DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Scenario-based Analysis of Collaborative Enterprise Architecture Management Tools

Nikolaus Katinszky

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Scenario-based Analysis of Collaborative Enterprise Architecture Management Tools

Szenario-basierte Analyse von kollaborativen Enterprise Architecture Management Tools

Author:	Nikolaus Katinszky
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Matheus Hauder, M. Sc.
Submission Date:	June 16, 2014

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, June 16, 2014 Nikolaus Katinszky

Abstract

Major challenges in the domain of Enterprise Architecture Management (EAM) are quickly changing enterprise environments, unclear and frequently altering requirements, and lack of communication among stakeholders. The collaborative EAM approach tries to meet these challenges by integrating collaborative aspects and simplify processes. Previous research in this field of study focused on organizational issues. Behavioral patterns and organizational structures for collaborative EAM exist. So far, however, there has been little discussion about technical assistance for collaborative EAM.

This master's thesis analyzes potential software tool support for collaborative EAM and evaluates collaborative aspects of existing EAM tools. Collaborative characteristics of EAM tools are derived. Tool functionalities that support and enable collaborative EAM are disclosed. A new tool category, called collaborative EAM tools, is identified. The theoretical findings are verified by a scenario-based analysis. Four EAM tools are evaluated and compared with respect to their functionality and collaboration capabilities.

The theoretical analysis shows that collaborative characteristics can be integrated into EAM tools to improve communication among stakeholders and reflect frequent requirement changes. Among the evaluated tools, only one solution implemented all theoretically identified collaborative functionalities. These findings indicate that collaborative tool characteristics can improve EAM performance and that these characteristics are barely implemented in today's EAM tools.

Contents

List of Figures

List of Tables

Part I.

Traditional Enterprise Architecture and Enterprise 2.0

1. Introduction

Information technology (IT) plays a decisive role today for enterprises in overcoming internal and external challenges. In recent years enterprises installed separate Enterprise Architecture (EA) departments to govern IT-related developments and manage their overall IT landscapes. Managing the EA involves aligning business with IT, ensuring IT operations to support business, and performing cost saving actions [RH13].

So called Enterprise Architecture Management (EAM) tools provide valuable support by storing information at a central location, providing capabilities to model architectures, and planning future architectural evolution [Ke10]. Thereby, they assist in coping with EAM challenges.

Unclear and changing demands, and lack of communication between stakeholders are important problems while performing EAM. A study among EA practitioners conducted by Hauder et al. [Ha13] revealed major EAM challenges in practice. Ad hoc EAM demands were perceived as challenging by 90% of the respondents. 85% of the participants agreed that unclear business goals are a challenge. EA demands unclear to the EA team are challenging for 75% of participants. Further identified challenges are, amongst others, conflicting interests among stakeholders (74% agreement), too quickly changing enterprise environments (71%), reluctant information providers (64%) and unavailable stakeholders (51%) [Ha13].

Coping with these challenges is a major subject in today's enterprise environment. In the adjacent domain of software development similar problems are successfully solved by applying collaborative and agile principles [Gl10]. Guidelines and concepts from the agile software development approach can be transferred into the EAM domain [Am14, Bu11]. Overcoming these critical challenges improves EAM efficiency dramatically: complexity is reduced, collaboration among stakeholders is improved, and quality of EAM artifacts is enhanced thanks to an appropriate level of detail and less delays [BBL12].

One concept to integrate collaborative aspects and simplify processes is called *collaborative EAM*. This approach differentiates from traditional EAM that has been practiced for years. Processes shall be streamlined to reduce complexity and delays [BBL12]. Dividing problems and solving them in continuous iterations is another object. Integrating feedback and welcoming change shall create flexibility regarding ad hoc demands, improve stakeholder interaction, and increase business alignment [BBL12]. Integrating EA-related stakeholders and fostering their participation avoids conflicting interests. Furthermore, knowledge from peers can be incorporated to take better decisions.

This existent approach is based on organizational and structural methods. Behavioral patterns and organizational structures for collaborative EAM are defined [BBL12, Le14]. Practices to streamline EA processes and iterative EA structures are available [BBL12]. Software tool support for collaborative EAM has not been examined yet. Existing collaborative structures and methods may be accompanied with EAM tools. Therefore, this thesis elaborates potential tool support for collaborative EAM.

1.1. Importance

The collaborative EAM approach promises to reduce complexity and delays, create flexibility to react quicker to environmental changes, allow advanced stakeholder interaction, increase business alignment, and improve decision taking [BBL12]. So far, technical assistance opportunities for collaborative EAM have not been examined. This work intends to contribute to existing research by analyzing potential tool support for collaborative EAM.

The new tool category of collaborative EAM tools is presented. Tools belonging to that category support in overcoming the above mentioned major EAM challenges. They can assist in the process of introducing the collaborative EAM approach in enterprises and facilitate its application. Besides fulfilling traditional tasks, collaborative EAM tools reduce process complexity and provide a platform to integrate further stakeholders, improve communication, and gather frequent feedback.

The theoretical findings are verified by a practical analysis. General functionality and collaborative characteristics of four EAM tools are evaluated. Therefore, a current state of existing EAM solutions is depicted, and recent developments in the EAM tool market are reflected.

EA practitioners can use the theoretical findings to discover how collaborative tool functionality can improve EAM processes and thereby, increase performance. The EAM tool evaluation is useful to get an overview of available market solutions. The detailed comparison of four existing products can support the EAM tool selection process in enterprises.

EAM tool vendors can use theoretical and practical analysis results to identify potential tool functionality that can be integrated in future releases. Thus, they can distinguish their product from competitors.

1.2. Problem Description

As stated earlier, one major challenge in today's EAM domain is conflicting interests among stakeholders [Ha13]. Stakeholders that represent the business entity of an enterprise might pursue different goals than IT representatives aim for. The EAM department faces a difficult decisions because of conflicting interests. By improving communication between stakeholders these conflicts might be solved. A related challenge is faced when stakeholders are unavailable. Important information might be requested but the responsible stakeholder cannot be reached. Clear structures and improved communication streams taking place on a regular basis could mitigate such problems.

Further challenges include unclear business goals and EA demands [Ha13]. When business goals are defined vaguely or insufficiently communicated, the EA department faces difficulties in aligning IT with business needs. This leads to deficient business support which results in inefficient business operations.

Too quickly changing enterprise environments and ad hoc demands are additional challenges [Ha13]. The EAM department needs to be able to adapt to unpredictable changes. If EAM is performed in a static, inflexible way, sudden changes cannot be integrated and thus, resulting EA artifacts become outdated and are not used.

The described EAM problems can be summarized as *missing communication among stakeholders*, *unclear demands*, and *changes occurring at a high pace*. The collaborative EAM approach aims at solving these problems by changing EAM structures and establishing lean processes. Collaborative EAM tools may support in coping with these challenges and thereby, increase performance. In this work it is examined how collaborative EAM tools can address these issues.

1.3. Research Scope

Collaborative characteristics of EAM tools are derived. Tool functionalities that support and enable collaborative EAM are disclosed. The new category called *collaborative EAM tools* is revealed. These theoretical findings are verified by a practical evaluation. A scenario-based evaluation approach is applied on a set of four tools. This thesis focuses on state of the art EAM tools that are available on the market today. During the scenario simulation modeling, analysis, and collaboration capabilities as well as EAM specific task support are analyzed. The results of the evaluations are summarized and compared with respect to functionality and collaboration characteristics. The tools are positioned into categories of four different dimensions.

Besides of Enterprise Architecture, the disciplines organization management, IT management and infrastructure management are touched.

1.4. Research Method and Outline

The methodological approach taken in this thesis is a mixed methodology based on the analysis of existing works and a practical evaluation of theoretically derived concepts. The thesis is composed of four parts.

Essential terms that are necessary in order to comprehend this thesis are explained, and existing research is provided in chapter 2. This includes the meaning of EA, tool support for EA and the Enterprise 2.0 approach. Theoretical findings about collaborative EAM, tool support for collaborative EAM, and a scenario-based tool evaluation approach are described in the second part including chapters 3 to 5. The evaluation results of four tools are depicted in chapters 6 and 7. The scenario evaluations are described in detail for one tool and summarized for three others. Then, the results are compared and a detailed analysis of implemented collaborative characteristics is performed. The evaluation part ends with a categorization of EAM tools. The thesis finishes with a concluding chapter where results are summarized, the relevance of the findings is discussed, and future research areas in the domain of collaborative EAM tools are revealed.

1. **Introduction**: The reader is introduced into the topic of collaborative EAM and corresponding tool support. The motivation for this work is provided by expressing current problems in the EA domain and revealing gaps in existing research.

2. **Overview**: In order to achieve a common understanding, fundamental terms of this thesis are explained. This includes a depiction of traditional EA definitions, value of EAM tools as well as existing EAM tool categories and evaluations. Moreover, the Enterprise 2.0 concept is briefly introduced.

3. **Collaborative Enterprise Architecture Management**: By presenting shortcomings of traditional EAM and characteristics of enterprises as complex systems, the importance of collaborative EAM is justified. Guidelines for collaborative EAM are explained and the meaning of lean and agile principles in the EAM domain is outlined. Furthermore, Enterprise 2.0 practices supporting EAM are depicted.
4. **Collaborative EAM tools**: Potential tool support for collaborative EAM is elaborated. The technology-to-performance chain model by Goodhue & Thompson is applied to identify a task-technology fit between Enterprise 2.0 functionality and EAM tasks. Collaborative characteristics of EAM tools are derived and the new category of collaborative EAM tools is introduced. The chapter ends with a list of available products.
5. **EAM Tool Scenario Evaluation**: A scenario-based approach to evaluate EAM tools is described. This includes a predefined information model that is imported into EAM tools, and a specification of scenarios that are being simulated.
6. **Scenario Simulation in leanIX**: The introduced information model is employed and the specified scenarios are simulated in the EAM tool leanIX. Evaluation results and snapshots of the tool are presented.
7. **Comparison of EAM Tools**: A summary of three additional EAM tool evaluations is provided. The results of in total four evaluations are analyzed with respect to general functionality and collaboration characteristics. Furthermore, the EAM tools are categorized.
8. **Conclusions and Outlook**: The thesis finishes with a brief summary of the results. Major findings are discussed and implications are described. Finally, future research areas in the field of collaborative EAM tools is presented.

2. Overview

This chapter introduces fundamentals for the remaining work. In order to guarantee a common understanding, the concept of Enterprise Architecture Management (EAM) is explained in section 2.1. Definitions from several sources are introduced and a selection of Enterprise Architecture (EA) frameworks is given. How software tools can support this task and what additional value they offer for Enterprise Architects is described in section 2.2. Moreover, categorization approaches and existing work on tool evaluations are provided. The chapter ends with a brief description of the Enterprise 2.0 approach in section 2.3.

2.1. Traditional Enterprise Architecture Management

Traditional EAM refers to the understanding of commonly accepted EAM definitions, and EAM processes and methods that have been practiced in enterprises for years. In a later section of this thesis another EAM type, called *collaborative* EAM, is introduced (cf. Chapter 3). Therefore, this strict distinction is necessary.

In the first part of this section a definition on architecture is given and explained. Based on this term further related terms like Enterprise Architecture and Enterprise Architecture Management are explained through definitions from various authors. The second part of this section is about Enterprise Architecture frameworks. The TOGAF framework and the complementing ArchiMate modeling language are introduced.

2.1.1. ISO/IEC 42010

Many different understandings and interpretations of EAM exist in science and practice. It can be noted that there is no general agreement on the definition of EAM. Schönherr [Sc09] provides a detailed literature analysis on EAM. Aier et al. [ARW08] analyze existing literature and EA practices in companies. Some definitions tend to focus on technical understanding while others concentrate on business-related contents. This section states some definitions to show different existing views. At first, the meaning of architecture in general is explained. The level of detail gets more granular when the terms EA and EAM are introduced.

Architecture in general is a term used in many different disciplines, e.g. in civil engineering, construction of houses or urban planning. In the area of information technology architecture relates to software-intensive systems and their environment. The definition of architecture according to the ISO/IEC 42010 standard is widely used [IE07]:

> **Architecture:** The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution [IE07].

According to this definition a system is made up of several components. They can interact with each other and with their environment. Architecture describes how components and their interactions are organized. Moreover, architecture states principles how the system is designed and how it evolves.

Bente et al. [BBL12] explain architecture as a structural and behavioral representation of a system and its constituent parts. Additionally, architecture provides long-term principles about the creation, evolution and operation of a system. The more complex a system becomes, the more important is accurate planning in designing, building and operating [BBL12].

The two definitions are very similar. The second one extends the ISO/IEC 42010 definition by the system operation activity. The diagram in figure 2.1 summarizes the findings. Architecture is made up of a system, and its relationship with the environment and stakeholders [IS14]. Systems fulfill purposes and stakeholders have interests in systems. An architecture is expressed through an architecture description.

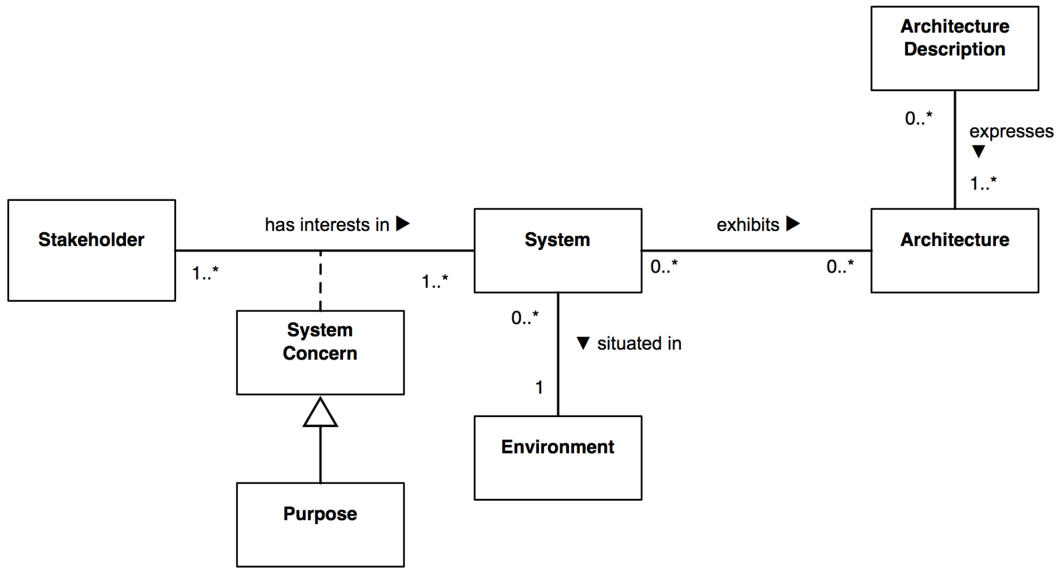

Figure 2.1.: Architecture and related elements [IS14]

A more specific architecture type is an **Enterprise Architecture**. The enterprise can be seen as a complex system which operates in a business environment. In this complex system many different stakeholders are involved that pursue different goals and interests: executives, business functions, IT employees, suppliers, and others. A formal definition of the term EA is provided by Kaisler et al. [KAV05]:

> **Enterprise Architecture:** An enterprise architecture (EA) identifies the main components of the organization, its information systems, the ways in which these components work together in order to achieve defined business objectives, and the way in which the information systems support the business processes of the organization [KAV05].

According to Ross et al. [RWR06], EA is an organizational structure that is responsible for business processes as well as IT infrastructure and reflects integration and standardization requirements of one company's operating model. It offers a long-term view of a company's processes, systems and technologies. Besides of fulfilling immediate needs, this long-term vision allows to create capabilities through individual projects [RWR06]. This description emphasizes alignment between business and IT as well as long-term focus.

Bente et al. [BBL12] define EA as a *'representation of the structure and behavior of an enterprise's IT landscape in relation to its business environment'* [BBL12]. Current and future use of IT in the enterprise are considered and a roadmap describes a path to achieve a future state.

Collectively, these definitions outline the following EA characteristics:

- structural and behavioral representation of business processes, systems and technologies,
- business and IT alignment,
- component integration and standardization,
- fulfillment of present needs,
- long-term evolution of enterprise IT and its environment, and
- creation of new capabilities through project-based activities.

Moreover, there are conceptual models, so called *information models*, that represent concepts, relationships, constraints, rules, and operations for a specified domain [Le99]. In the domain of EA, an information model aims at establishing an enterprise-wide consistent terminology in order to simplify communication between stakeholders and provide information about the EA [Bu10]. It is often referred to as *meta model*. It reflects an agreed conceptualization of the EA or parts thereof. Only relevant areas that meet the information needs are covered because gathering required information causes expenses. Consequently, an EA and its corresponding information model provide a holistic view of the entire enterprise [Bu12].

Based on findings from Buckl et al. [Bu10], Buschle et al. [Bu12] describe this holistic view that consists of vertical layers and cross-cutting aspects (cf. Figure 2.2).

The vertical layers represent in a bottom-up order infrastructure, application and business layers that are interconnected. *Infrastructure elements* are hardware devices like servers, networks or routers. They are provided as *infrastructure services* to the *application and information* layer. On this layer applications use infrastructure services and available information to realize *business services*. Business services are applied in *business processes* on the business layer. They can be reused and rearranged to create new business processes. The *organization* as a structural unit executes these business processes in an efficient way and therefore builds core competencies for the enterprise known as *business capabilities* [Bu12].

Besides of horizontal layers, there are cross-cutting elements that influence all layers. A *vision* that describes the enterprise's aspired target state determines *goals* for specific objects that shall be achieved. These goals can be measured by *Key Performance Indicators (KPIs)* that answer *questions* about the level of achievement. *Strategies* describe an ordered set of activities that are pursued to achieve the desired target state, the EA Vision [Bu10]. *Projects* should be aligned with the EA strategy. They are a *'planned and organized set of activities for*

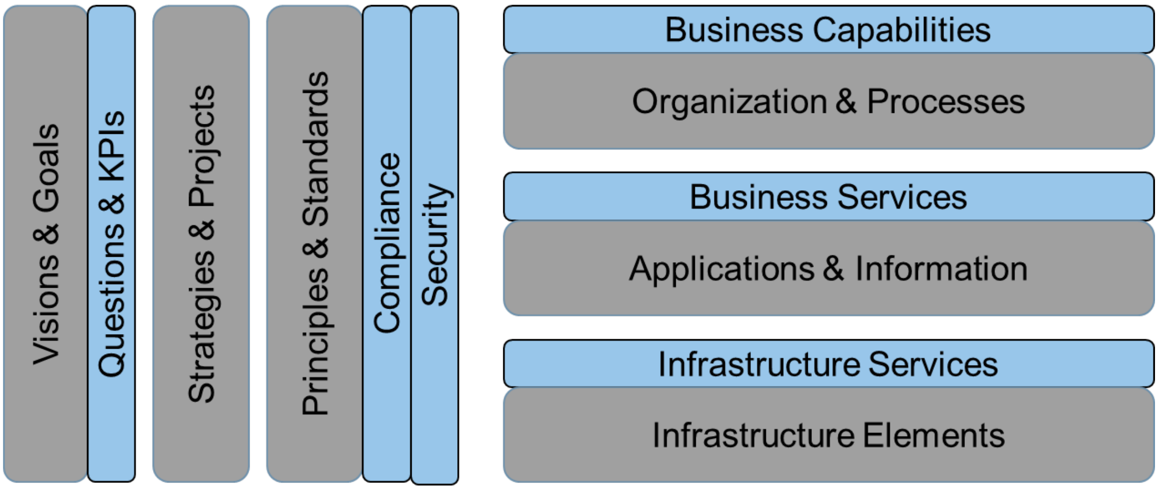

Figure 2.2.: Holistic view on Enterprise Architecture [Bu12]

realizing a specific, clearly defined one-time goal' [Ho09] and contribute to goals. *Principles* state enterprise-wide guidelines and can justify decisions about EA design. *Standards* are based on principles and best practices. They are *'pre-defined design norms'* [Ho09] and describe how EA principles can be applied on the information model. Moreover, there are factors brought in from the enterprise's environment, e.g. *compliance* and *security*, that influence EA.

Elements of the aforementioned holistic view are created, managed, and used as a structured approach in the IT Governance function **Enterprise Architecture Management** [BBL12]. EAM typically interacts with other functions, e.g. requirements management, IT project portfolio management, service portfolio management, release management, change management or applications development [Ke10]. It describes activities that are performed for the establishment and continuous development of EA [AGW11]. Typical EAM activities are:

- defining the IT strategy,
- modeling architectures,
- evolving the IT landscape,
- assessing and developing capabilities,
- developing and enforcing standards and guidelines,
- monitoring the project portfolio,
- leading or coaching projects and
- managing risks involved in IT [BBL12].

2.1.2. Enterprise Architecture Frameworks

EA frameworks can be regarded as a *'set of assumptions, concepts, values, and practices'* [BBL12] that facilitate a model view on real enterprises. They provide a fundamental structure for

EAM [BBL12]. Additionally, they assist in organizing enterprise IT and offer best practices and established techniques. In this way, they support enterprise architects with abstract guidelines and principles that help in developing, maintaining and planning EA.

In this section TOGAF and ArchiMate are introduced. TOGAF is assumed to be the mostly used framework [Ke12], while ArchiMate is a modeling language that is often used in combination with TOGAF. The TOGAF framework is a fundamental part of this thesis as it is utilized in combination with agile principles in section 3.4.4.

Further popular frameworks are the Zachman framework [Za14] or the Federal Enterprise Architecture Framework (FEAF) of the U.S. Federal Government [Wh14]. A comprehensive list of available frameworks is developed by Matthes [Ma11] who describes 50 EA frameworks in detail and summarizes additional 45.

