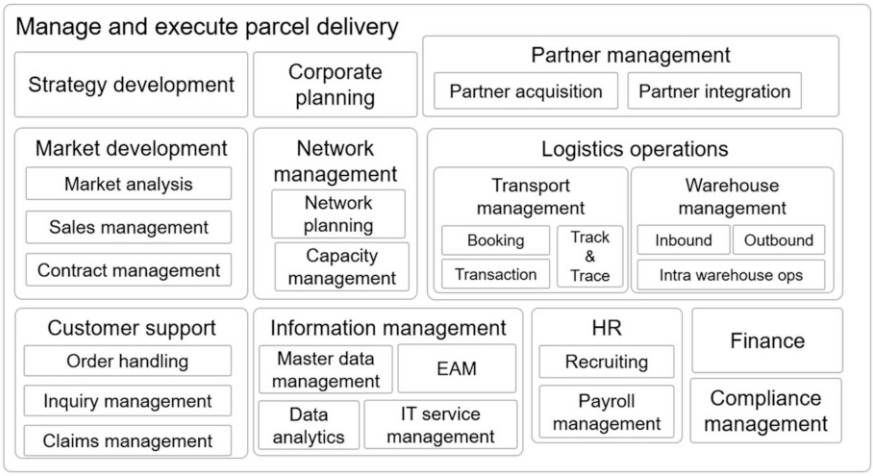

Fig. 2.25 Example capability map: parcel logistics company

3. Provide a written description for each capability based on your knowledge of the parcel logistics industry.

Exercise 2.5 (Capabilities of a University) You are the Chief Process Officer (CPO) of the Deakin University and are now tasked to create a capability map of the university. You should address the task in the following way:

1. Create a first draft of your map
2. Identify relevant sources for capabilities
3. Refine your draft based on your sources

Create a capability map with exactly two levels (meaning having several capabilities that are further decomposed into sub-capabilities).

Exercise 2.6 (Business Objects and Capabilities) Your line manager is getting confused with the concepts of *business object* and *business capability*. Please, explain in your own words major differences between the two of them. Your explanation needs to include:

1. characterisation of each term
2. examples illustrating differences
3. purpose of either of them

Exercise 2.7 (Business Objects of a University) You are the Chief Process Officer (CPO) of the Deakin University and you are tasked to create a high-level business object model. Identify relevant business objects of the university and cover at least the following domains:

1. University management and strategy
2. Managing human resources (academic and administrative)

3. Course management, studying and examinations
4. Performing research and publishing research results
5. Facility management and finance

Describe each business object properly by using complete English sentences.

Exercise 2.8 (Vision, Objectives and KPI) You are about to found a new consulting company in the supply chain business. In order to have a clear direction for your customers and employees, you should provide a vision for your business as well as objectives and corresponding KPI for determining your success.

Please, write down the following:

- *Vision* for providing guidance to your staff but also tell prospective customers about your business.

  1. What are you aiming at with your business?
  2. Why are you different from similar businesses?
  3. Which impact will your business have on the environment and society?

- *Objectives*, showing what you want to achieve within the next year
- *KPI* for measuring whether you achieved the objectives

References

1. M. Dumas, M. La Rosa, J. Mendling, H.A. Reijers, *Fundamentals of Business Process Management*, 2nd edn. (Springer Berlin, Berlin, 2018)
2. J. Liker, *The Toyota Way, 14 Management Principles from the World's Greatest Manufacturer* (McGraw-Hill Education, 2004)
3. R. Whittle, C. Myrik, *Enterprise Business Architecure, The Formal Link between Strategy and Results* (Auerbach Publications, 2005)
4. M. Hammer, J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*, updated edition, ser. Collins Business Essentials (Harper Business, 2006)
5. R. Burlton, *Developing your capability architecture: It's all about being able to get things done*, 2017. [Online]. Available: https://www.bptrends.com/essentials-of-business-architecture-developing-your-capability-architecture-its-all-about-being-able-to-get-things-done/ (visited on 03/12/2021).
6. P.A. Khosroshahi, M. Hauder, S. Volkert, F. Matthes, M. Gernegroß, Business capability maps: Current practices and use cases for enterprise architecture management, ed. by T. Bui, ser. in *Proceedings of the Annual Hawaii International Conference on System Sciences, Hawaii International Conference on System Sciences* (2018)
7. M. Wißotzki, *Capability Management Guide, Method Support for Enterprise Architectures Management* (Springer Vieweg, 2018)
8. P.A. Khosroshahi, Using business capability maps for application portfolio complexity management, Doctoral Dissertation, Technische Universiät München, München, 2018
9. Business Architecture Guild (ed.), *A Guide to the Business Architecture Body of Knowledge, BIZBOK® Guide*, V 8.5. Business Architecture Guild, 2020
10. P. Desfray, G. Raymond, *Modeling Enterprise Architecture with TOGAF, A Practical Guide Using UML and BPMN* (Morgan Kaufmann, 2014)
11. J. McGovern, S.W. Ambler, M.E. Stevens, J. Linn, V. Sharan, E.K. Jo, *A Practical Guide to Enterprise Architecture*, ser. The Coad Series (Prentice Hall, 2004)