To operate in the world of tamed agility, establish the correct extent of agility depending on the basic conditions for the project and also to effectively use the instruments presented in this book, enterprise IT needs a number of skills.

### 23.1 General Software Technology and Methodology Skills

Enterprise IT in the New School of IT obviously needs a number of software technology skills. These include:

- **Requirements management**: Dynamic requirements management plays a particularly important role in software development using tamed agility (especially for the development of software in the context of mobilization and digitalization). In addition to the technical skills [e.g., according to IREB (2016)], the ability to differentiate between essential and non-essential requirements is especially important.

- **Business analysis and modeling**: Most software systems are intended to provide commercial benefits. This means it is essential that enterprise IT understands business relationships, is able to structure them, and documents them at a suitable abstraction level. These descriptions must abstain from making any technical decisions. From a methodology perspective, this requires process and object modeling knowledge. A command of corresponding modeling languages is needed.

- **Architecture management**: The management of enterprise architectures (the architectures of entire application landscapes) is a classic component of enterprise IT. However, emergent architectures are often encountered in the context of mobile applications and the digitalization of business processes. They are difficult to plan but have to be integrated into the application landscape anyway.
This is challenging at interfaces. Even though classic architecture management models and processes that tend to count on plannability and long-term further development are not made for this, the emergent architectures nevertheless have to be managed and their further development must be planned and realized. This requires knowledge of traditional architecture management on the one hand (models, languages, processes) and, on the other hand, the insight that certain systems have to be integrated in spite of characteristics that do not “fit.”

- **Automation and digitalization techniques**: Ultimately, many IT projects are about automating processes or supporting manual activities in order to boost productivity. Numerous techniques and methods are used here, which are subject to technology changes. Knowledge of these techniques is essential. Detailed knowledge of specific techniques is required along with assessing the lifecycle of related technologies. Techniques for the synchronous and asynchronous linking of heterogeneous systems, bus systems, various types of middleware systems, and protocols to connect distributed and mobile systems are currently part of the dominant technology canon.

### 23.2 New School of IT Skills: Mobility

While the mobility of data and applications is only one dimension defining the New School of IT, it plays an important role—if only because mobility cannot be abstracted away. Development methods and instruments that encapsulate the mobility of data and applications in a layer, so that classic, non-mobile programming models are maintained, simply do not exist yet—and neither is such a solution foreseeable. Applications have to be mobile-friendly. The architect has to decide in the course of designing applications what parts of them have to be available on mobile devices, and developers have to take into account in the course of development that telecommunication connections are unreliable in practice. But this is not the only reason the mobility dimension is relevant. Mobile applications are also operated in various usage contexts and potentially have to run on many different devices, and testing mobile systems poses a number of specific challenges. In the following, we discuss the skills of enterprise IT that are required due to the increasing demand for data and application mobility.

- **Usability engineering and user experience design**: Users are familiar with mobile systems. By and large, they are accustomed to appealing and easy to use interfaces. Their experiences define their expectations of everything mobile. Appealing applications that can be learned without effort are intuitive to use and efficient are expected. This alone means the requirements for the usability of mobile applications are extremely high. But making things nice and colorful is not sufficient for usability. Usability engineering needs to be taken into account systematically in the development process. This requires usability engineers