TOGAF

TOGAF stands for *The Open Group Architecture Framework* and is developed by EA experts within The Open Group consortium. The first version of TOGAF was published in 1995 and is based on the Technical Architecture Framework for Information Management (TAFIM) [Th14a]. The latest version is TOGAF version 9.1. It is publicly available and free for use. The TOGAF documentation [Th14a] is divided into 6 modules:

- Architecture Development Method (ADM)
- ADM Guidelines and Techniques
- Architecture Content Framework
- Enterprise Continuum & Tools
- TOGAF Reference Models
- Architecture Capability Framework

The following descriptions are based on the TOGAF 9.1 documentation [Th14a]. Further details about the structure of the framework and specific processes are depicted in this document.

The **Architecture Development Method** (ADM) describes a process for planning, designing, realizing, and governing an EA lifecycle [BBL12]. The ADM process is presented in figure 2.3. It is the core element of TOGAF and has similarities with the waterfall software development model that consists of several phases [BBL12]. A main difference is that the ADM is a continuous iterative process where the scope and level of detail can change with each iteration. It is also possible to jump back and repeat earlier phases. In this way, ADM provides flexibility and adaptability. Each ADM phase is described in the TOGAF documentation through *objectives*, *approach*, *inputs*, *steps* and *outputs*.

In the **Preliminary Phase** (cf. Figure 2.3) the enterprise's architecture capability is determined and established. In other words, preparations are performed in order to establish a solid foundation inside the enterprise.

In the **Architecture Vision** phase a high-level vision of the to-be EA is developed. Capabilities and expected business value are being clarified. A resulting document is the

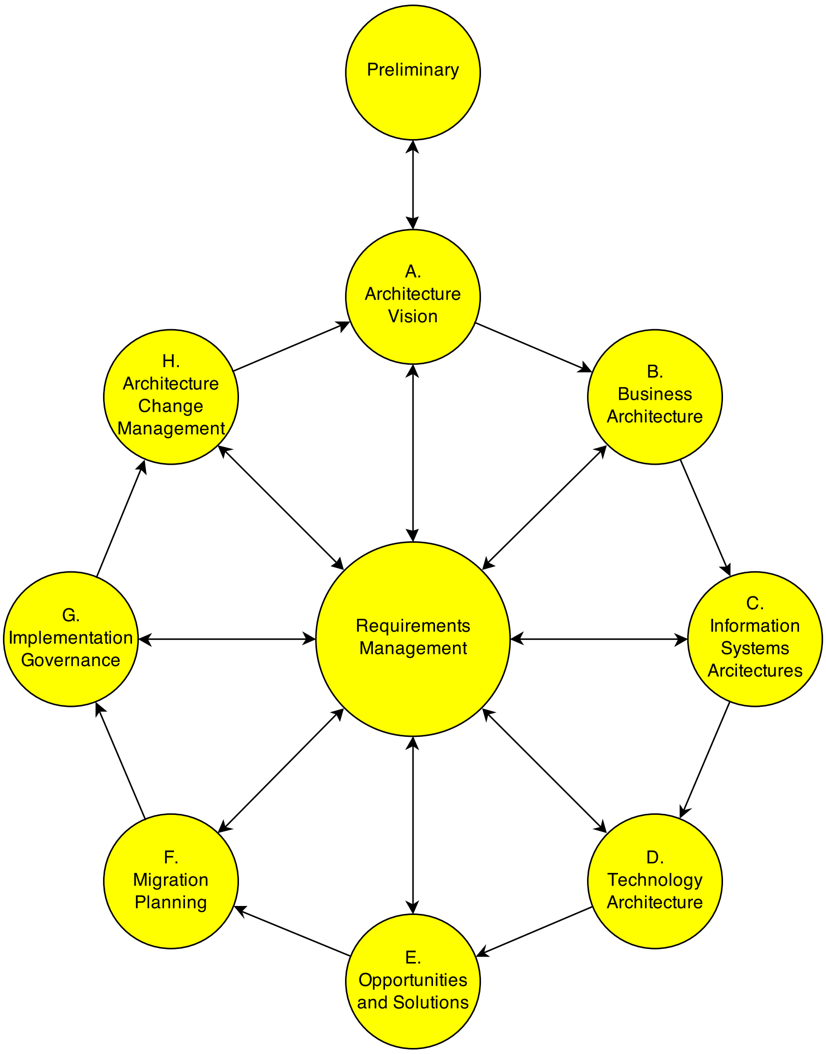

Figure 2.3.: TOGAF Architecture Development Method (ADM) (adapted from [Th14a])

Statement of Architecture Work that includes project description and scope, overview of the elaborated architectural vision, and project plan and schedule.

In the following three phases, **Business Architecture**, **Information Systems Architecture** and **Technology Architecture**, the target architectures for the corresponding layers are developed based on the architectural vision and the overlying layers.

In the **Opportunities and Solutions** phase the beforehand created layered architectures are combined to one complete version, the Architecture Roadmap. Moreover, it is checked if an incremental approach is necessary. If so, further transition architectures are defined that ensure business operations.

The **Migration Planning** phase finalizes the Architecture Roadmap and sets up an Implementation and Migration plan. Additionally, the overall approach, its business value and transition architectures are explained to key stakeholders.

Architecture Roadmap or transition architectures are implemented in the **Implementation and Governance** phase. It needs to be verified that the implementation conforms to the planned architecture. Moreover, particular governance functions are applied.

The **Architecture Change Management** phase ensures that the architecture life-cycle is maintained, the Architecture Governance Framework is executed and architecture capabilities meet the original requirements.

Requirements Management plays a central role in the TOGAF ADM process and is integrated in all phases. Architecture requirements can be identified and integrated into the process in any ADM phase. This deep alignment of requirements management allows to react flexible to new requirements and ensures that the developed architecture always conforms with actual needs.

Besides the ADM process, the **ADM Guidelines and Techniques** module provides architectural principles, patterns, and advice on how to adapt TOGAF to particular usage scenarios. It is explained how iterative concepts can be applied in the ADM, specific application scenarios can be integrated, e.g. in the area of security or service-oriented architectures (SOA), or how requirements can be derived and gap analysis can be performed.

The TOGAF module **Architecture Content Framework** defines a structural model for architectural work products that are created during the ADM process. It defines three types of work products (*deliverable*, *artifact* and *building block*) and brings a content metamodel that illustrates how building blocks in the architecture are defined and related to each other. The Content Framework offers an underlying structure that complements the ADM. It standardizes inputs and outputs and describes '*what the architecture should look like when it is done*' [Th14a]. The application of this content framework is not mandatory. Other frameworks, e.g. the Zachman Framework, can also be applied instead. One advantage of the the Architecture Content Framework is that there is a clear map between the ADM processes and the underlying content framework [BBL12].

Usually, it is not possible to create a unified EA that meets all requirements. Architectures follow different purposes and are related to each other. Depending on the stakeholder, different architectural views can be useful. Architectural views can e.g. differ in the level of detail, described time frame or technology and business detail. The TOGAF part **Enterprise Continuum & Tools** explains how EA can be split up, managed and organized within a repository.

The module **Reference Models** describes two reference models that can be used within TOGAF, the *Technical Reference Model* and the *Integrated Information Infrastructure Reference Model*.

Materials and guidelines how to establish appropriate organization structures, roles, processes and architecture governance functions within an enterprise are provided by the **Architecture Capability Framework**.

At this point it shall be noted that not every described TOGAF part needs to be implemented by an enterprise. TOGAF acts like a toolkit where individual parts can be selected. It can guide enterprise architects and give them a direction to evolve EAM.

ArchiMate

ArchiMate is an open EA modeling language developed by The Open Group. The current version is ArchiMate 2.1. It provides tools to describe, analyze, and visualize EA relationships. It is possible to model the EA, its motivation, programs, projects and migration paths that need to be executed to achieve the target EA [Th14c]. ArchiMate is compatible with the TOGAF ADM process. These two Open Group projects complement each other and can be applied as a combination.

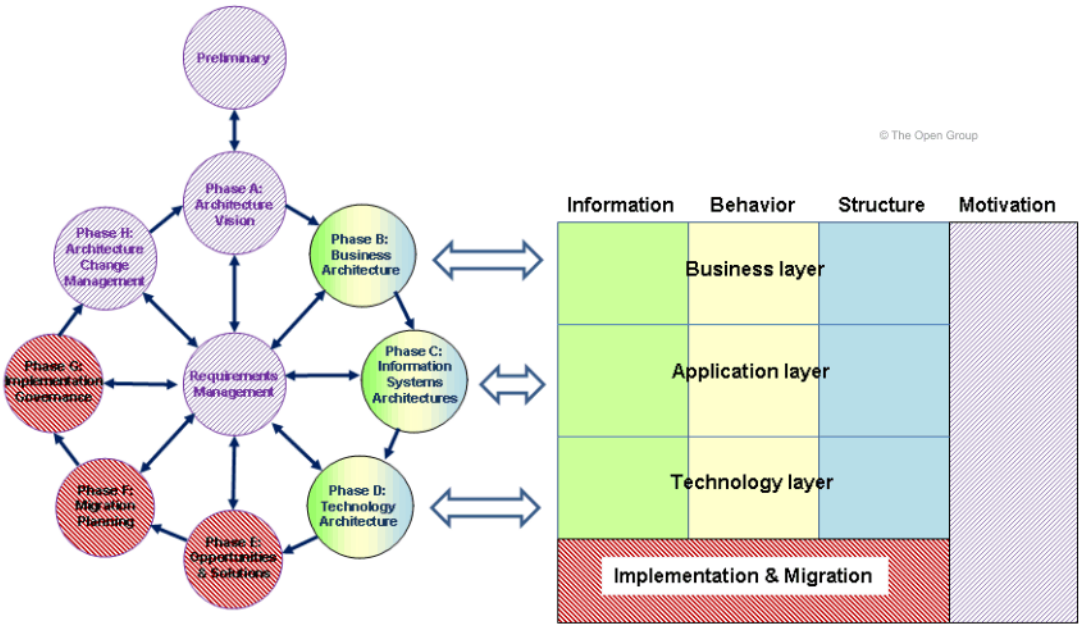

Figure 2.4.: Correspondence between ArchiMate and TOGAF [Th14b]

The following descriptions are based on the ArchiMate 2.1 specification [Th14b]. This specification explains the ArchiMate structure, objects and visualizations in more detail. Figure 2.4 illustrates the alignment between TOGAF and ArchiMate. ArchiMate defines three architectural layers, business, application and technology layer, that can be linked to the ADM process phases where the corresponding architectures are developed. In addition, three aspects are regarded in ArchiMate:

- *active structure aspects* where structural concepts like business actors, applications or infrastructure elements are represented that perform activities;
- *behavior aspects* that express how activities are executed by actors (e.g. processes or services), and
- *passive structure aspects* that illustrate objects on which behavioral activities are performed on, usually information or data objects.

After the ADM architecture planning phases the implementation and migration phases are supported by the *Implementation & Migration* extension. ArchiMate's *Motivation* extension supports TOGAF ADM's architectural vision, requirements engineering and maintenance phases. Through these extensions it is possible to model requirement changes quickly and adapt changed business needs to the EA modeling process.

The three architectural layers are linked to each other via relationships, e.g. business elements can make use of application services or infrastructure services can be used by application components. Business objects can also be directly realized by technical components which means business and technology layer can be connected directly and the application layer can be skipped.

In order to allow different representations of the same model ArchiMate realizes views through the concept of *viewpoints*. Stakeholders can define their own viewpoints where only the content is visualized they are interested in. The complex overall EA model is abstracted to the stakeholder's relevant information. In this manner different perspectives of the same information model can be illustrated. Depending on the purpose of a view, different viewpoints are useful. The viewpoints that are used for design activities differ from the ones that are applied for decision support activities. Some viewpoints illustrate content of a single layer while others focus on relations in between. There is a huge amount of predefined viewpoints provided by the ArchiMate documentation. Examples are *Business Process viewpoints*, *Application Usage viewpoints* or *Layered viewpoints*. The latter one visualizes all three architectural layers and allows to identify and model relations of objects between various layers. Figure 2.5 displays an exemplary Application Usage viewpoint where application components realize application services that are used by business processes.

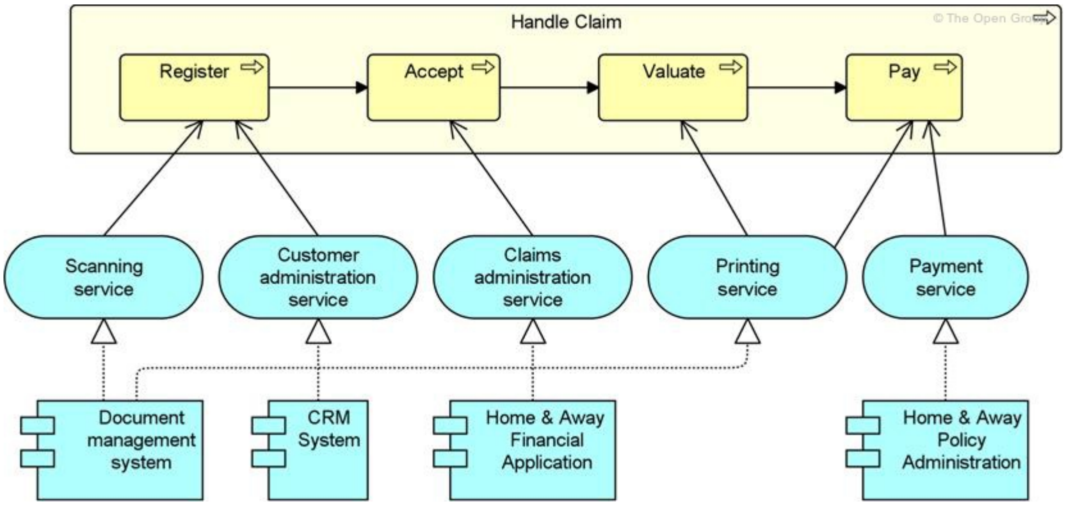

Figure 2.5.: ArchiMate application usage viewpoint [Th14b]

The ArchiMate language focuses on core concepts, objects and relationships of enterprise modeling that can be used for general purposes and use cases. Besides, ArchiMate can be extended to support specific domains. Two extensions, the Motivation extension and the Implementation and Migration extension, are already predefined in the official ArchiMate design document. Apart from that, ArchiMate can be extended by every user to meet particular needs. Attributes to existing object types and relationships can be added and new concepts can be defined by specializing existing ones.

In conclusion, ArchiMate is quite popular in practice because it is flexible and can be aligned with existing frameworks. Moreover, the ArchiMate notation is supported in many EAM tools which eases adaption.

2.2. Tool Support for Enterprise Architecture Management

In this section EAM support through software tools is described. Provided value of EAM tools is explained in section 2.2.1. Existing categories that EAM tools can be divided into are introduced in section 2.2.2. Afterwards, existing literature on EAM tool evaluations is presented in section 2.2.3.

2.2.1. Value of EAM tools

The EAM tasks presented in section 2.1 require lots of manual effort to be fulfilled. Lots of models, documents and other outputs are created in EAM processes. The fastest way to compose these artifacts might be to draw and note them manually on paper sheets. Some disadvantages of this proceeding are that artifacts cannot be reused, corrections are difficult to integrate and outputs can only be accessed locally. These and many other disadvantages outweigh by far the benefit of quick handwritten composition.

Another way of dealing with EAM artifacts is to create them with traditional word processing and spreadsheet software like Microsoft Office. EAM is performed this way in many enterprises, especially in these where EAM is a rather new topic [Ke10]. Application information, process documentations and other EAM knowledge are managed and locally stored in files on many different devices. If this knowledge is shared, files are usually send via email to co-workers or information is printed on paper sheets. This approach leads to many disconnected information silos in the enterprise [Ke10]. Information becomes inaccessible which leads to information gaps. The current landscape state cannot be modeled because it is impossible to gather all required information. Artifacts become outdated or redundant and, therefore, unusable [BBL12]. The EAM process becomes complicated. EAM analysis cannot be performed because of missing information and, eventually, the aforementioned EAM tasks cannot be fulfilled sufficiently.

Storing all required and important EAM information in one central enterprise-wide repository can solve these problems. This is where specialized EAM tools come into play. The central approach makes information accessible for all relevant stakeholders. The repository can be seen as one single source of truth where the latest and most accurate versions of all EA artifacts exist [Ke10]. It integrates information from many enterprise-wide systems. Redundancy is eliminated and documents of different architectural layers that were created by various stakeholders can be linked. The centrally stored information can be used to visualize landscapes with all architectural layers. Analysis taking into account factors from business, application and infrastructure layer can be performed to improve EAM performance [Ke10]. Software tools that support these central approach help in fulfilling EAM tasks and support EAM better than the previously mentioned approaches.

The three described information storage approaches are illustrated in figure 2.6.

Besides of centralized information access EAM tools can offer additional features. EAM tools bring either fixed or adaptable information models. In the first case enterprises need to align their already existent information model with the predefined tool model. In the latter case the tool can be adapted to reflect the enterprise's situation. Based on the underlying information model, views visualizing architectural layers for particular stakeholders can be created and total views of the overall landscape can be modeled [Ke10]. Another option is that views are automatically generated based on integrated data.

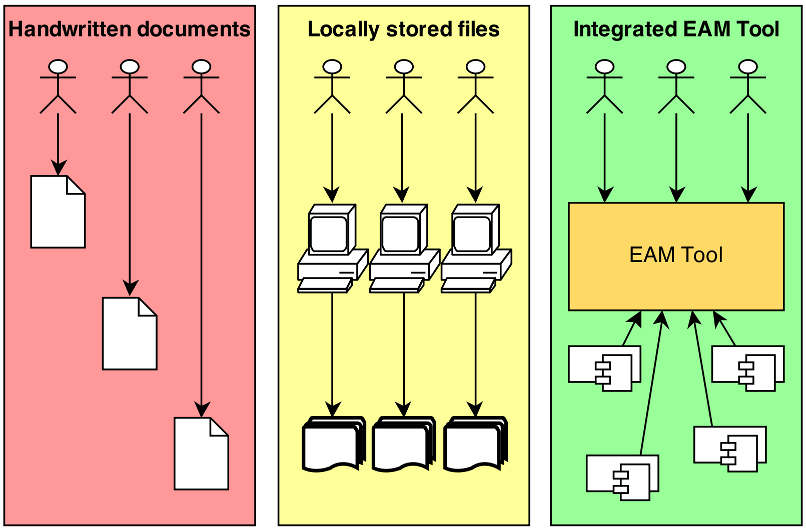

Figure 2.6.: Different approaches of artifact storage

Exporting these views so that they can be used in other environments is a useful feature. A time dimension can be considered and future states can be planned and simulated [Ke10]. Reporting capabilities aggregate EA data and represent important information. Quantitative analysis depending on various factors and KPIs can be performed to identify architecture improvement potentials [BBL12]. As EAM is a cross-functional task, EAM tools integrate capabilities to support activities that are linked with EA. Potential integrated capabilities are requirements management, project portfolio management, or business process management [Ke12]. Table 2.1 summarizes possible features of EAM tools.

Feature
Centralized repository with interfaces to other enterprise software
Fixed or adaptable information model
EA modeling capabilities
Automatically or manually generated visualizations for different stakeholders
Planning and simulation of EA evolution over time
Reporting capabilities and quantitative analysis
Support for business process management, requirements management, project portfolio, or other connected disciplines

Table 2.1.: Possible Features of EAM tools (adapted from [Ke10, BBL12, Ke12])

Covering all described features in one EAM tool solution is very difficult. Usually, EAM tool vendors focus on some features to meet specific market needs. In the following section

EAM tool categories are presented.

2.2.2. EAM Tool Categories

As mentioned before, EAM tool vendors specify on selected features to distinguish their products from competitors. Thus, EAM tools pursue different approaches to implement some features more detailed than others. In order to compare EAM tools and describe them according to their key characteristics a categorization is useful. A few attempts have been made in previous research to categorize EAM tools.

Lankhorst [La05] identifies six categories of tools that can be used for EAM (cf. Figure 2.7). *EA modelling tools* consider all architectural layers and domains and support modeling activities on an abstract level. *IT management tools* are used for managing enterprise IT assets. *Software design and development tools* describe modeling tools originating from software development where EA modeling capabilities and diagram types were added. *Business process design tools* are business process modeling tools that were extended by high-level EA and IT-related concepts. *Business process management tools* are designed for operational management of business process management. Lastly, there is the category *repositories* which includes metadata repositories and IT management tools that cover to some extent EA modeling and analysis capabilities.

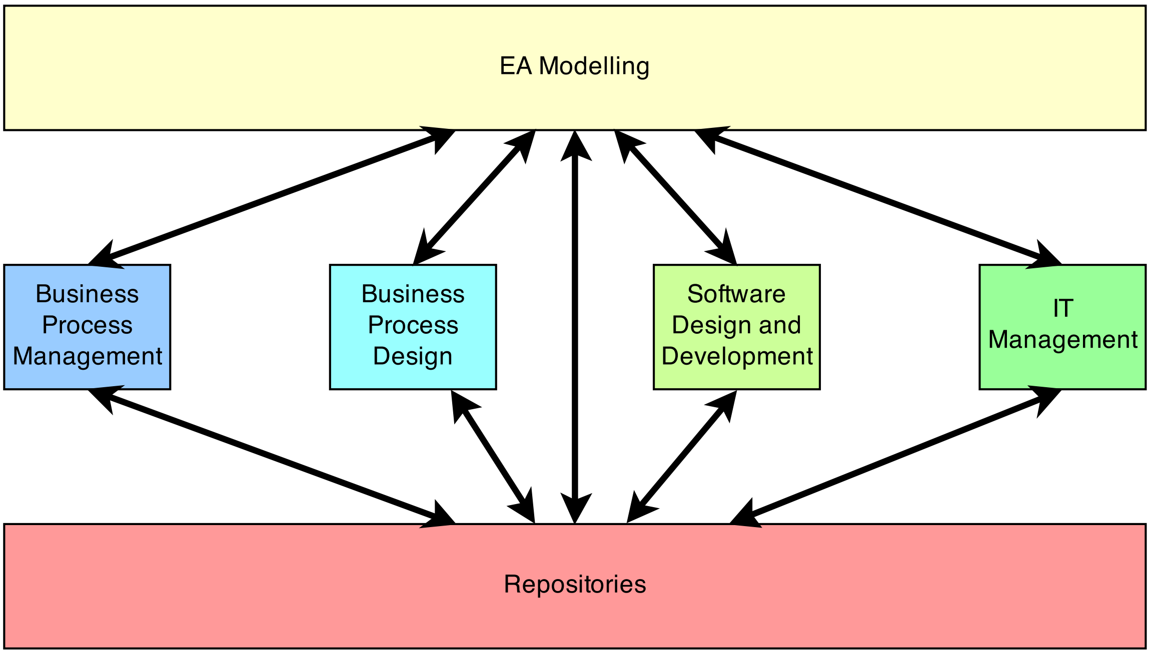

Figure 2.7.: EAM tool categories according to Lankhorst [La05]

Keuntje et al. [Ke10] distinguish between three EAM dimensions where each dimension contains multiple categories (cf. Table 2.2). There are some loose boundaries between categories. Therefore some tools cannot be positioned into a single category but can belong to multiple categories.

The first dimension, degree of flexibility, distinguishes between three categories. EAM

Dimension	Category
Degree of flexibility	Information model-centered
	Method-centered
	Process-centered
Customization effort	EAM off the shelf solution
	EAM platform
Integration approach	Automatic integration
	Manual import

Table 2.2.: EAM tool categories according to Keuntje et al. (adapted from [Ke10])

tools that pursue an *information model-centered* approach allow lots of flexibility and enterprise specific configurations. The information model offers maximum flexibility and can be adapted to specific needs. This high degree of freedom leads to lots of configuration options for visualizations and customizable user management. The *method-centered* approach brings a predefined information model and some predefined visualizations and reports. The information model is often based on best practices and methods. Most parts of the model are fixed and only few adaptions are possible. Visualization configurations are also less flexible. Some user management roles can be predefined, too. This category provides more guidance to the user. The *process-centered* category goes one step further. The information model as well as visualizations and reports are fixed. Some EAM processes and execution plans are already implemented that can be applied immediately. Tools of this category provide spare flexibility but lots of user guidance.

The second dimensions considers the customization effort and extension possibilities of EAM tools. *EAM off the shelf solutions* include fixed core functionality that cannot be changed or extended. Introduction of these tools is quite straight and few configuration and customization effort is necessary before the tool can be used. On the other hand, tools categorized as *EAM platform* require customization and configuration efforts during the introduction process. Core functionality, models and visualizations are defined loosely and can be customized either by the vendor itself or other experts according to enterprise demands.

The third dimension introduced by Keuntje et al. [Ke10] is about data integration. The first category *automatic integration* describes tools that import data automatically from existing enterprise systems like CMDBs or business process tools. Data changes and updates are performed automatically, no manual imports are necessary. In contrast, tools that belong to the category *manual import* require manual data import and update operations. Data has to be existent in specific file structures like XML or Microsoft Excel spreadsheets. These tools allow less automation but usually provide better modeling and EAM capabilities [Ke10].

Comparing the findings from Lankhorst [La05] and Keuntje et al. [Ke10], it can be noted that Lankhorst's categories are more generic. The author takes into account adjacent tool categories like business process or IT management tools and investigates their EAM potential. On the opposite side, Keuntje et al. focus on categories that can be linked directly to EAM. Their categorization approach is more detailed and provides more variety. Hence, this approach is used in the remaining work.

2.2.3. EAM Tool Evaluations

After discussing the value of EAM tools and potential categorization approaches, existing research is presented where EAM tools were compared with each other and evaluated. Existing literature on this topic is scarce.

In combination with the above described categorization approach, Lankhorst [La05] positions eleven tools into the composed categories. The evaluated tools pursue different purposes and do not solely focus on EAM. Five tools originate from business process modeling and were extended by some EAM functionalities. One tool is a repository tool that can be applied in the context of EAM. According to Lankhorst, the remaining seven tools offer extensive and flexible support for EA. Today some of the evaluated tools are not available anymore because they were acquired by competitors or focus on other disciplines.

With the *Enterprise Architecture Management Tool Survey 2008* Matthes et al. [Ma08] perform an extensive in-depth analysis on EAM tools. Nine EAM tools are evaluated based on predefined scenarios and a given information model. The scenarios include general tool functionality and support for specific EAM tasks. In section 5 the information model and the applied scenarios are explained in detail.

Schekkerman [Sc14] created an EA tool selection guide for the *Institute For Enterprise Architecture Developments* (IFEAD). Requirements for an appropriate tool selection are described and an overview of existing products is given. An EA tool review framework is presented that consists of two dimensions, core functionality of EA development and tool utility to different kinds of professionals. A list of 32 tool vendors and their corresponding solutions is depicted. Moreover an extensive list of questions about tool selection is provided. A real tool evaluation is not performed, the document shall act as guide for the EAM tool selection process.

The *Pragmatic Enterprise Architecture Framework* (PEAF) conductes a survey across EAM tool vendors [Pr14]. 31 tool vendors participated and answered questions about specific functionality in the areas of importing, exporting, relationships, user interface, diagrams, impact analysis, meta model, target and intermediate models and management. The evaluations are supplied by the vendors themselves and are not verified.

Gartner evaluates EAM tools in the annually published *Magic Quadrant on Enterprise Architecture* report [Ga14]. 14 tools are evaluated based on the dimensions *ability to execute* and *completeness of vision*. Each dimension is made up of seven respectively eight criteria. The evaluation is based on vendor surveys, vendor presentations and customer feedback. Gartner places the tools in a matrix based on the two dimensions and classifies them as *leaders*, *challengers*, *visionaries* and *niche players*. Along with functionality the magic quadrant takes into account vendor's market strength, financial strength, global coverage and support structure, and sales strategies. Additionally, a list of strengths and cautions for each vendor is provided. The 2013 version of the magic quadrant is depicted in figure 2.8.

The *Enterprise Architecture Visualization Tool Survey 2014* by Roth et al. [RZM14] focuses on state-of-the-art EA visualizations. 18 tool vendors and 109 EA practitioners provide insights about current visualization capabilities and market demands. Tool vendors provide

Figure 2.8.: Gartner Magic Quadrant for Enterprise Architecture Tools 2013 [Ga14]

feedback and screenshots of visualization options, and practitioners state EAM tools in use, used visualization types and major pain points.

Table 2.3 summarizes existing work evaluations and comparisons of EAM tools.

2.3. Enterprise 2.0

The term *Enterprise 2.0* has been shaped by Andre McAffee. It refers to the use of Web 2.0 platforms and technologies in the context of enterprises.

> **Enterprise 2.0**: Enterprise 2.0 is the use of emergent social software platforms within companies, or between companies and their partners or customers [Mc14].

Social software 'enables people to rendezvous, connect or collaborate through computer-mediated communication and to form online communities' [Mc14]. In other words, social software supports human beings interacting with each other through computer technologies.

Platforms are 'digital environments in which contributions and interactions are globally visible and persistent over time' [Mc14].

Emergent describes software that 'contains mechanisms to let the patterns and structure inherent in people's interactions become visible over time' [Mc14]. Structures and usage patterns shall not be predetermined but are evolved through interactions by users over time. Adding structural information through linking and tagging content is one example

Author(s)	Year	# of tools	Research target	Evaluation approach
Lankhorst [La05]	2005	11	Short overview of available EAM solutions	Unknown; positioning into categories
Matthes et al. [Ma08]	2008	9	In-depth analysis of features and functionality in EAM tools	Scenario-based simulation
Schekkerman [Sc14]	2013	32	Support enterprises in the EAM tool selection process	No evaluation; framework presentation and list of tools
PEAF [Pr14]	2013	31	Vendor-based tool evaluation	Vendor survey
Gartner [Ga14]	2013	14	Vendor placement in the market	Vendor surveys and presentations, customer feedback
Roth et al. [RZM14]	2014	18	Identification of visualization trends and market demands	Vendor and practitioner survey

Table 2.3.: Comparison of existing EAM evaluations

how these structures can evolve. Moreover, software shall be **freeform** which means that it is 'optional, free of up-front workflow, egalitarian, or indifferent to formal organizational identities, and accepting of many types of data' [Mc14]. Using Enterprise 2.0 platforms shall not be an additional burden for users but participation shall be carried out voluntarily.

Enterprise 2.0 software platforms are based on web 2.0 approaches and technologies that describe a second generation of web context and interaction by users. More details about web 2.0 are explained in O'Reilly's work on the definition and characteristics of web 2.0 [O'07].

Enterprise 2.0 tools are characterized by the *SLATES* functionalities [BBL12]:

- **Search**: Users can search for content they are looking for.
- **Links**: Users can link structured information through Uniform Resource Identifiers (URIs)[1].
- **Authoring**: Users are supported in content creation by tools
- **Tags**: Users are able to categorize content. Therefore, they can build structures and patterns over time.
- **Extensions**: Users receive recommendations about additional interesting content based on algorithms.

[1]One specific URI type is the Uniform Resource Locator (URL) that is used to identify web pages.

- **Signals**: Users can subscribe to feeds or alerts and receive notifications about changes, updates and new content.

Enterprise 2.0 platforms offer the opportunity to share knowledge, foster discussions and connect with new peers [BBL12]. Employees interact and share knowledge usually in organizations with their close colleagues they work with on a daily day base. This group can be classified as strong tied peers. To access further or more specific knowledge sources this group contacts peers they know occasionally. The available knowledge is extended through weak relationships. Additionally, there are potential peers that could be contacted to collect further knowledge. Lastly, in big enterprises there is a group of people that is unknown to single individuals and never gets contacted to share knowledge. This situation is described in figure 2.9.

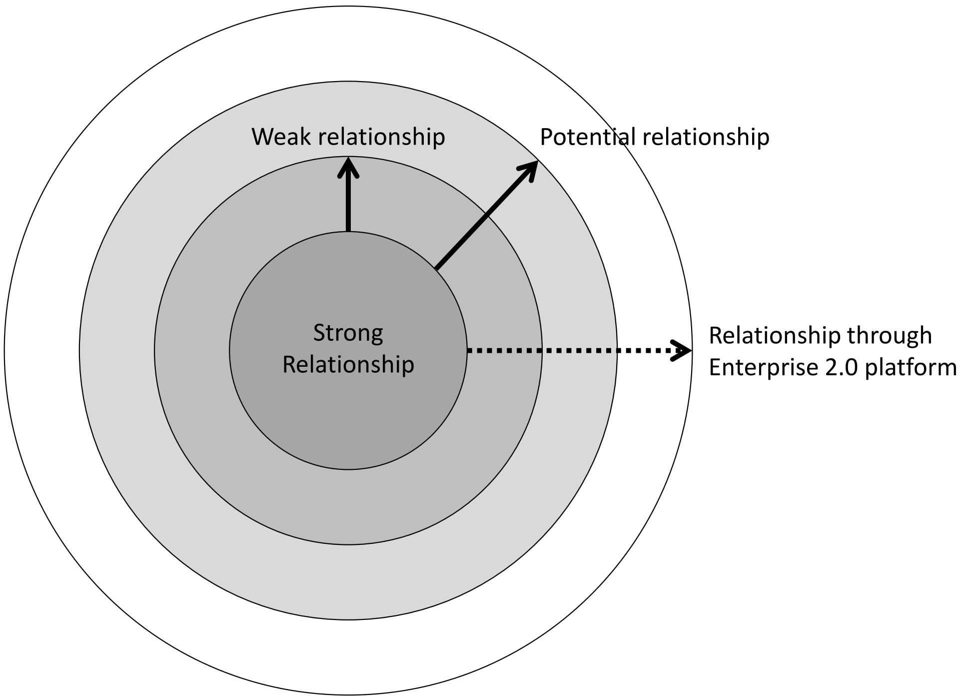

Figure 2.9.: Relationships in an enterprise (adapted from [BBL12])

Through the use of Enterprise 2.0 platforms the available knowledge can be extended. Peers that were inaccessible before can now be integrated to solve problems or share knowledge (cf. dotted arrow in Figure 2.9).

After introducing the concepts of traditional EAM and Enterprise 2.0, these two approaches are combined in the following chapter to derive a new EAM type, collaborative EAM.

Part II.

Conceptual Formulation

3. Collaborative Enterprise Architecture Management

The traditional EAM approach and Enterprise 2.0 have been introduced in the previous chapter to guarantee a common understanding for this thesis. In this chapter traditional EAM is looked at in more detail and potential shortcomings are presented (cf. Section 3.1). Afterwards, the characteristics of enterprises as complex systems are described (cf. Section 3.2). These two findings lead to the idea of collaborative EAM, a new type to manage and organize an EA. Underlying guidelines are introduced in section 3.3. The chapter ends with the application of specific lean and agile principles (cf. Section 3.4) and Enterprise 2.0 concepts (cf. Section 3.5) that distinguish collaborative EAM from the traditional approach. The findings in this chapter are mainly based on the work from from Bente et al. [BBL12].

3.1. Shortcomings of traditional EAM

The aforementioned approach of traditional EAM (cf. Section 2.1) promises to reduce complexity and development efforts, ease integration, decrease IT costs and risks, and improve alignment between business and IT. EAM tools aim at supporting this discipline, thus improving efficiency. Unfortunately, some shortcomings in the application of EAM can limit or even dissolve these benefits. The success of EAM depends on the way EAM is organized and established on an enterprise level. Potential shortcomings in the business organization can be classified into the four dimensions *perspective*, *governance*, *strategy* and *transformation*.

Perspective: Perspective describes the extent to which EAM is performed. The long-term vision and planning can be focused too much and therefore only a rough overview of IT and business landscape is existent. In this case the architectural view is too detached and a wrong picture of real IT is created. The target vision cannot be reached because there are gaps between EA picture and real IT that cannot be filled. On the other side, if enterprise architects only focus on fine-grained details, the long-term vision and the holistic view get lost. Consequently, IT only acts as a supporting unit, business alignment cannot be achieved and new business opportunities cannot be enabled. A balance between these two described extreme situations is desirable.

Governance: In general, EA principles cannot support every potential decision perfectly. If there are significant advantages, it can be reasonable to not comply with particular principles. One example: in order to reduce complexity and development effort, it could be reasonable to use a different programming language than the one stated in the enterprise's principles. If rules are overly strictly enforced by the EA team, flexibility is lost or IT projects get delayed because of unnecessary discussions. Eventually, EA might not be taken seriously by the IT department and the EAM department fails to fulfill its mission. On the other hand, if no principles or guidelines are communicated, every department could use their preferred technology. Lots of separated solutions would be created where integration and communication between each other becomes impossible. Thus, an appropriate level of governance is recommendable.

Strategy: Today there is a fast pace of change in business needs and environments. Traditional EAM brings lots of bureaucracy like many different visualizations of the same model or long review and approval cycles. This bureaucracy is counterproductive to flexible and dynamic businesses. In addition, EAM sometimes concentrates too much on long-term vision instead of focusing on near future. The connection to reality can get lost. Another potential anti-pattern is the so-called *analysis paralysis* where a situation is over-analyzed and the original purpose gets lost. The opposite situation where only a hands-on mentality is existent in an enterprise, and planning and strategy are neglected, is also undesirable.

Transformation: An IT landscape shall evolve in a continuous pace. If IT transformations are performed too fast, new technology could be deployed that is immature or not understood by the staff. Models and documents might be outdated. The risk of IT-related failure increases. Otherwise, if change gets neglected, IT becomes outdated and fails to align with business needs.

In order to avoid all the described anti-patterns a proper balance between communication and documentation needs to be achieved. The key in achieving this balance is to avoid isolated teams but foster collaboration between EA-related entities.

3.2. Enterprises as Complex Systems

As stated in section 2.1 enterprises are complex systems. There are many different environments enterprises have to interact with. Some examples are illustrated in figure 3.1. Enterprises need to align with stakeholders, legal obligations, global effects and need to manage limited resources. These and other factors result in complexity.

The way these complex systems can be managed differ from the way simple systems are dealt with. Simple systems show the same behavior under the same external conditions and similar behavior under similar conditions. In contrast, complex systems can produce varying outputs and show different behavior under similar external conditions. Simple systems can be seen isolated from their environment, while this segregated view cannot be used for complex systems. Bente et al. [BBL12] describe five principles that shall be applied when working with complex systems such as enterprise architectures:

> **Principle 1**: Stay at coarse granularity in analyzing complex systems and predicting their future.

Because of the complexity of enterprise architectures, their fine-grained analysis and future prediction requires extensive efforts. The outcome of such deep analysis is hard to predict and too time-consuming. While managing complex systems, a general understanding of functionality and coarse predictions of multiple future outcomes are preferable.

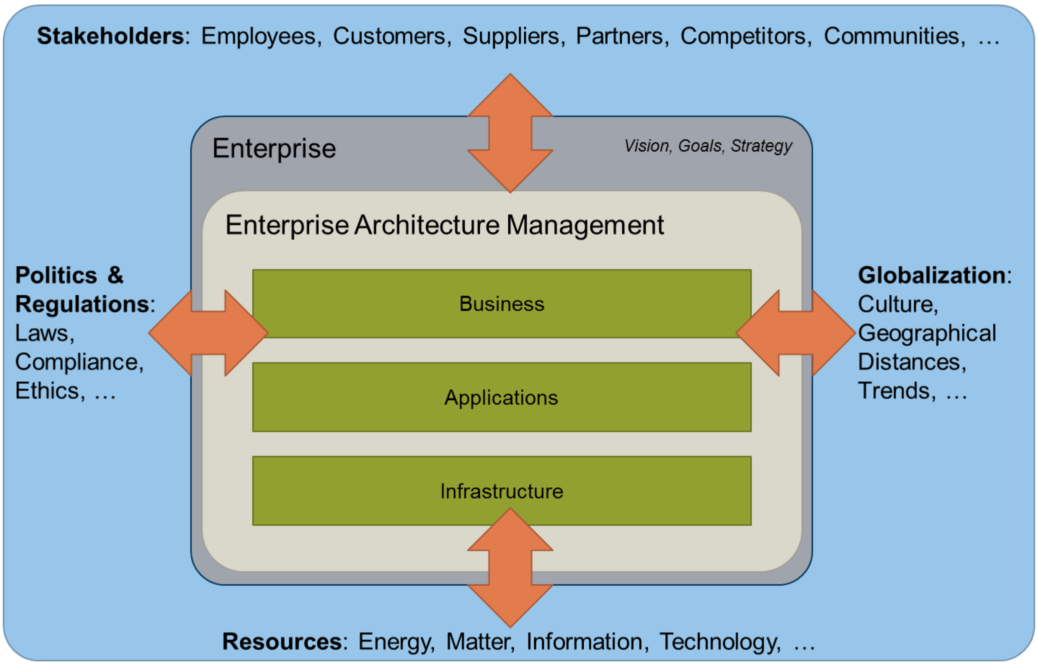

Figure 3.1.: Enterprises as Complex Systems

> **Principle 2**: Strengthening a complex system's ability to change wins over optimizing the status quo.

EAM is a permanent task and environment will eventually change and require adaptions. These changes need to be considered and aligned with the EA. The *perfect* EA is out of reach. Instead of trying to find the perfect state, it is more reasonable to design an EA adaptable to change. Thus, an enterprise can react quicker to demands and stay competitive compared to its contenders.

> **Principle 3**: Management, in particular EA, has about the same complexity as the system it manages.

This principle states that the management of a complex system can never be simple because of the complex object that is being managed. According to Bente et al. [BBL12] this principle shall eliminate expectations that assume that EAM is a simple task.

> **Principle 4**: Complex systems cannot be managed at an object level but only at a meta level (management by rules).

Managing enterprises on an object level means detailed planning and interactions of individual elements. This approach is too complex because the variety of potential actions is too large. Abstractions and inaccuracies are inevitable. Abstract rules and connections to subunits, that are responsible for certain actions on object level, are sufficient. This approach is called *management by rules*. Abstract rules define a framework and self-management capabilities are provided to subunits that are responsible for execution.

> **Principle 5**: Complex systems are best managed at the edge of chaos.

The *edge of chaos* in this principle describes a *'fine line between too little and too much control over a system'* [BBL12]. EA is most efficient if there is just enough control to prevent anarchic actions and deadlock caused by bureaucracy.

Another finding is that network structures are more effective than hierarchies in managing complex systems.

3.3. Guidelines for Collaborative EAM

The aforementioned shortcomings are caused by too complex and inflexible processes, excessive bureaucracy and lack of participation from relevant stakeholders. One approach to eliminate this problems is to integrate lean, agile and participation concepts into EA. These techniques are more suitable for complex systems like EA. In the domain of software development lean and agile principles became popular because they reduce complexity and enable flexibility. A human-centered perspective is created. Applying these principles to EA can achieve the same benefits. Teams that practice EA in isolation cause insufficient solutions because relevant stakeholders are not integrated into the decision process. Instead, human interaction should be emphasized and active participation from EA-related departments should be fostered. The approach can be summarized under the term *collaborative EA* and can be split up into three guidelines which are depicted in figure 3.2.

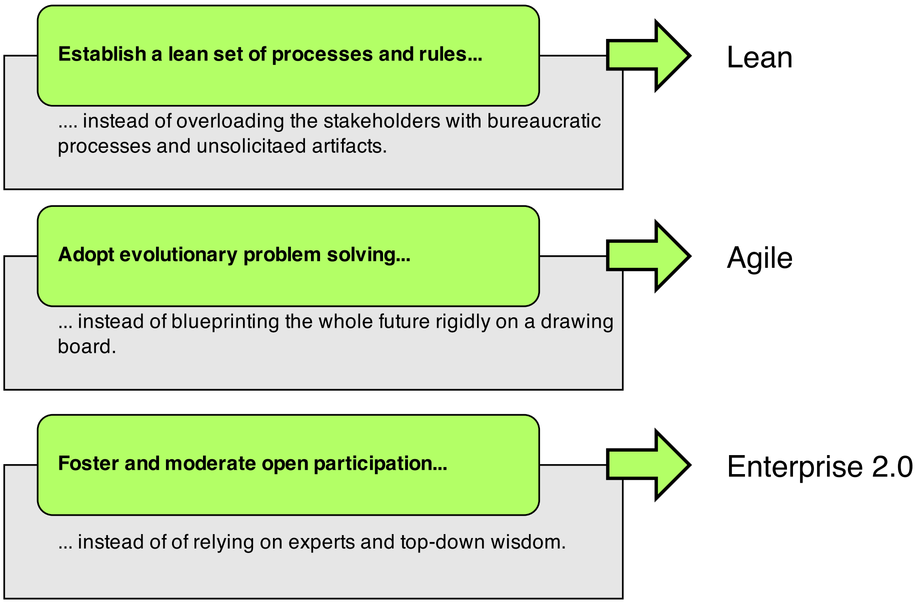

Figure 3.2.: Collaborative EA Guidelines [BBL12]

The first two guidelines refer to lean and agile methods. These methods describe a behavior of welcoming change, working in increments and regular human interaction.

- The first guideline aims at *applying lightweight processes* and avoid unnecessary bureaucracy. Overloaded processes shall be simplified.
- *Incremental change and progress* is fostered in the second guideline. Complex problems are split up and solved in individual cases. Instead of predicting the whole future, continuous deliveries are produced that provide near-term benefits. These deliveries can be presented to stakeholders at an early point in time. Feedback and change requests provided by stakeholders can then be integrated into the continuous EA process.
- *Participation from stakeholders* is promoted in the third guideline. They are asked to provide their knowledge and opinions and thus, are integrated into the decision process. In this way, a wide acceptance can be achieved.

The following chapter explains in detail how these methods can be applied to EA.

3.4. Applying Lean and Agile Principles on EAM

Before applying lean and agile principles on EA, it is first presented where these methods originate from and what their key characteristics are. At the end of this section a study on the application of lean and agile EA principles in practice is summarized (cf. Section 3.4.5).

3.4.1. Foundations of Lean and Agile Software Development

Lean Software Development

Lean software development has its roots in the lean thinking concept that was introduced at Toyota automotive in the late 1940s [PP03]. It can be summarized in seven lean principles for software development [PP06]:

- Eliminate waste,
- Build quality in,
- Create knowledge,
- Defer commitment,
- Deliver fast,
- Respect people, and
- Optimize the whole [PP06].

The fundamental and most important principle is to *eliminate waste* and hence, streamline processes to the very essential steps. The goal is to focus on processes that add value and eliminate processes that are not value-adding, the waste. The principle *build quality in* shall ensure that all developed software artifacts obtain a sufficient level of quality. Building and keeping important knowledge about complex processes is the aim of the principle *create knowledge*. *Defer commitment* describes a principle that important decisions are tried to postponed to stay flexible and to be open to multiple options and to adopt change. Responding quick to customer requirement changes is meant with *deliver fast*. Letting people take decisions that are best qualified for the specific topic is part of the principle *respect people*. *Optimize the whole* refers to the focus on the overall picture in software development. Decisions shall be taken based on the overall solution and not on particular components. A more detailed description about lean software development is provided by Poppendieck & Poppendieck [PP03].

Agile Software Development

The agile software development methodology evolved as a counterpart to the traditional waterfall software developing method. Software projects applying the waterfall method often overrun deadlines, do not meet the budget and do not consider changed requirements. Therefore, an agile approach evolved that emphasizes the introduction of agility into the software development process to adapt to changed demands. The core characteristics are summarized in the *Manifesto for Agile Software Development* [Be14]:

- 'Individuals and interactions over processes and tools,
- Working software over comprehensive documentation,
- Customer collaboration over contract negotiation,
- Responding to change over following a plan.'

Agility describes an iterative, time-boxed approach. Software development is split up into time frames of fixed length where working software artifacts are developed. The customer is very much involved into the development and provides regular feedback that is considered in the following development cycles. Human interaction and adapting change are key characteristics. One of the most famous agile methods is *Scrum*. Cohn [Co10] provides a detailed explanation of the agile software development methodology and Scrum.

The lean software development principles and the agile manifesto do have some similarities. For example, the guidelines *build quality in* and *working software over comprehensive documentation* both emphasize the quality of delivered artifacts. Human interaction and adapting change are also underlined in both methodologies. Nevertheless, it can be noted that lean techniques provide principles while agile methods describe practices. That is why the techniques complement each other and can be applied in combination very well. The following sections show in detail how the principles can be applied on EA.

3.4.2. Creating Lean EA Processes via Streamlining

EA processes are often overloaded or do not provide additional value. It occurs that process descriptions do not reflect the actual performed processes because descriptions are either too complex or outdated. Unnecessary process steps that do not add value can be identified and eliminated. Eliminating this *waste* results in lean processes that focus on pragmatism. The different types of waste that can occur in EA processes can be sorted into seven classes:

- Partially done work,

- Overarchitecting,
- Redundant processes,
- Handoffs,
- Task switchting,
- Delays and
- Defects.

The categories are now introduced in detail. Examples of waste that can occur while performing the EAM tasks introduced in section 2.1.1 are given.

Partially Done Work

EA processes produce artifacts like models, specifications or documentations. Artifacts that are not fully completed and remain in an unfinished state can be classified as *partially done work*. Artifacts that represent information in an insufficient way for different stakeholders can also be included into this category. For example, the non-existence of appropriate model views for specific stakeholders can lead to misunderstandings and misuse of information. These cases can also be categorized as *partially done work*. Reasons for semi-finished artifacts can be lack of time, the emergence of more important tasks, or missing emphasis on completing tasks. Usually, unfinished artifacts are not much used in enterprises. Hence, the creation of these unfinished and unused artifacts requires lots of resources that can be spent more effectively. Table 3.1 illustrates examples of this waste type grouped per EAM task.

Overarchitecting

The waste type overarchitecting describes a behavior where documents are produced that are not needed and not used by anyone. Additionally, documents can be created in such level of detail that they are too detailed and not read. The effort is put into too detailed activities that are not interesting for anyone. This kind of waste leads to documents that cannot be used on the level EAM is executed. Overarchitecting examples are listed in table 3.2.

Redundant Processes

Waste because of redundant processes describes EAM that focuses the correct level of detail but is too work-intensive. EAM is not organized efficient and needs too much resources to achieve acceptable results. Potential reasons are too much bureaucracy or repetitive activities that do not add value. Hierarchical organized enterprises can suffer this waste type when decisions have to be taken by top management that does not have the required expertise. In these cases rationale needs to be communicated across several levels before a decision can be taken. Redundant processes also describe a situation where individual solutions are developed although an easy-to-adopt industry-wide standard would be sufficient. Further examples linked to EAM tasks are presented in table 3.3.

EAM Task	**Waste due to *Partially Done Work***
Defining IT Strategy	No or insufficiently reviewed strategies
Modeling the Architecture	• Unstructured information • Inappropriate representation for target audience • Lack of central document repository and clear storage guidelines • No clear policy about the use of outdated information
Evolving the IT Landscape	• Outdated or incomplete documentation and specification of architectures • Systems not integrated in EAM process • Systems in transition to an EA-compliant status
Assessing and Developing Capabilities	• No established EA organization • Architectural skill gaps identified but no actions proceeded
Developing and Enforcing Standards and Guidelines	• Unfinished EA principles and standards (incompletely elaborated, discussed with stakeholers, and documented) • Insufficient top-level management support causing incomplete implementation of EA standards and guidelines • Standards not considering business requirements • Standards and guidelines exist but are not published inside the enterprise
Monitoring the Project Portfolio	• IT projects are executed but not communicated to EA team • Violations of EA principles are ignored and have no consequences
Leading or Coaching Projects	• Project architects are left alone • Enterprise architects are only requested as technology consultants
Managing Risks Involved in IT	Too IT-centric view on risks, neglecting business consequences

Table 3.1.: EA Waste due to Partially Done Work (adapted from [BBL12])

EAM Task	**Waste due to *Overarchitecting***
Defining IT Strategy	• IT strategy based on technology hype, vendor setup or product rather than business strategy • Overly complex solutions for simple problems
Modeling the Architecture	• Modeling in too much detail and lack of abstractions • Architecture view is too formal for particular stakeholders • Uncontrolled flow of information: forwarding too much information to too many recipients
Evolving the IT Landscape	• Performing IT transformations without alignment of business needs or IT strategy • Investing in technology up-front in anticipation of future needs • Planning too far into the future
Assessing and Developing Capabilities	Bloated EA organization
Developing and Enforcing Standards and Guidelines	• Enforcing EA guidelines and standards that turn out to be irrelevant, abandoned or ignored • Too detailed specification of EA guidelines and standards
Monitoring the Project Portfolio	Too much involvement of enterprise architects in very detailed activities of IT projects
Leading or Coaching Projects	• Enterprise architects only working as project architects • Enforcing EA concepts in IT projects just because of compliance, not because of rationale
Managing Risks Involved in IT	• Listing risks that are unlikely, too far in the future or have already occurred • Applying EA as an overreaction to prevent problems from the past popping up again

Table 3.2.: EA Waste due to Overarchitecting (adapted from [BBL12])

EAM Task	**Waste due to *Redundant Processes***
Defining IT Strategy	Creating a new IT strategy that is not linked to business strategy
Modeling the Architecture	• Involving the wrong audience in reviews • Too many review iterations • Creating models without particular benefit
Evolving the IT Landscape	• Analyzing the current architecture too deeply and forgetting about the original target • Too many meetings with too many participants • Hierarchical chain of command that requires lots of bottom-up communication before decisions can be taken
Assessing and Developing Capabilities	Creating custom frameworks from scratch where industry-specific blueprints and EA methods are sufficient
Developing and Enforcing Standards and Guidelines	• Use of proprietary technology where standard frameworks can be customized to personal needs easily • Too much compliance and bureaucracy
Monitoring the Project Portfolio	Too many meetings with too many participants
Leading or Coaching Projects	Management of project architects with an unnecessary level of detail
Managing Risks Involved in IT	Too much risk management on an overly detailed level

Table 3.3.: EA Waste due to Redundant Processes (adapted from [BBL12])

Handoffs

The waste type handoffs relates to frictions that take place when artifacts are handed over to other parties. If the recipient does not understand the received artifact, communication is needed and delays show up. This waste type especially occurs when people with diverse backgrounds participate in the EA process. If the sender perceives the handover as a strict boundary and does not care about the remaining work, collaboration and proceeding works become very difficult. This so-called *silo mentality* causes lots of needless effort. Handoffs can also occur when a design piece, e.g. a future IT landscape, was created by personal that is no longer in the company and current personal is unable to understand the existing document. Further examples are provided in table 3.4.

Task Switching

Task switching refers to a situation where one architect needs to perform lots of activities in parallel. The person is overloaded with different information and gets distracted from core activities because of minor tasks. This results in reduced effectiveness. Typical reasons are understaffed EA departments or insufficiently described responsibilities. Table 3.5 states additional examples for the task switching waste type.

Delays

Delays occur often in the EA area because many stakeholders have different main priorities than EA. On the one hand, enterprise architects need to wait for input from others, e.g. wait for information, approvals or feedback. On the other hand, the EA team can also cause delays when IT departments would like to use new technologies that are not yet integrated into EA guidelines. Waste due to delays are is listed in table 3.6.

Defects

Defects in EA correspond to all kinds of misguided design. These flaws can have many different sources and are difficult to avoid because of the human-centric perspective. Nevertheless, an aim should be to minimize the amount of defects. Table 3.7 illustrates some typical EA defects from different areas.

Instruments to identify Waste

The described types of waste and unnecessary parts of the EA process need to be identified and consequently removed. Hence, implemented EA processes need to be analyzed with respect to potential waste to be eliminated. A detailed process analysis is necessary that looks at every process step and questions if value is added to the overall goal. In traditional lean techniques this is called *value stream analysis*. There are multiple practical instruments that can be applied in this analysis:

- **EA waste matrix**: An EA waste matrix combines the introduced waste types with EAM tasks inside an enterprise. It illustrates which EAM tasks do have potential for waste removal. The EA waste matrix can be used as a first attempt to identify possible

EAM Task	**Waste due to *Handoffs***
Defining IT Strategy	IT personal strictly executing business orders without proactive thinking to enable new opportunities through IT
Modeling the Architecture	• Information access is restricted because of technical limitations, overly strict security guidelines or legal constraints • Incompatible file formats
Evolving the IT Landscape	• IT personal isresponsible for specific parts but no one takes over end-to-end responsibility or keeps in mind the overall picture • Imprecise system documentations and unclear where to find information • Geographical distance between EA stakeholders makes communication and meeting arrangements difficult • Relying solely on external partners for EA core activities
Assessing and Developing Capabilities	• Only searching outside of the organization EA capabilities but forgetting to foster EA skill creation internally • EA organization has no control over its own processes
Developing and Enforcing Standards and Guidelines	• Not taking into account concerns from IT ground personal when new standards and guidelines are created • Project architects that are unaware of EA guidelines lead to delayed approvals for technical solutions
Monitoring the Project Portfolio	• Outsourcing situations where the original enterprise completely looses interest for the outsourced activity after handover • Lack of proper status reporting
Leading or Coaching Projects	• Lack of communication between enterprise architects and project managers • Lack of effective enterprise-wide knowledge management system
Managing Risks Involved in IT	Lack of communication between EA and business department so that business or IT risks are underestimated

Table 3.4.: EA Waste due to Handoffs (adapted from [BBL12])

EAM Task	**Waste due to *Task Switching***
Defining IT Strategy	Permanently reworking the IT strategy, trying to integrate volatile or contradictory business strategies
Modeling the Architecture	• Working on too many architectures in parallel due to lack of available EA personnel or due to wrong level of detail • Modeling tools with poor usability or lack of tools • Too much detailed information causing an overload
Evolving the IT Landscape	• Unproductive meetings that consume too much time • Responsibility for too many applications
Assessing and Developing Capabilities	EA organization understaffed
Developing and Enforcing Standards and Guidelines	Lack of enterprise-wide knowledge and importance of EA causing unnecessary discussions and explanations of basic concepts
Monitoring the Project Portfolio	Enterprise architect position misused as program manager for IT projects
Leading or Coaching Projects	• Enterprise architect involved too deeply in running IT projects • Enterprise architect's expertise abused to fix particular technical problems • Too much time spent on insufficient communication environment
Managing Risks Involved in IT	Reactive risk management where only occurred problems are approached

Table 3.5.: EA Waste due to Task Switching (adapted from [BBL12])

EAM Task	Waste due to *Delays*
Defining IT Strategy	Waiting for approvals, business strategies or business decisions
Modeling the Architecture	Waiting for information from business or IT
Evolving the IT Landscape	Waiting for information or feedback from business experts or IT departments
Assessing and Developing Capabilities	• organizational delays in the hiring process • Skilled architects cannot be moved to the EA team because of political or budget issues
Developing and Enforcing Standards and Guidelines	• Missing feedback from stakeholders regarding guidelines • Waiting for approval for guidelines and standards
Monitoring the Project Portfolio	Waiting for information on IT projects
Leading or Coaching Projects	EA process is causing project delays because EA point of view is not decided yet
Managing Risks Involved in IT	Business side does not provide feedback on risk assessment

Table 3.6.: EA Waste due to Delays (adapted from [BBL12])

waste. It can be set up at workshops with stakeholders from different backgrounds. Thereby, concerns and knowledge from different departments can be considered. Table 3.8 illustrates an exemplary EA waste matrix.

- **Process activity mapping**: Process activity mapping is useful to analyze process sequences. It illustrates how EA artifacts flow in one process and where improvements are possible. There is now fixed standard notation and additional factors like timelines, KPIs or swimlanes that assign that assign activities to actors can be added.

- **Design structure matrix**: A design structure matrix is a simple matrix that illustrates which process steps provide inputs for other process steps. An example is presented in table 3.9. The intersection of column k and n is filled with a checkmark if step k provides input to step n, e.g. *3 Model business architecture* provides input to *4 Evaluate technical solutions*. Then, the intersection of column 3 and row 4 is filled. All fillings that are below the diagonal line are fine. They represent sequential progress. If fillings are above the diagonal line, the process returns to a previous step. Review cycles and iterations are often reasons for these backward jumps. These activities provide potential for waste elimination and need to be analyzed in further detail.

- **Pipeline response matrix**: This matrix displays process tasks as pipes and shows the relation of time that is spend for performing the process step vs. time that is spend on concurrent work. It helps in identifying delays and unnecessary efforts that are caused by additional tasks.

EAM Task	**Waste due to *Defects***
Defining IT Strategy	• IT transformations not aligned with business strategy • IT strategy based on wrong assumptions
Modeling the Architecture	• Models that neither reflect current nor any future state • Abandoning the use (or choosing unsuitable) modeling languages and notations • Presenting confusing models to nontechnical stakeholders
Evolving the IT Landscape	• Taking IT landscape decisions based on political reasons or personal preferences, not on rationale • Performing changes to the IT landscape that are not accepted by business
Assessing and Developing Capabilities	Creating an EA practice without proper planning
Developing and Enforcing Standards and Guidelines	Wrong platform, technology, or vendor decisions based on political reasons
Monitoring the Project Portfolio	Pursuing projects because of personal preferences
Leading or Coaching Projects	Disconnection of IT projects and EA
Managing Risks Involved in IT	Inappropriate risk assessment

Table 3.7.: EA Waste due to Defects (adapted from [BBL12])

EA Activity	Potential for Waste Removal (● = High, ◑ = Medium, ○ = None)						
	Partially Done Work	**Overarchitecting**	**Redundant Processes**	**Handoffs**	**Task Switching**	**Delays**	**Defects**
Defining the IT Strategy	◑	○	○	○	○	●	○
Modeling the Architectures	●	●	◑	○	○	○	○
Evolving the IT Landscape	◑	○	○	◑	○	○	○
Assessing and Developing Capabilities	○	○	○	◑	◑	○	○
Developing and Enforcing Standards and Guidelines	○	○	◑	○	○	●	○
Monitoring the Project Portfolio	○	○	○	○	○	○	●
Leading or Coaching Projects	○	○	○	○	○	○	○
Managing Risks Involved in IT	○	○	○	○	○	○	○

Table 3.8.: Exemplary EA Waste Matrix (adapted form [BBL12])

	EA Process Step	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	Aggregate business requirements	■						
2	Review by business department	✓	■	✓				
3	Model business architecture		✓	■		✓		
4	Evaluate technical solutions			✓	■			
5	Review by IT team			✓	✓	■		
6	Formalize technical specification			✓	✓	✓	■	✓
7	Handover to development team						✓	■

Table 3.9.: Exemplary Design Structure Matrix (adapted form [BBL12])

The introduced instruments help in identifying EA process steps that can be improved. As a result of the analysis, it is useful to create a list of process improvement proposals that acts as a base for concrete improvement actions. In this way, process overload can be reduced and lean processes can be build.

3.4.3. Introduction of EA Scrums

The Scrum method is useful to integrate stakeholder feedback on a regular basis and foster communication between multiple departments. As EA is a discipline with lots of interactions with business and IT, the method can act as a useful support. Scrum can be applied in different contexts:

- It can be applied in a usual IT project where enterprise architects are involved. If there is a lot of architectural work to do, the architects can initiate a dedicated EA scrum to keep track of their work. They are still part of the project team and work closely together with other team members. The EA scrum is useful to align project work with predefined EA guidelines and create new standards and guidelines for the entire enterprise based on project work. Figure 3.3 presents such a setup.

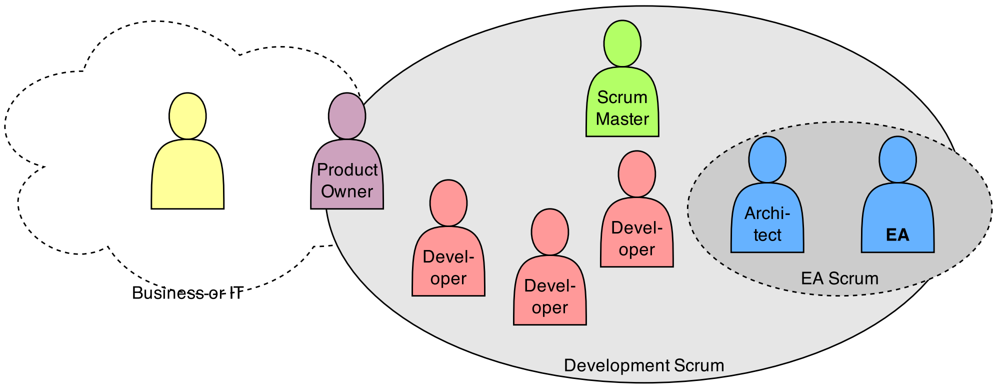

Figure 3.3.: Agile IT project with integrated EA Scrum team (adapted from [BBL12])

- Another useful situation to apply the Scrum methodology is in dedicated EA projects, e.g. with the aim of modeling architectural concepts. The Scrum team members can consist of EA and IT team members and one enterprise architect acts as product owner who keeps in contact with business. The Scrum principles provide a fixed schedule with clearly defined design phases and review dates. This minimizes delays because deliverables need to be presentable at regularly scheduled reviews. An EA project with a Scrum organization applied is illustrated in figure 3.4.

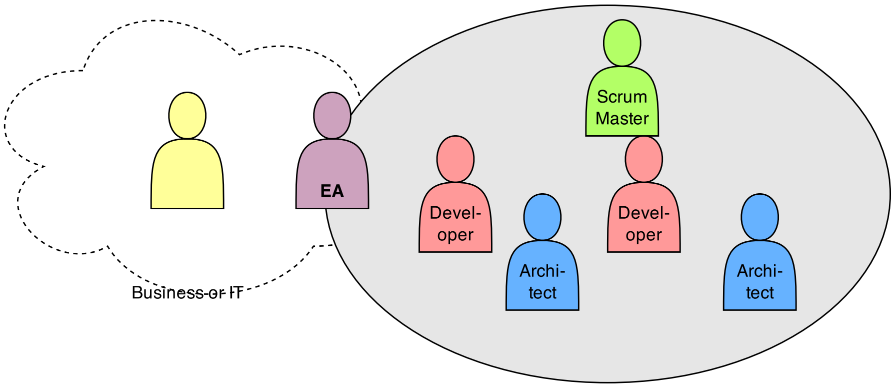

Figure 3.4.: Dedicated agile EA project (adapted from [BBL12])

- Scrum can also be applied to steer or monitor multiple projects. Lead architects of each project meet regularly for overall Scrum meetings. In these meetings they can for example present their current project state, generalize guidelines or coordinate project steps. Again, because of the Scrum characteristics delays are not tolerated and communication and feedback is improved. The underlying projects do not need to follow an agile approach, they can also apply the waterfall model. An enterprise architect acts as scrum master or product owner, or a combination of both. Figure 3.5 illustrates the described structure.

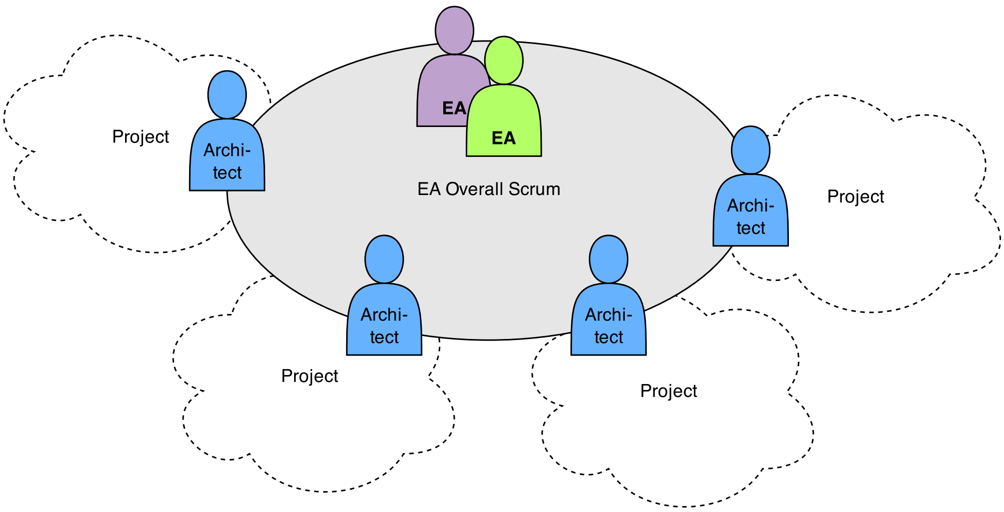

Figure 3.5.: EA overall Scrum (adapted from [BBL12])

- The EA team can also be organized in a Scrum structure. The leader of the department acts as scrum master. There are multiple product owners because EA usually is concerned with multiple artifacts at the same time. These product owners are assigned to stakeholders and have a deep knowledge of their corresponding business or IT area. A rhythm of sprints followed by fixed scheduled reviews should be established. Thus, stakeholders get integrated into EA work regularly and EA becomes a key player inside the organization. Delays are prevented and quality improves because results need to be demonstrated at these reviews. An exemplary setup is portrayed in figure 3.6.

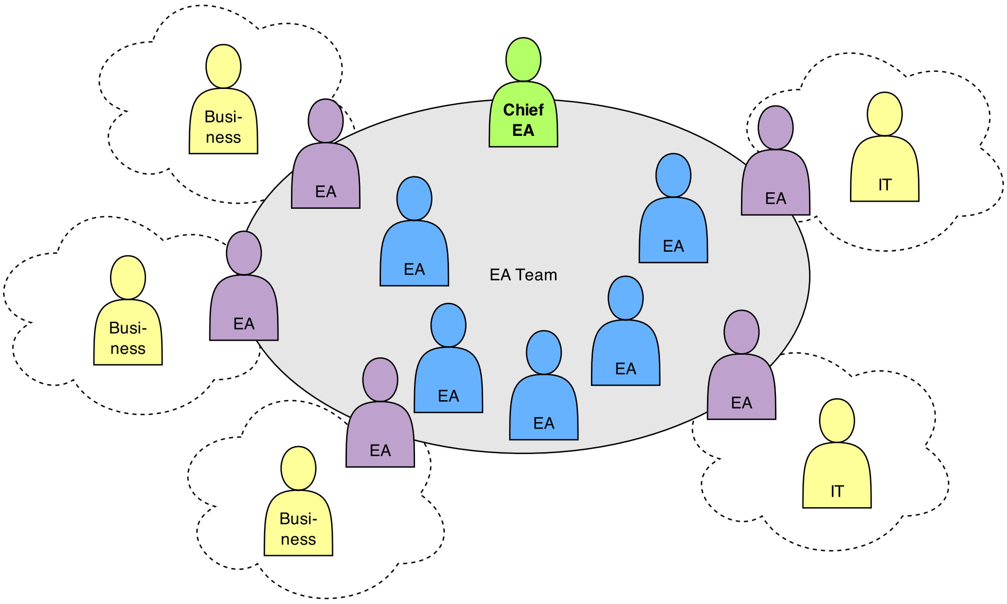

Figure 3.6.: EA team organized in Scrum structure (adapted from [BBL12])

The agile structures stated above do not require big organizational changes and can be tested in pilot projects before they get established in the entire enterprise. Establishing these structures should lead to an improved information flow between EA, business and IT. The EA team can adapt to a right perspective, a balance between future vision and level of detail. EA guidelines can be better communicated and adapted to stakeholder concerns to become more realistically. Design and development pace should increase and thanks to quickly provided feedback, defects should be reduced.

3.4.4. Iterative EA practices with TOGAF

As mentioned in section 2.1.2 TOGAF supports iterations in the ADM process. Therefore, agile principles can be applied in combination with the TOGAF framework. A way in which iterative work can be performed in the ADM process is displayed in figure 3.7.

Especially the phases A-F (cf. Figure 3.7) where architectural concepts are created can be executed in iterations. The iterations can be performed on different levels of detail. In early stages of EA projects when work is done on an abstract level and detail work is postponed, it might be useful to apply short loops. Thus, feedback can be gathered quickly and integrated into further design. TOGAF proposes three levels of granularity: the *enterprise strategic architecture* level is comprised of *segment architectures* that again can be partitioned into *capability architectures*. When the EA project is started on the enterprise strategic level the described short cycles prevent overarchitecting and ensure flexibility as well as business and IT alignment. Other ADM phases can also be performed through iterations such as transition planning, architecture governance or architecture capabilities (cf. Figure 3.7). Enterprises can also create their own iteration cycles based on individual needs.

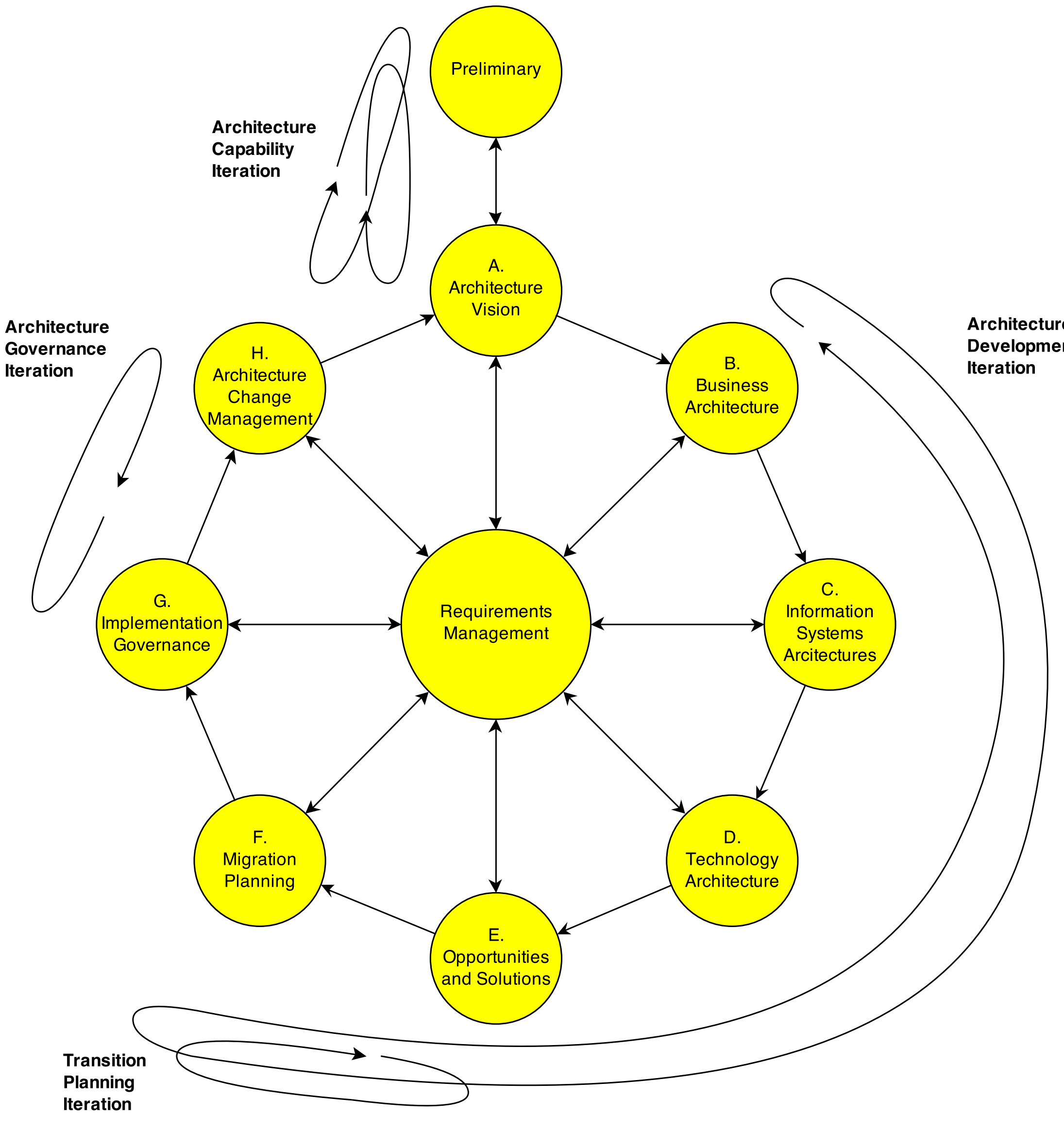

Figure 3.7.: TOGAF ADM process with iterations (adapted from [BBL12])

Another important artifact of agile methodology is the backlog that lists prioritized requirements. It can also be applied in EA and used in combination with TOGAF. Table 3.10 represents an excerpt of an exemplary *EA backlog*. Requirements can be ordered per TOGAF architectural level and split up into further EA tasks that are linked to the ADM process phases. Iterations are displayed through the repetition of phases. The agile EA backlog also considers the progress from enterprise strategic architecture over segment architecture to capability architecture. The backlog order is not fixed and can be adapted to consider changes.

	TOGAF Phase(s)			
Requirement	**ESA**	**SA**	**CA**	**EA task**
Consolidate IT landscape	A			Define KPIs and methodology
	B			Elicit business strategy
	C			Gather existing standards and guidelines
	D			Evaluate used technologies
	E			Build work packages (for SA)
	F			Estimate effort and time
Determine current IT landscape		B		Model business processes
		C		Determine and model application architecture
		D		Model infrastructure architecture
		E		Combine architectures
Review created artifacts		B		Review business model
		C		Review application architecture
		D		Review infrastructure architecture
		E		Integrate review feedback
		F		Split up into more detailed activities (for CA)
Analyze application relations			C	Identify interfaces
...			...	...
ESA = enterprise strategic architecture, SA = segment architecture, CA = capability architecture				

Table 3.10.: Exemplary EA Backlog (adapted form [BBL12])

The *Kanban* board is another agile tool that can be applied in agile practices. The idea is to use task cards that flow through a sequence of process steps. The task cards are put on a board that visualizes the progress. It can also be used in combination with the TOGAF ADM process to create an *EA Kanban board*. An example is depicted in figure 3.8.

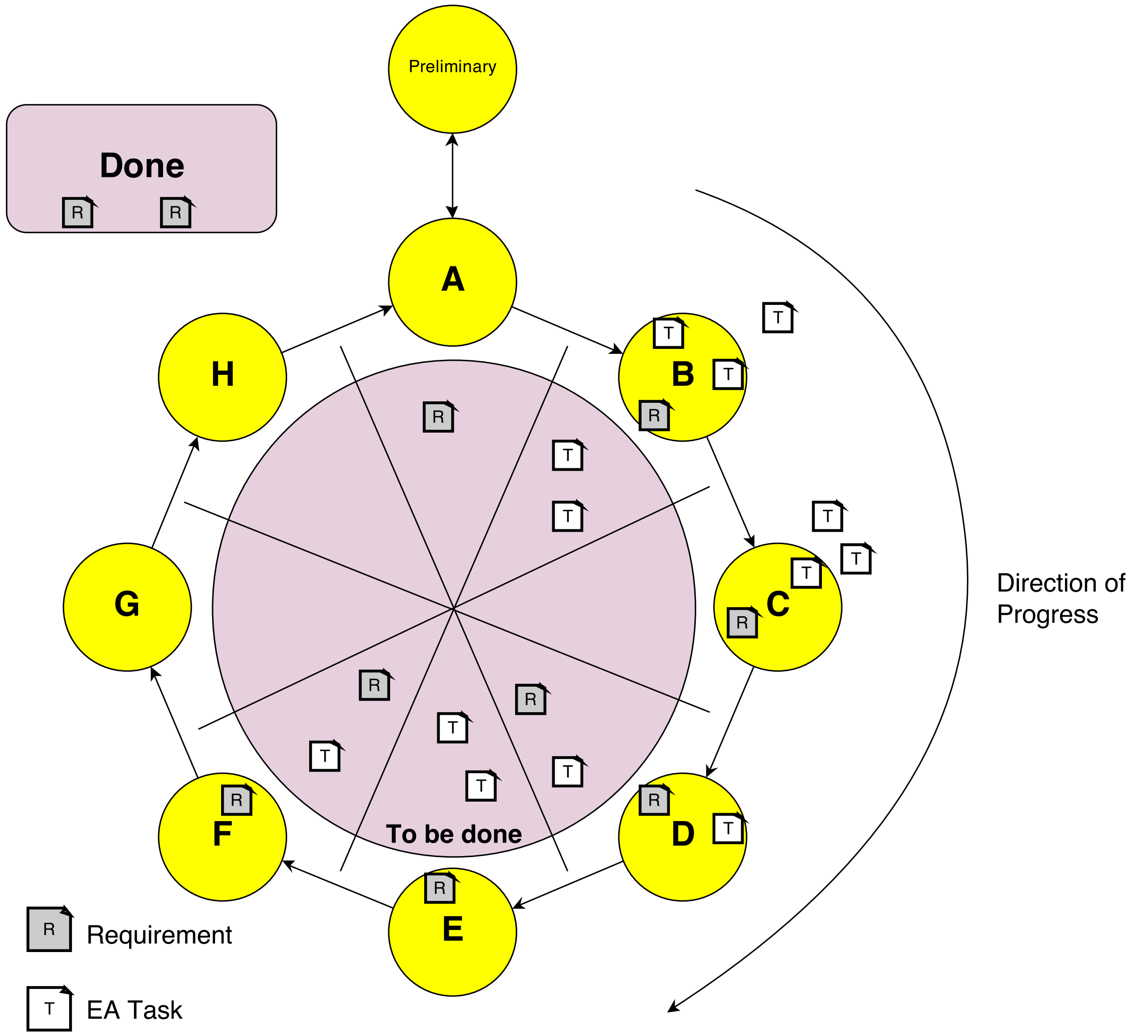

Figure 3.8.: Exemplary EA Kanban board (adapted from [BBL12])

- The task cards marked with *R* represent requirements from the agile EA backlog. These *Kanban cards* arrive at phase A, architectural vision, and flow through the ADM phases. The amount of concurrent requirements that are processed in one phase should be limited to avoid overhead and delays.
- The inner circle of the EA Kanban board represents the EA backlog and all tasks that need to be done. The EA backlog is divided per ADM process phase. If requirements are not processed yet, they are put on hold there.
- When a requirement finishes one phase it goes to the backlog of the next phase. Before the requirement is processed in this phase, further EA tasks are derived that need to be completed before the requirement can move on to the subsequent phase.

- These EA tasks are put into the phase's *to be done* area and are only moved into the phase when someone works on them. They are marked with *T* in the visualization. The creation of new EA tasks is performed by enterprise architects. It gives them flexibility on which level of detail they would like to perform the corresponding phase.
- If one task card is finished, it is put outside the phase and work on remaining task cards of the phase can start.
- When all tasks of the corresponding requirement are done, the requirement has finished the phase and can be put into the backlog of the next phase. This approach is repeated for all requirements and all applicable phases.

The EA Kanban board creates a constant flow in the ADM phases A-H that results in reduced cycle times. It represents current progress and increases visibility. Tasks can be planned flexible depending on various factors. Tasks can also be dropped to speed up the completion of one cycle. Then the whole TOGAF cycle needs to be iterated again, though. However, it can be helpful to avoid overloads or delays.

In this section three agile instruments that can be applied in combination with TOGAF to create agile EA processes were presented. Flexibility can be enhanced, requirements can be structured and adapted easily, and iteration cycles adjusted to be aligned with specific needs.

3.4.5. Lean and Agile EA Principles in Practice

An empirical analysis on the application of lean and agile principles in EAM is conducted by Hauder et al. [Ha14]. 102 participants participated in an online survey. The results of this study indicate that agile principles are applied especially in the field of EAM introductions and operations. Principles concerning time or quality are less frequently applied.

Incremental and iterative approaches, which can be especially well performed in Scrum teams, are pursued by 87% respectively 79% of all participants. 93% of the respondents operate cross-functional which can be supported e.g. through EA overall Scrums (cf. Figure 3.5). 59% accomplish EAM in small subteams. Integrated EA Scrums in agile IT projects such as depicted in Figure 3.3 could be a suited approach to organize these small subteams. Additionally, 59% perform EAM with defined roles & responsibilities which is important in the previously introduced Scrum approaches.

On the other hand, only half of the participants focus on requirements and a minority of respondents (28%) exactly respond to the stakeholder's demands. Previously introduced instruments like the EA backlog (cf. Table 3.10) or the organization of the EA team in Scrum structures (cf. Figure 3.6) could increase requirements and stakeholder awareness. 37% of participants advance EAM with a constant speed which could be fostered by applying the EA Kanban board (cf. Figure 3.8). Although 72% of participants make use of reflections and retrospectives, and 71% incorporate feedback, only 51% answered that stakeholders provide feedback to the EAM team. Moreover, only 38% of the survey participants state that their artifacts are actually used by stakeholders. Consequently, stakeholders in practice are not integrated well into the EA process. EA Scrums and iterative approaches like the EA Kanban board could improve this situation.

Selected findings of the study are shown in Figure 3.9. Lean and agile EA instruments that could support these principles are added in an extra column.

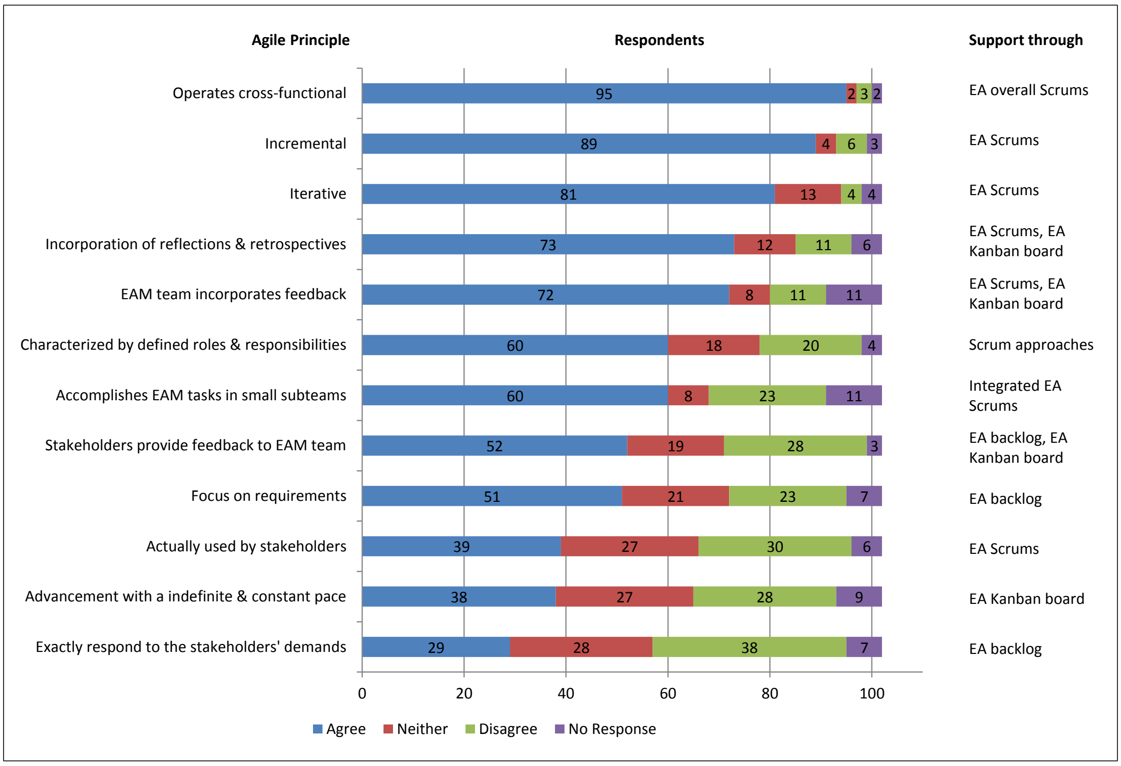

Figure 3.9.: Agile EA principles in practice ($n = 102$) (adapted from [Ha14])

In conclusion, the results of the study show that agile principles are applied today in enterprises partly. The use and application of lean, agile and iterative approaches like the ones previously described could improve EAM and its acceptance.

3.5. Fostering Participation through Enterprise 2.0

Enterprise 2.0 allows different ways of participation. People can participate to share knowledge, improve decision making, or in the area of EA participate in transformations.

Knowledge can be shared and accessed through Enterprise 2.0 platforms. EA insights, guidelines, and arguments for and against decisions can be communicated to a broad audience. One way to provide such knowledge is the installation of special EA blogs. Loosely connected peers get the chance to get in touch with EA. It has to be noted that the communication stream can become unidirectional when blogs are only read but not interacted with. Therefore, participation and interaction should be encouraged. Integrating features to comment on blog posts or rate the content help in establishing an active, viable and accepted platform.

Wikis are another powerful platform to share knowledge and foster participation. In the context of EA the wiki approach can be used for activities that are usually performed in isolation by enterprise architects. Models can be published in wikis and interested stakeholders are invited to confirm, change or discuss about published models. In this way, the modeling activity is opened to a broader audience with diverse knowledge. Other artifacts like standards, application catalogs or entire IT landscapes can also be published in wikis to integrate other peer's opinions. The use of EA wikis can go that far that artifacts are not organized and managed solely by enterprise architects anymore but by a crowd of participants. Business process designers, IT developers or project managers can provide their insights into the EA activity to combine different backgrounds and achieve better results. Newer research approaches like hybrid wikis allow to add structured content by users. This brings new opportunities for participation. Peers can add tags or attributes to wiki pages to link them with other content and structure information. A new way of collaborative EAM is achieved.

Interactions like comments or content ratings are also a great source for feedback. Everyone inside the company gets the chance to convey his or her opinion on particular EA topics. The EA department can collect feedback easily and gets further input besides of regular feedback meetings. Opinions and concerns of people that are not directly connected with the EA team can be gathered.

Participation fostered through Enterprise 2.0 methods can also help in decision making. In complex problems such as the detection of future IT opportunities wisdom of crowds can be seen as valuable input. As noted in section 3.2, network structures can lead to better decision making than hierarchical structures. Potential IT opportunities can be published on social platforms where they can be accessed by a large audience. A crowd can help in deciding on what opportunities shall be pursued. This leads to high acceptance of newly pursued projects inside the enterprise. Peers can also offer their participation in projects through Enterprise 2.0 platforms. Moreover, individuals can participate that were originally not incorporated.

Enterprise 2.0 platforms are also useful to link artifacts of different architectural layers. Through direct links, drill-downs and other elements navigation on the same layer and across layers is eased. Artifacts are no longer kept in domain specific silos but are connected with each other. Possibilities to look into different domains that were not accessible before are created and collaboration is improved.

Microblogs are a useful instrument to keep track of all changes inside an EA. Event messages about changes in the IT landscape, events on operational level, or any other event can be generated automatically and posted under specific categories. Extensive search and filter capabilities allow to access information for particular needs. The Enterprise 2.0 platform can act as a central repository for changes and all EA-related events. Therefore, it helps to structure communication flows and provides collaboration features. These central repository can be especially useful for transformations where non predicted changes can occur often and should be communicated to all relevant stakeholders quickly. Thus, Enterprise 2.0 platforms provide information transparency and reduce communication gaps.

The described Enterprise 2.0 instruments can help in improving participation in EA and integrate various stakeholders. Only a subset of potential examples is mentioned in this section and there are additional Enterprise 2.0 instruments that can support EAM. In conclusion, Enterprise 2.0 can support EAM very well in providing collaboration opportunities. EAM becomes more interactive and in combination with lean and agile principles reduces complexity, allows flexibility, and gathers knowledge from additional stakeholders.

4. Collaborative EAM Tools

The previously explained approach of collaborative EAM (cf. Chapter 3) is based on organizational and structural methods. In this chapter potential tool support of collaborative EAM is elaborated. Therefore, potential tool characteristics for supporting collaborative EAM are derived in section 4.1. The new tool category collaborative EAM tools is introduced. A selection of this kind of tools is provided in section 4.2.

4.1. Characteristics

In this section it is analyzed how collaborative EAM can be supported through EAM software tools. The technology-to-performance chain model by Goodhue & Thompson [GT95] is used to derive tool characteristics and functionalities that are particular useful. This model expresses that task characteristics and technology characteristics can be used to derive a task-technology fit. The task-technology fit describes the degree to which a technology assists in performing tasks. While the original model focuses on task of individuals, it can also be applied for activities that are performed by groups such as collaborative EAM tasks. Thus, the task-technology fit expresses how well technology supports tasks that are performed by groups or individuals. A task-technology fit leads to performance increases and higher utilization which itself again increases performance. The model is shown in figure 4.1.

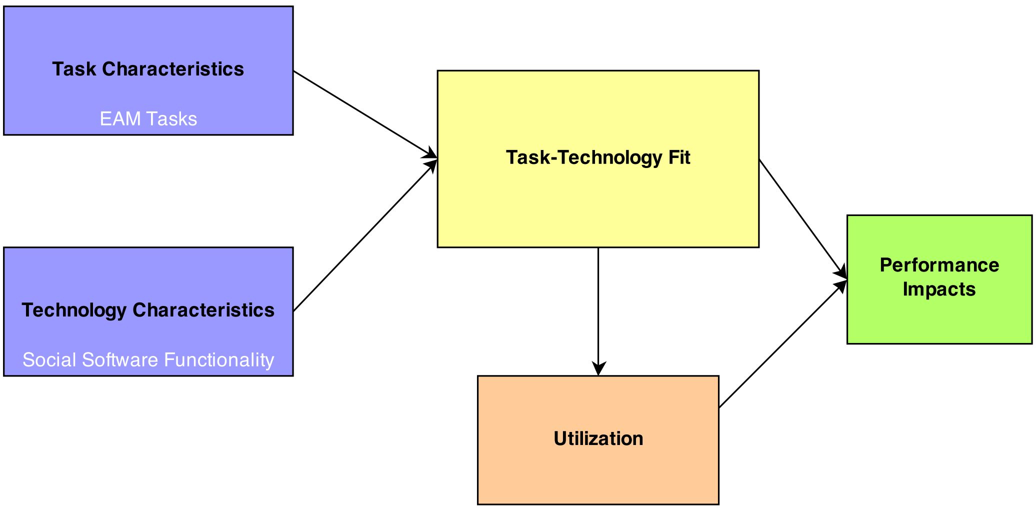

Figure 4.1.: Technology-to-Performance Chain Model (adapted from [GT95])

In the EAM environment the traditional EAM tasks mentioned in section 2.1.1 can be treated as task characteristics. Enterprise 2.0 platforms or social software shall be considered in the dimension technology characteristics. If there is a task-technology-fit between EAM tasks and these social software functionalities, EAM can be supported through these collaborative technologies and overall performance increases. In table 4.1 social software functionality is mapped with EAM tasks. The SLATES functionalities introduced in section 2.3 are used as technology characteristics. A cell filled with a checkmark means that the specific SLATES functionality can support the corresponding EAM task.

SLATES Functionality/ EAM Task	**Search**	**Links**	**Authoring**	**Tags**	**Extensions**	**Signals**
Defining the IT Strategy	✓	✓				✓
Modeling the Architecture	✓	✓	✓	✓	✓	✓
Evolving the IT Landscape	✓	✓	✓	✓	✓	✓
Assessing and Developing Capabilities	✓		✓	✓		✓
Developing and Enforcing Standards and Guidelines	✓	✓	✓	✓	✓	✓
Monitoring the Project Portfolio	✓	✓	✓	✓		✓
Leading or Coaching Projects	✓		✓			✓
Managing Risks Involved in IT	✓	✓		✓		✓

Table 4.1.: Mapping of SLATES functionalities with EAM Tasks

Table 4.1 indicates that EAM tasks can be supported by SlATES functionality in different ways:

- *Defining the IT strategy* can be assisted by search, link and signal functionalities. Searching for and linking information are typical activities that are performed during this EAM task. Furthermore, new or updated IT strategies could be spread using the signal functionality. As the IT strategy is usually defined by a small group of (senior) managers and not by a crowd of individuals authoring does not really assist. Same goes for tag functionalities to structure content, and user-related recommendations that are provided by extension functionalities.
- All SLATES functionalities can provide support for the EAM tasks *modeling the architecture*, *evolving the IT landscape* and *developing and enforcing standards and guidelines*. In these tasks objects or architecture searches are performed. Architectures of different

layers can be linked with each other. Authoring capabilities can be used to model or evolve landscapes, and to create standards. Tags can be used to create user-defined structures of architectures and guidelines. The extension functionality can suggest information about related objects. Moreover, users can be notified about updates through signaling functionality. Collaborative features are able to assist these tasks and thereby, improve their efficiency.

- The task *assessing and developing capabilities* which is about gathering, storing and keeping EA knowledge in the enterprise can be supported by search, authoring, tag and signal functionalities. EA knowledge can be created by multiple users. Search, tag and signal services help in accessing knowledge easily.

- Apart from extensions, all SLATES functionalities can be used for *monitoring the project portfolio*. Users should be able to search for projects, link projects with affected applications or other projects, create content about project status or particular problems, tag projects that belong into the same category, and receive updates on project changes. The extension feature plays only a minor role in project portfolio management.

- The activity *leading or coaching projects* is mainly about communication and working with people. Architects can use EAM tools to search or create new content but this activity is performed mostly outside of an EAM tool.

- *Managing risks involved in IT* relies on existing content. Therefore, search, link and tag functionality are useful to find and access content. User notifications for important events also support this task.

In summary, SLATES functionalities can provide lots of assistance for performing EA tasks. A task-technology fit between Enterprise 2.0 platform characteristics and EAM task exists. Especially search, link, authoring, tag and signal features seem to be able to provide valuable support. Changed EA requirements can be considered quickly through these collaborative and simplifying functionalities. The extension functionality which provides users automatically with additional content based on algorithms seems to be less beneficial in EAM. A possible explanation for this might be that EAM is to a less extent about exploring further possibilities. Instead, users usually know what they are looking for and which task they want to perform.

As described in Goodhue's & Thompson's model (cf. Figure 4.1) a task-technology fit implicates performance impacts. EAM tools that integrate above stated functionality can provide positive effects because they foster participation which results in higher utilization which in turn results in performance increases. These increases can be achieved by providing the ability to

- search for content in an easy way,
- link content,
- provide authoring capabilities to create new content,
- tag capabilities to categorize content according to user's demands, and
- signal functionality to notify users about updates.

These abilities reflect the collaborative EAM guideline of fostering and moderating open participation (cf. section 3.3).

As explained in section 3.3 another collaborative EAM guideline aims at lean EA processes. Using bloated EAM tools that are difficult to use and require lots of effort to create artifacts, hence appear inappropriate. Instead, a simple, not overloaded but lean user interface is desirable. Users shall be able to create sufficient artifacts in a simple and easy way. Adaptions and changes shall be integrated quickly into the EAM tool.

A third guideline of collaborative EAM promotes agility and iteration. EA Scrums and other organizational methods can support these goals. This guideline refers to organizational behavior and is difficult to integrate into EAM tools. A little attempt could be to integrate version control to allow iterative work. Version control itself does not guarantee agility and iteration though, organizational practice is still necessary. Therefore, it only has minor effects on this collaborative guideline.

EAM tools that implement social software functionality to foster collaboration, and provide a lean user interface that stimulates lean EAM processes can be called *collaborative EAM tools*. Characteristics of collaborative EAM tools and examples how they can be integrated into EAM tools are presented in table 4.2. The stated examples are just a subset of potential implementation options.

Characteristic	**Purpose**	**Implementation options**
Search	Quick and easy content access	Well-positioned search bar
Links	Interconnection of architectures and objects	Linked objects provided through drop-down lists, social bookmarks
Authoring possibility	User-based knowledge gathering	Comment and feedback functionality, blogs
Tag feature	Structured content	Tagging service
Signal feature	Notifications on EAM updates	Feeds, subscriptions, user notifications
Lean User Interface	Effective and easy to use EAM tool	Simple user interface with good usability and reduced complexity

Table 4.2.: Characteristics of collaborative EAM tools

It is important to point out that conventional Enterprise 2.0 platforms that provide SLATES functionalities cannot be categorized as collaborative EAM tools in general. Collaborative EAM tools still focus on EA support and hence, should integrate the traditional EA tool features from table 2.1. Therefore, EAM tools with collaborative and lean characteristics (cf. Table 4.2) can be treated as collaborative EAM tools.

4.2. Available Products

In this section available market solutions of collaborative EAM tools shall be stated. An internet search has been conducted where the following keywords were queried in multiple combinations: 'collaborative', 'enterprise architecture tools', 'lean' and 'agile'. Furthermore, lists of EAM tools were scanned. The websites of potential candidates were analyzed. Especially, core functionality and marketing messages were looked at. Table 4.3 is a result of this review stating vendor, tool name and corresponding URL. Tools were integrated into the list if the vendor emphasizes collaborative, lean and agile features. It shall be emphasized that this list is based on vendor information. The mentioned tools were not subject of a detailed analysis and additional tools with similar functionalities might exist.

Vendor	**Tool**	**URL**[1]
LeanIX GmbH	leanIX	`https://leanix.net/en/product/knowledge`
Obeo	Obeo Smart EA	`http://www.obeosmartea.com/product/features`
Orbus Software	iServer Enterprise Architecture Solution	`http://www.orbussoftware.com/enterprise-architecture/capabilities/`
Softplant	Living EAM Platform	`http://www.living-enterprise-architecture.de/#LEA-Produkte`
Visual Paradigm International	Visual Paradigm Enterprise Edition	`https://www.visual-paradigm.com/features/enterprise-architecture/`

Table 4.3.: List of collaborative EAM tools (alphabetical order)

[1] All URLs accessed on June 7th, 2014.

5. EAM Tool Scenario Evaluation

As mentioned in section 2.2.3 different attempts to evaluate EAM tools exist. In this thesis major differences between traditional EAM tools and collaborative ones are elaborated (cf. Section 4.1). The scenario simulation applied by Matthes et al. [Ma08] allows to look into tools in detail, test specific functionality, and compare tools with each other. Thanks to the exemplary information model and corresponding example data, the scenarios can be simulated on the same data set. Consequently all evaluated tools face equal requirements and challenges. In this chapter a short introduction on the used information model and the applied scenarios is provided. As these artifacts are based on the Enterprise Architecture Management Tool Survey 2008, a detailed explanation can be found in Matthes et al. [Ma08].

5.1. Information model

The tool evaluation shall be performed on a detailed level and include cross-functional tasks. Therefore, the information model needs to integrate the described architectural layers and cross-functions of figure 2.2 to reflect typical EAM tasks. Object-oriented modeling is applied for the information model design to make use of the benefits accompanied with this approach and to ensure a common understanding. A graphical representation of the applied information model is depicted in figure 5.1. The consisting object types are introduced briefly below in an alphabetical order.

- **ApplicationSystem**: An application system is a software system that supports at least one business process. In this information model infrastructure components are excluded and are managed as separate object types. This object type is the abstract description of a deployed instance (cf. *DeployedApplicationSystem*). Application systems follow architectural blueprints.
- **ArchitecturalBlueprint**: An architectural blueprint consists of several blueprint elements. It describes the architecture on an abstract level.
- **ArchitecturalSolution**: Architectural Solutions employ actual solution elements. They can be seen as real instances of the abstractly defined architectural blueprints. An attribute can be assigned multiple values that state specific properties of the architectural solution, e.g. high availability.
- **BluePrintElement**: A blueprint element is an abstract building block of an architecture (cf. *ArchitecturalBlueprint*). Implementation-specific details are disregarded. Examples are database or web server. Actual implementations of blueprint elements are realized through solution elements.
- **BusinessObject**: Business objects are business entities (e.g. an invoice) that are used in business processes. They can be manipulated by multiple *functions*.

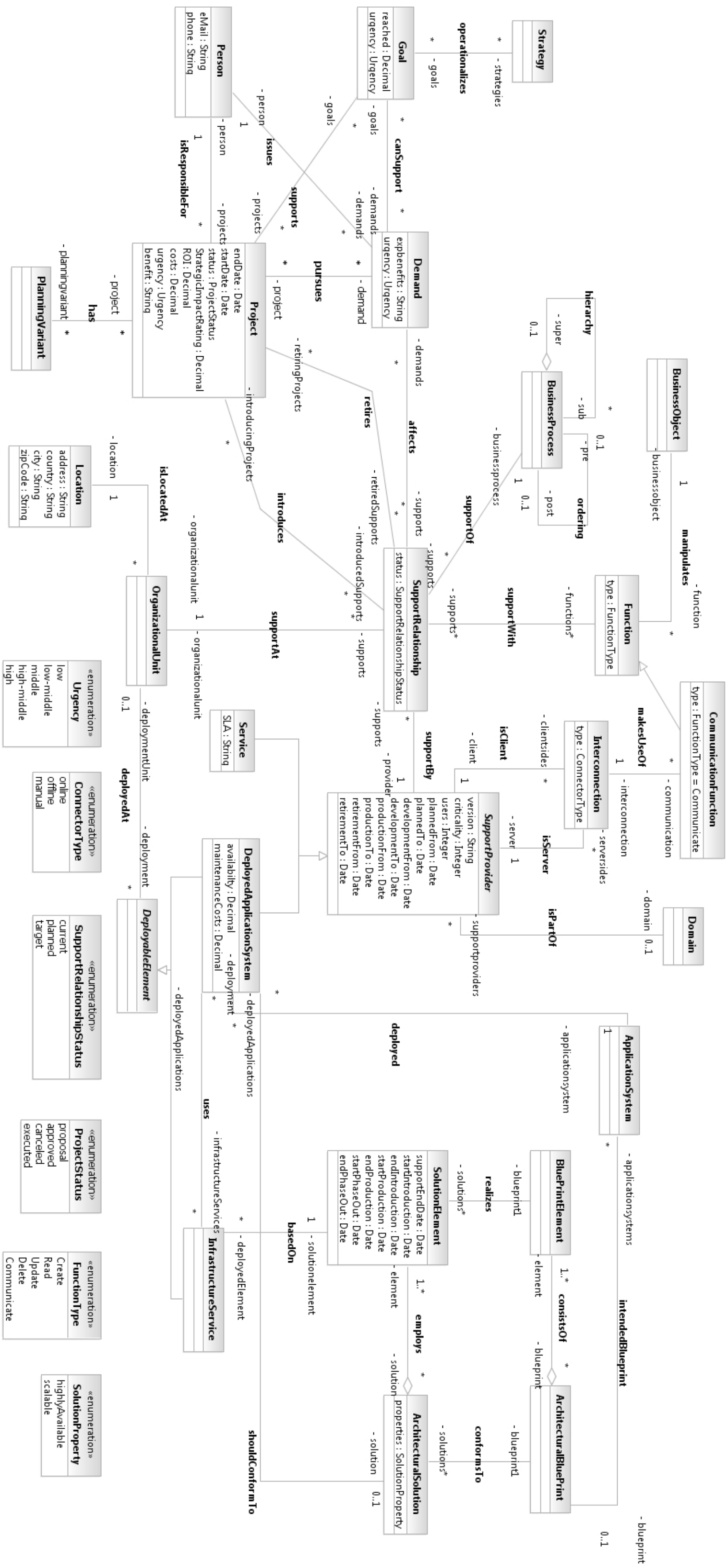

Figure 5.1.: Information Model

- **BusinessProcess**: Business processes describe a connected sequence of logical functions. In the context of this thesis they describe high-level sequences and not individual process steps. One business process can belong to larger *super-process* or can have smaller *sub-processes*. They can be part of an ordered sequence of processes with corresponding predecessor and successors processes.
- **CommunicationFunction**: The communication function is a specialized function (cf. *Function*). It uses interconnections to exchange business objects during business operation.
- **Demand**: Demands or requirements can be issued by persons to change the application landscape. They affect the way business processes are supported (cf. *SupportRelationship*). Demands can support organizational goals and be pursued in projects. One complex demand can also be split up into multiple projects.
- **DeployableElement**: A deployable element can be deployed at an organizational unit. This organizational unit is responsible for the deployed instance.
- **DeployedApplicationSystem**: A deployed application system is an instance of the abstract object type *ApplicationSystem*. As a *DeployableElement*, it can be located at an organizational unit. It is a subclass of *SupportProvider* and hence inherits these attributes. Moreover deployed application systems use infrastructure services to support business. They should conform to *ArchitecturalSolutions*.
- **Domain**: Domains are logical enterprise units that group supportive concepts together for one business division.
- **Function**: Functions are operations that manipulate business objects to support business. Possible operations are *create*, *read*, *update*, *delete* (CRUD) or the special *communicate* operation.
- **Goal**: Objects or targets that should be achieved can be stated as goals. Goals operationalize strategies. The degree of achievement should be measurable and can be increased through performing projects.
- **InfrastructureService**: Infrastructure services are technical functionalities that are based on solution elements. They can be used by deployed application systems through well-defined interfaces.
- **Interconnection**: An interconnection is a link between two entities that supply business support (cf. *SupportProvider*), e.g. deployed application systems. One support provider, the *server*, supports the other support provider, the *client*, to help in delivering business support. Additionally, the interconnection's attribute *ConnectorType* defines the type of interconnection. Possible values are *online*, *offline*, and *manual*.
- **Location**: A location is a geographical place where an organizational unit is located.
- **OrganizationalUnit**: Organizational units are subdivisions inside an enterprise based on the internal structure. Deployable elements are hosted by these units.

- **Person**: In this context a person is a human being that can issue demands or can take over responsibility for projects.
- **PlanningVariant**: Planning variants describe a group of interrelated projects that pursue the same future application landscape scenario. Differing planning variants refer to different future scenarios of application landscape and business support.
- **Project**: Projects are a set of activities that are performed to adapt the application landscape. They can pursue one to multiple demands. Projects include attributes such as start and end date, project status, costs or other economic values. Moreover, they can be linked to one planning variant.
- **Service**: This object type refers to service oriented architectures. Opposing to deployed application systems, a service is an implementation independent function that can be accessed by several business services. A service level agreement contains detailed information about the service.
- **SolutionElement**: Actual implementations of blueprint elements are realized through solution elements. They implement a particular technology. Included attributes provide information about life-cycle phases and support end date.
- **Strategy**: A strategy is a long term plan that aims at achieving the enterprise's vision and mission. It can be expressed through goals.
- **SupportProvider**: A support provider is an abstract entity that supports a business process at an organizational unit. Potential instances of support providers are deployed application systems and services. Attributes that are also inherited by instances include information about life-cycle (*in planning*, *in development*, *in production* and *in retirement*), business criticality, version and number of users.
- **SupportRelationship**: This object type describes the relationship between supported business process, support provider and organizational unit. Moreover it is stated which functions and thereby, which operations, are performed on business objects. Instances of *SupportRelationship* are affected by changes of the application landscape.

During the EAM Tool Survey 2008 some exemplary data was created that reflects this information model. It is based on a fictional department store called *SoCaStore*. This data is also used in the tool evaluation of this thesis.

5.2. Scenario Description

In this thesis EAM tools are evaluated by analyzing 17 scenarios. The scenarios are taken from the Enterprise Architecture Management Tool Survey 2008 [Ma08] and are based on the experience from previous projects executed at the *sebis* chair at TU München. A list of relevant EAM criteria and input from practitioners was used to create these scenarios [Er06]. Further explanations about the creation process can be found in the corresponding report.

The scenarios can be split up into two groups. Eight scenarios simulate specific tool functionality and represent requirements that are not directly related to EAM tasks but can contribute to their fulfillment. Nine scenarios consider requirements that need to be met while performing typical EAM tasks and EAM sub-processes.

5.2.1. Scenarios for Analyzing Specific Functionality

A list of all scenarios analyzing specific functionality is presented in table 5.1.

Scenario
Importing, Editing, and Validating Model Data
Creating Visualizations
Interacting with, Editing of, and Annotating Visualizations
Communication and Collaboration Support
Flexibility of the Information Model
Support of large scale Data
Reporting
Usability

Table 5.1.: Scenarios for Analyzing Specific Functionality

Importing, Editing, and Validating Model Data

This scenario analyzes importing and export functionality. Exemplary data from Microsoft Excel spreadsheets is imported into the tool repository and other import sources are identified. Automatic data verification and consistency is checked. Moreover, it is analyzed how data can be edited manually and through a potential mass update feature. Adding links or documents about further documentation to object instances is another part of this scenario.

Creating Visualizations

Supported data visualization types are examined. Automatic generation and manual creation of visualizations are tested. Exemplary visualization types are cluster maps, process support maps, portfolio matrices or time interval maps. The Enterprise Architecture Visualization Tool Survey 2014 [RZM14] provides detailed descriptions about visualization types. Additionally, customizing capabilities and flexibility of existing visualization types are looked at.

Interacting with, Editing of, and Annotating Visualizations

Interacting with and editing existing visualizations is focused in this scenario. Additional information shall be considered in visualizations, e.g. through annotations or extended views. Furthermore, it is analyzed how the appearance of visualizations can be adapted, e.g. object movements, resizing and zoom functionality, highlighting of selected information, or filtering possibilities. Another point that is looked at is how data that is changed in visualizations is handled in the data repository.

Communication and Collaboration Support

In this scenario communication and collaboration features are examined. Functionality about user management with corresponding user roles and user access control is validated. It is studied if the tool supports collaborative work among multiple users. Examples are versioning of artifacts to avoid conflicts, automatic notification processes, comment functionality, discussion platforms, tagging services, or other Enterprise 2.0 platform functionality. As the focus of this thesis lies on collaborative EAM tools, this scenario is extended to the original one in the EAM Tool Survey 2008.

Flexibility of the Information Model

This scenario analyzes flexibility and adaption options of the information model. It is checked to which extend the previously described information model can be mapped to the tool and how the information model can be adapted after data has been imported. Besides editing attribute and relationship types, new object types shall be added. Adaption of graphical representation of object types and the possibility to export and (re-)import information models is also checked.

Support of large scale Data

The tool's capabilities to support large scale data are analyzed in this scenario. A large amount of application systems is imported into the tool and performance and convenience of tool use are studied. Moreover, finding, editing and visualizing large scale data is examined. It is checked whether the tool's response time is still acceptable with a large amount of data and if the tool's performance is perceived as fluent and quick.

Reporting

The tool's reporting capabilities are examined. Predefined reports are looked at and customization options to change the report design, represented data, and integrated KPIs. Potential report output formats are determined. Additionally, available possibilities to support impact analysis (what-if scenarios) are analyzed.

Usability

This scenario examines the tool's usability. It is analyzed how user-friendly and comfortable the tool is designed, and how easy new users can utilize the tool. Furthermore, available tool documentations and user help systems are considered. This examination is very subjective and influenced by the author's personal opinion. Other users might perceive different experiences.

5.2.2. Scenarios for Analyzing EA Management Support

The following group of scenarios simulate typical EAM tasks. The application landscape is the central artifact of these scenarios. There exist different landscape versions. The *current* landscape represents the as-is status, the *planned* landscape refers to a future to-be state at a specific point in time, and the *target* landscape describes a long-term envisioned state. There can be multiple planned landscapes representing different points in time and planning variants. The target landscape is not an actual architecture to be reached but depicts an architectural state current and planned application system can be based on. Table 5.2 summarizes scenarios that are applied to analyze tool support for EA management tasks.

Scenario
Landscape Management
Demand Management
Project Portfolio Management
Synchronization Management
Strategies and Goals Management
Business Object Management
SOA Transformation
IT Architecture Management
Infrastructure Management

Table 5.2.: Scenarios for Analyzing EA Management Support

Landscape Management

The scenario landscape management is concerned with the documentation of current, planned and target IT landscape. Moreover, the landscape evolution from current to planned state is analyzed. Landscape states shall be compared and differences identified. In order to reach a planned state, it shall be determined which necessary projects need to be performed. Moreover, different views of the application landscape shall be created and maintained.

Demand Management

Demand management is about gathering and documenting requirements and business demands that shall be pursued in projects. Furthermore, demands shall be linked to affected EA elements. Similar demands can be combined to be pursued in one project proposal. It is analyzed how demands and their relationships can be managed and visualized in the tool.

Project Portfolio Management

In this scenario the process of managing project portfolios is simulated. Project proposals are gathered in the tool and linked with affected EA elements. Attributes like budget, economic impact or strategy alignment shall be considered. The project portfolio shall be analyzed to select projects to be pursued. It is examined how the tool reflects projects and how the project selection and planning process is supported. An important visualization to be created is the project portfolio matrix which reflects return on investment, costs, impact, and other attributes in one chart.

Synchronization Management

Synchronization management focuses on dependencies between projects and affected objects like application systems or organizational units. This scenario is linked with project portfolio management. It is examined how the tool is able to reflect and portray project delays. One outcome is that dependent projects should be postponed. Moreover, it is analyzed how conflicts caused by limited resources or dependencies can be considered in the tool.

Strategies and Goals Management

This scenario analyzes how business strategies and goals are reflected in the EAM tool. Measurable goals shall be linked to organizational units. The tool shall provide representations to visualize how EAM supports and is aligned with business strategy. One process is to break down the strategy into smaller pieces and to track by which organizational units and applications these small strategy steps are fulfilled.

Business Object Management

The tool's capability to model business objects and their use in the enterprise are examined in this scenario. Operations performed on business objects and the exchange of them shall be reflected in the tool. The flow of business objects between application systems shall be modeled and visualized. In addition, the corresponding business processes where business objects are used shall be depicted.

SOA Transformation

In this scenario the tool's capabilities to support in transforming an architecture into a service-oriented architecture are analyzed. Potential future SOA landscapes shall be modeled. By performing top-down or bottom-up analysis potential application systems that could be replaced by services shall be identified. The effects a SOA transformation would cause on the application landscape shall be presented by the tool and attributes like service level agreements shall be integrated. Moreover, affected and dependent applications, business processes and infrastructures shall be detected.

IT Architecture Management

IT architecture management is about managing architectural blueprints that represent the target architecture. The application landscape shall be standardized. This scenario analyzes how tools reflect architectural blueprints, target architectures and homogenization of technologies. Existing architectures and used technologies shall be visualized. Existing applications shall be adapted to conform to and new applications shall be based on blueprints.

Infrastructure Management

This scenario simulates infrastructure management which is about the application of infrastructure elements and technologies. The tool's capabilities to answer questions about used technologies, infrastructure release planning and vendor support are analyzed. Operating and license costs of infrastructure shall be considered as well as organizational units that host infrastructure elements. Furthermore, information about life-cycle status and applications that use infrastructure elements shall be provided.

Part III.

Application and Evaluation

6. Scenario Simulation in leanIX

LeanIX is a rather new tool developed by the German company LeanIX GmbH. The vendor emphasizes on its website the collaboration features, a simple user interface and reduced complexity. In this chapter leanIX is evaluated. The aim of this evaluation is to verify if leanIX can be categorized as a collaborative EAM tool. Therefore, leanIX's collaborative characteristics and capabilities to support EAM tasks are analyzed. The collaborative characteristics identified in section 4.1 are checked. Furthermore, the aforementioned information model from section 5.1 is employed and the scenarios from section 5.2 are simulated.

The evaluation is based on the online trial demo version 1.10.0 that can be tested free of charge for 4 weeks.[1] It is provided as a web-based service and can be accessed through the browser.

6.1. Evaluation of Specific Functionality

In this section specific EAM functionality of leanIX are evaluated.

6.1.1. Importing, Editing and Validating

LeanIX offers the possibility to import objects through predefined Microsoft Excel templates. After data have been inserted into a spread sheet, they can be validated by performing a test import, or be imported directly into the system. When performing the test run, a detailed spreadsheet is generated that shows which objects could be successfully imported or which errors occurred.

Data can be exported from the system into Microsoft Excel spreadsheets. In these generated spreadsheets mass updates can be performed for a limited amount of attributes like 'Name', 'Description', 'Lifecycle' or 'Release'. After performing a mass update in Microsoft Excel, data can be again imported into leanIX. Unfortunately, interfaces and other important object attributes cannot be set via mass updates but need to be changed in every object manually.

Objects can be edited in leanIX in the 'Inventory' tab. Each attribute can be edited either by entering individual text or using drop-down menus that show potential input options (cf. Figure 6.1).

Additional documents can be linked to each object by adding an URL and a description. Files cannot be uploaded from a local device but need to be provided through an URL.

[1] Online trial available at `https://leanix.net/en/site/contact?type=trial`, accessed on June 2nd, 2014.

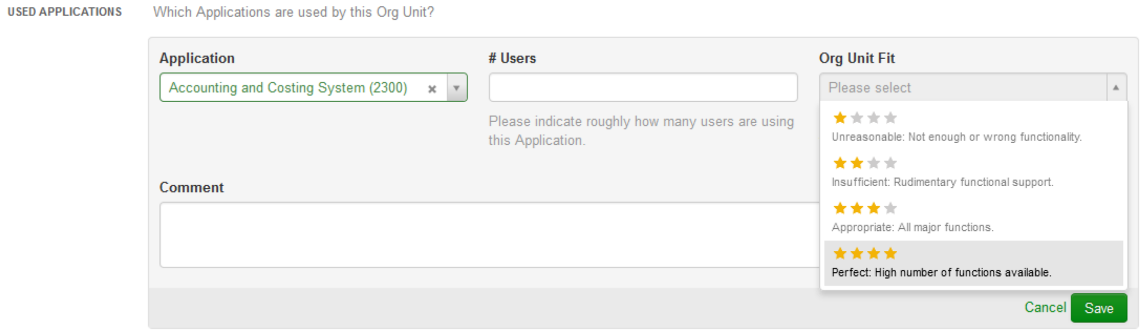

Figure 6.1.: LeanIX - Edit object

6.1.2. Creating visualizations

There are multiple visualizations existent in leanIX that are grouped into the categories *'Metrics'*, *'Cost'*, *'Heat map'* and *'Roadmap'*.

- The *Metrics* category offers portfolio matrix visualizations for applications, projects and providers (cf. Figure 6.2).

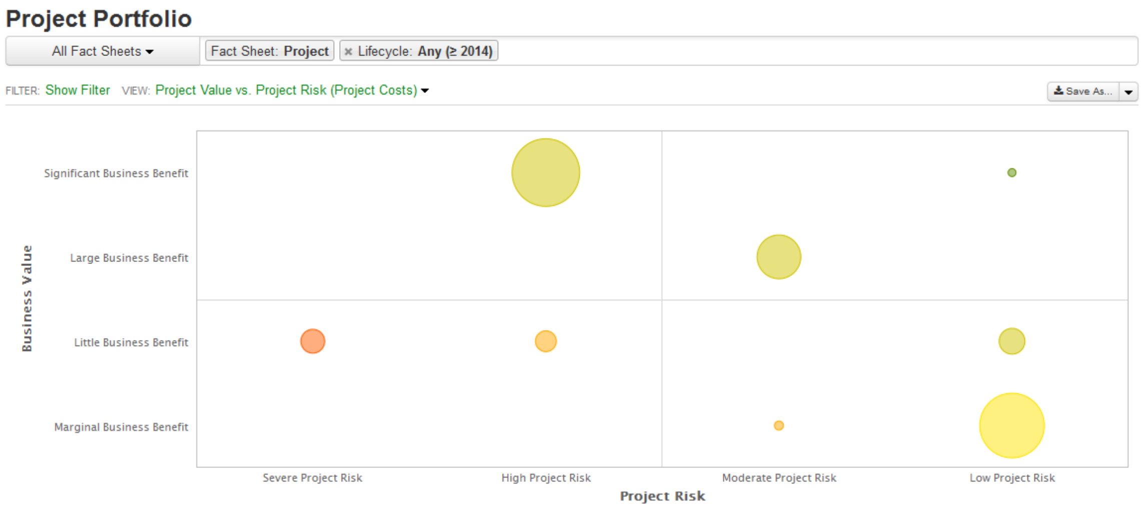

Figure 6.2.: LeanIX - Project Portfolio Matrix

- The *Cost* category offers bar charts for business capabilities, pie charts for provider costs, and a project cost matrix (cf. Figure 6.3).
- In the *Heat map* category one can generate cluster maps for applications (cf. Figure 6.4), interfaces and infrastructures as well as location maps for application usage, application sourcing and infrastructure locations.
- Time interval maps are available in the 'Roadmap' category for applications, projects (cf. Figure 6.5) and infrastructure.

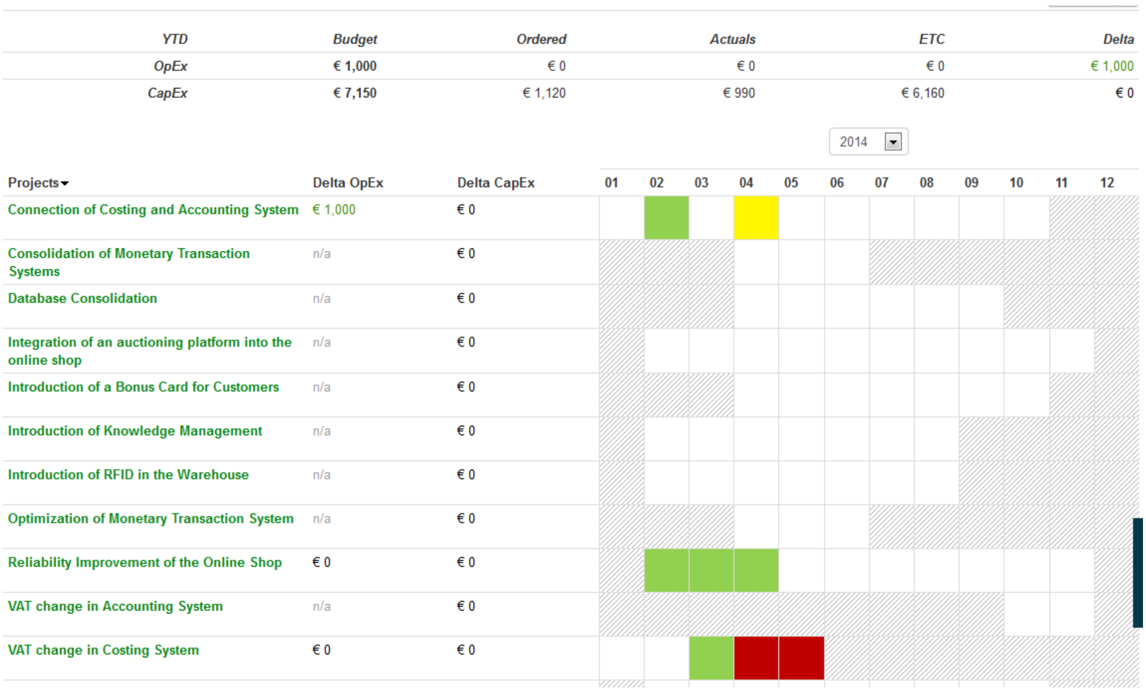

Figure 6.3.: LeanIX - Project Cost Matrix

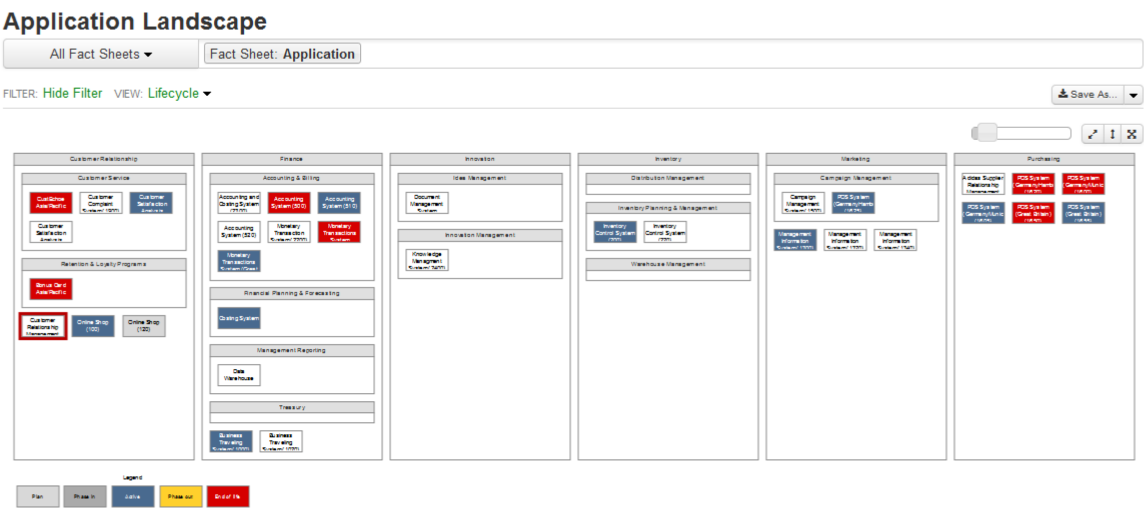

Figure 6.4.: LeanIX - Application landscape

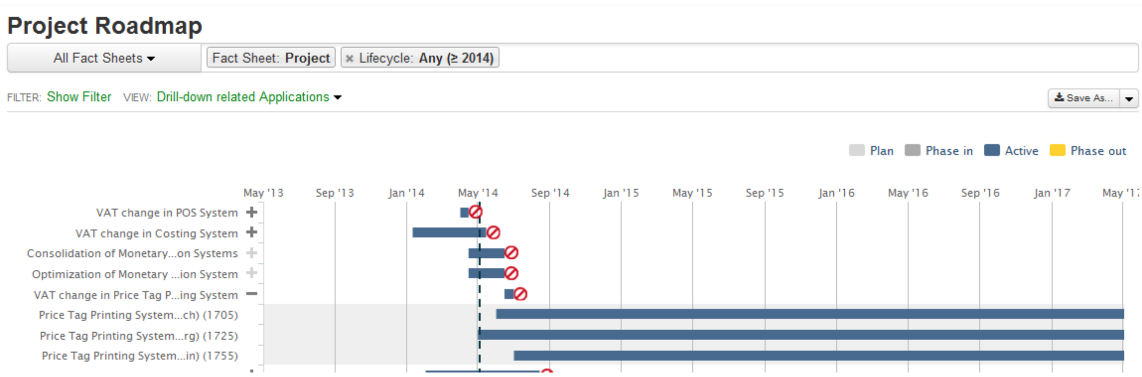

Figure 6.5.: LeanIX - Project roadmap

Visualizations are generated automatically and displayed as interactive elements in the browser. The object types to be displayed are predefined per visualization and cannot be changed. Creating visualizations manually is not possible.

6.1.3. Interacting with, Editing of, and Annotating Visualizations

Adaptions of generated visualizations can be made through activating or deactivating filters. Filters can be activated according to attributes like time interval, lifecycle, organizational unit and many more.

It can be set whether objects to be displayed have to fulfill all selected conditions (AND) or at least one (OR). Activated filters are displayed at the top of the visualization and can be deactivated there easily. Moreover, the color coding of objects can be changed in some visualizations. For example, in the application landscape diagram application colors can be adapted based on lifecycle, project status or functional/technical fit.

After regenerating diagrams, earlier applied filters are deactivated. In the evaluated version the only way to save filtered visualizations is to save them as bookmarks in the browser. With the help of a bar one can zoom into visualizations to get a detailed view (cf. Figure 6.6).

Groups of objects can be selected, e.g. applications belonging to the same business domain. A list of these applications is then displayed below the visualization. Through these lists, specific objects, e.g. affected applications, can be accessed directly.

6.1.4. Communication and Collaboration Support

The tool vendor, LeanIX GmbH, states that they put special attention on collaboration while developing their software. There are three kind of user roles one user can get assigned in leanIX: *Viewer*, *Member* and *Admin*. Viewers can view and subscribe to all objects. Members can modify and create new objects, and admins can manage other users and export data into other environments.

Every user can subscribe to objects as *Responsible* or *Observer* to get directly informed if any changes are made on the subscribed object. A subscription has the advantage that each user can filter for the objects he or she is interested in and one can directly contact

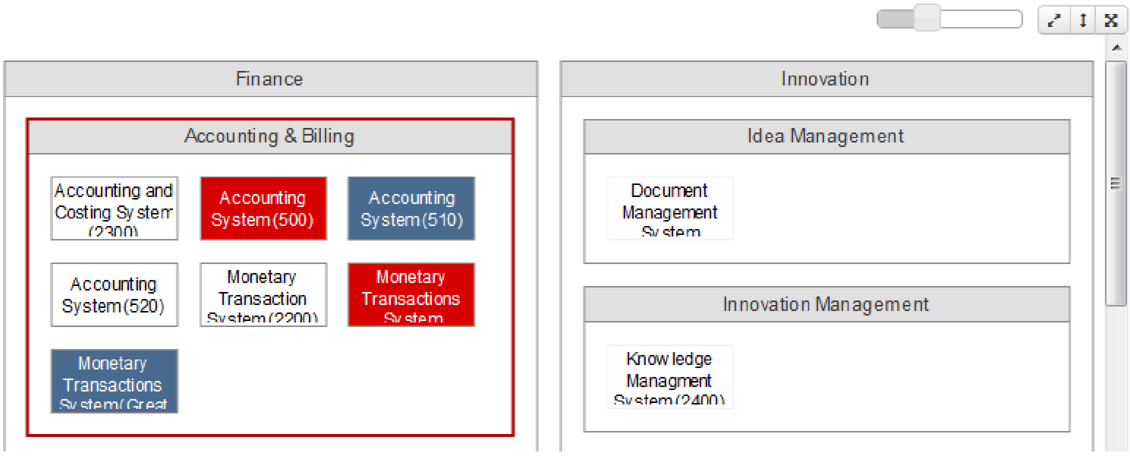

Figure 6.6.: LeanIX - Detailed view through zoom bar

those responsible via e-mail. One can get a quick overview about all subscribed objects. Moreover, visualizations can be adapted that only subscribed objects are included.

It is possible to comment on every object and to open new topics, e. g. users can ask specific questions or provide feedback. Other subscribed users get notified and can reply to this open topic (cf. Figure 6.7). This feature facilitates discussions and collaboration.

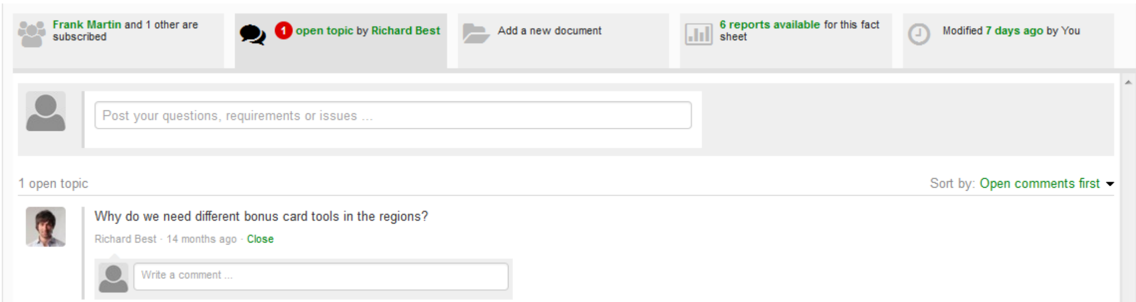

Figure 6.7.: LeanIX - Open topics

Furthermore, leanIX integrates a tagging service that allows users to structure content according to individual categories. Tags can be assigned to every object instance on-the-fly. In this way objects can be categorized and grouped quickly. It is possible to search for objects with specific tags and use tags to filter visualizations. Objects can be added to existing tag categories through an auto-completion feature. Figure 6.8 shows a snapshot that was taken while tagging an application system.

After logging in, the user gets redirected to a newsfeed where recent changes and updates are displayed (cf. Figure 6.9). There is a filter possibility to filter for news about subscribed objects, newly created objects or open topics. Users can subscribe to the newsfeed via RSS feeds to access it from other locations.

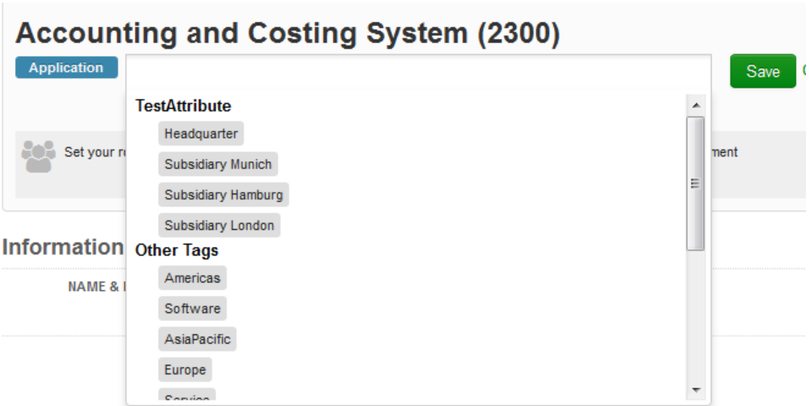

Figure 6.8.: LeanIX - Tagging an application

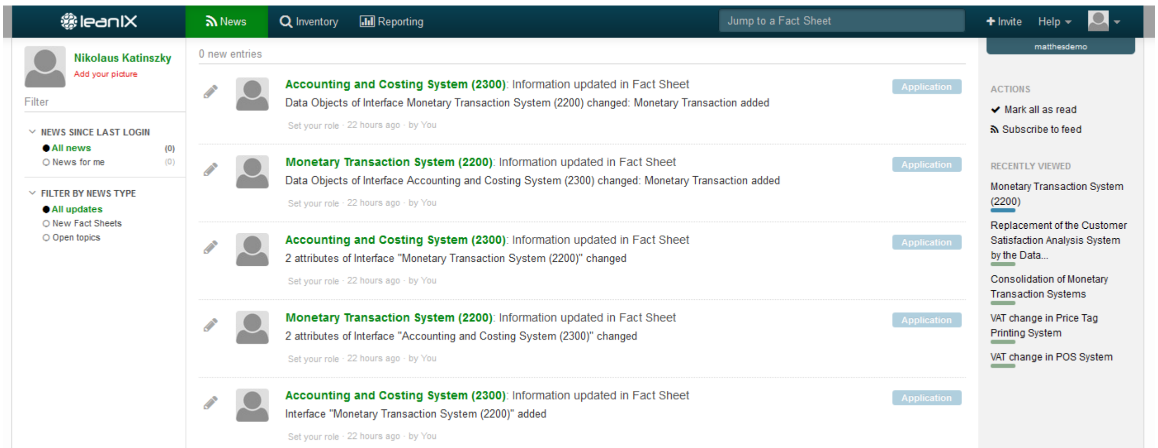

Figure 6.9.: LeanIX - Newsfeed

6.1.5. Flexibility of the Information Model

LeanIX brings a fixed information model where additional object types cannot be created. The names of existing object types can be changed but semantics remain the same.

New custom fields can be added that can be assigned to object types. Different values can be defined for custom fields. These values can then be linked to object instances by tagging the object instance with a specific value. The resulting tags can be used in visualizations for filtering and color highlighting (cf Figure 6.10).

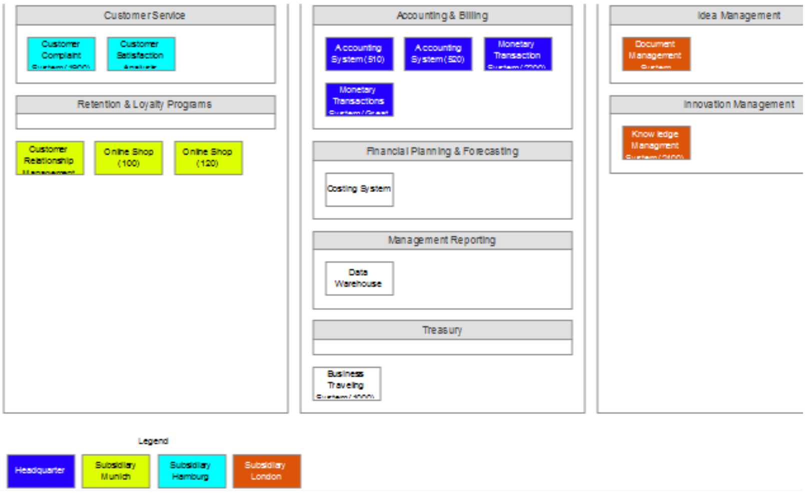

Figure 6.10.: LeanIX - Visualization with highlighted custom values

6.1.6. Support of large scale Data

During the evaluation leanIX works smoothly and quick with a small amount of data of around 40 applications and corresponding relations and interfaces. The import of a bigger amount of applications takes a few minutes but is successful. Unfortunately, the mass update feature is very limited and only allows to mass-update few attributes. Important attributes like interfaces or location need to be updated manually by the user in leanIX which can be very exhausting.

The infinite scrolling implementation of object lists is helpful for large amount of data because only the data that is currently viewed by the user is loaded. If the user scrolls through a list, the list gets extended automatically. In this way, the user does not notice that there is that much data in the system. The search field is also very useful to access one specific object quickly, no matter how many objects exist.

The overview in visualizations is limited with a large amount of data. Either text labels are difficult to read or the user has to scroll a lot manually inside the visualization to get text labels illustrated in a proper size. Performance issues cannot be noticed.

6.1.7. Reporting

There are no tabular reports available in leanIX. The *Reporting* tab in leanIX only includes visualizations. Creating custom reports or visualizations is not possible. Only predefined visualizations where object types cannot be changed are available. Below each visualization there is a list of selected objects. This list links to detailed information.

Generated visualizations can be filtered for different attributes. Some visualizations offer a drill-down functionality. For example, projects can be drilled down to see affected applications. Some visualizations are especially useful for reporting. The project cost tracking visualization (cf. Figure 6.3) illustrates the status of all projects as well as planned and current costs. Filtered and adapted visualizations can be saved as local browser bookmarks. They can be exported as PDF files.

6.1.8. Usability

LeanIX is easy to use. New users get to know the application very quick. The tool has a clear structure that always provides a good overview. There are only three main menu items where every functionality can be accessed.

The '*Jump to Fact Sheet*' field with search and drop down selection is especially useful to access objects quickly. The newsfeed is helpful for the user to get a quick overview about all changes and events. The tagging functionality allows quick grouping.

The infinite scrolling implementation where additional content at the bottom of the page is loaded automatically creates a modern, lean look and feel.

The leanIX documentation is accessed through a web browser. The level of detail is limited but the most important questions are answered.

6.2. Evaluation of EA Management Support

In this section leanIX's EA Management support capabilities are evaluated based on the previously explained scenarios (cf. section 5.2.2).

6.2.1. Landscape Management

LeanIX supports landscape management through predefined visualizations and application attributes. The application landscape diagram (cf. Figure 6.4) allows to filter for different points in time. The application attribute 'Lifecycle' can be highlighted according to the application status: *Plan*, *Phase in*, *Active*, *Phase out* or *End of life*. A value to define a special target landscape is not available but could be realized through assigning applications manual tags.

The visualization Interface map is a cluster map that represents relations between applications. The interface type can be highlighted through color coding. Color coding can be applied for other relevant application attributes like business criticality, functional or technical fit, project status or technology risk.

The visualization Application roadmap (cf. Figure 6.11) is useful to see when applications were or will be exchanged by other applications. There is a drill down functionality to see successor applications, child applications, related projects or IT components.

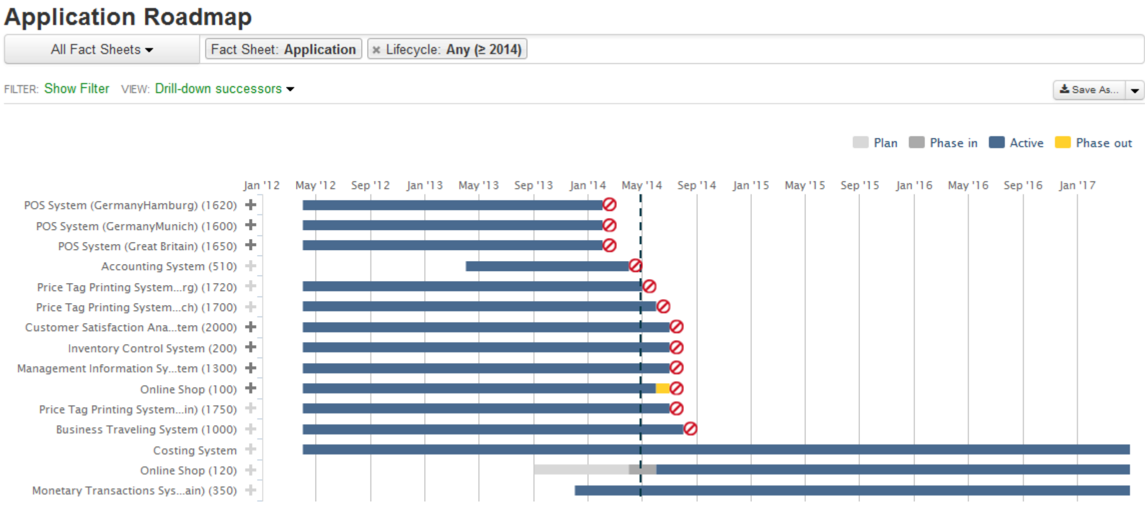

Figure 6.11.: LeanIX - Application roadmap

The given filtering functionality in all visualizations can also be useful for landscape management. In this way, applications can be filtered for specific criteria like business processes, used IT components, lifecycle, manually assigned tags and other attributes.

6.2.2. Demand Management

There is no object type demand or requirement integrated in leanIX. The manual tagging functionality could be used to reflect demands. Applications or projects could then be tagged with a demand the corresponding objects try to meet. These manually assigned demands can also be highlighted through color coding in visualizations. Another option is that demands are expressed via comments through the *Open topic* functionality.

Applications include an attribute called 'Functional fit' that reflects how well one application meets business requirements. A number between 1 and 4 can be assigned to this attribute.

6.2.3. Project Portfolio Management

There is an object type 'Project' integrated into leanIX' information model. This object type has the following predefined attributes: lifecycle, project value, project risk, affected applications, budget and actual costs, involved providers and their corresponding costs. Moreover, a project status can be updated: it can be set to green, yellow or red, depending on the project situation. Current costs as well as a cost estimation can be entered. These maintained attributes can then be used in visualizations to perform project portfolio management inside leanIX.

The project portfolio matrix (cf. Figure 6.2) helps to compare projects with each other based on project value, project risk and costs. It is useful to select between projects that shall be performed and others that better be dropped.

The project cost matrix (cf. Figure 6.3) is useful to track the project status over time as well as planned and actual costs.

A project roadmap visualization (cf. Figure 6.12) illustrates projects on a time interval map. Successor projects or related applications can be integrated through the drill down functionality.

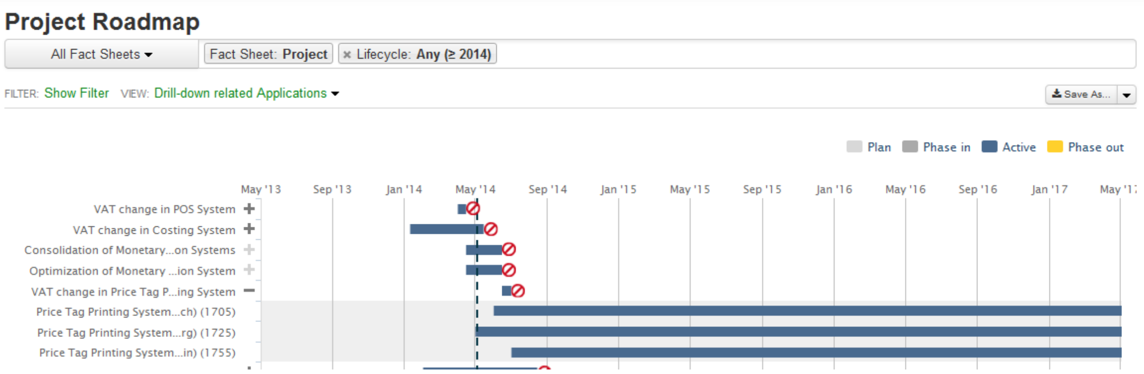

Figure 6.12.: LeanIX - Project roadmap

Filtering options are available in all visualizations in order to filter for specific project attributes, affected applications providers or manually assigned tags. All these features and visualization options are helpful for performing typical project portfolio management tasks like project selection, time and cost tracking or the identification of dependencies.

6.2.4. Synchronization Management

Synchronization management can be performed in leanIX by using the visualization project roadmap (cf. Figure 6.12). Applications that are affected through projects can be visualized. Manual annotations or adaptions are not possible. Business units or other EA elements that are affected through particular projects cannot be mentioned in the project detail page. There are only relations between projects and applications available. Successor and predecessor projects can be named. These connections do not consider project delays.

6.2.5. Strategies and Goals Management

The object types 'Strategy' or 'Goal' are not available in leanIX. LeanIX focuses on IT-relevant areas. Some object type attributes can be used for strategic management: the application criticality can be useful for strategic decisions as well as project value, project risk and the corresponding project portfolio matrix (cf. Figure 6.2) may be relevant. Another interesting part for the IT-Strategy could be the object type 'Provider' where hardware and software providers can be integrated. There is an interactive pie chart visualization available that compares the provider costs (cf. Figure 6.13).

6.2.6. Business Object Management

Business object management is supported in leanIX through the object type 'Data Object'. Applications and operations that are performed on these objects can be set. Moreover, the

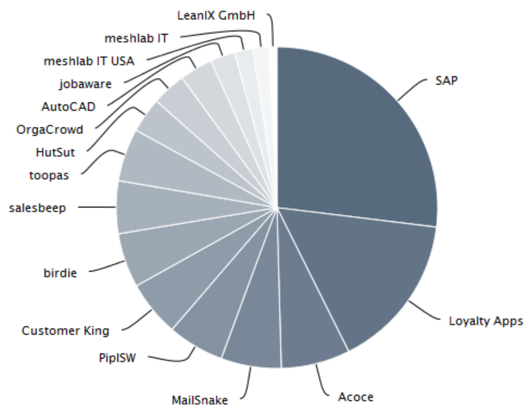

Figure 6.13.: LeanIX - Provider cost visualization

interface that the business object is transferred through can be specified. The interface direction, frequency, type and technology can be named.

Visualization options are limited. A filter can be set that only applications are displayed that use particular data objects inside the application landscape visualization and inside the interface map. The actual data object itself is not visually represented.

6.2.7. SOA Transformation

LeanIX does not offer any business or application service object types. An interface map can be used to identify potential applications that could be replaced with services. Filter options and the application roadmap (cf. Figure 6.11) can be used to see the application development over time. The application landscape diagram (cf. Figure 6.4) illustrates which applications support which business capabilities. The filter can be used to focus on specific business processes, locations or other relevant criteria. There are no layered views or reports available in leanIX.

6.2.8. IT Architecture Management

Architectural blueprints can be managed in leanIX through the predefined object type 'Technical Stack'. Children or parent objects can be linked to build structures. Blueprints that use same architectural elements are difficult to model because one element can only be linked with one parent object. IT components can be assigned to technical stacks. In this way, it is possible to look up by which technical parts architectural blueprints are realized.

The IT component diagram (cf. Figure 6.14) can be used to see which IT components rely on which blueprints. The visualization can be filtered to focus on specific blueprints or technologies. A time dimension filter is useful to see the development over time. The lifecycle attribute can be highlighted through color coding.

Application objects can use IT components. Through this relation it is possible to look up which blueprint they follow.

Figure 6.14.: LeanIX - IT component diagram

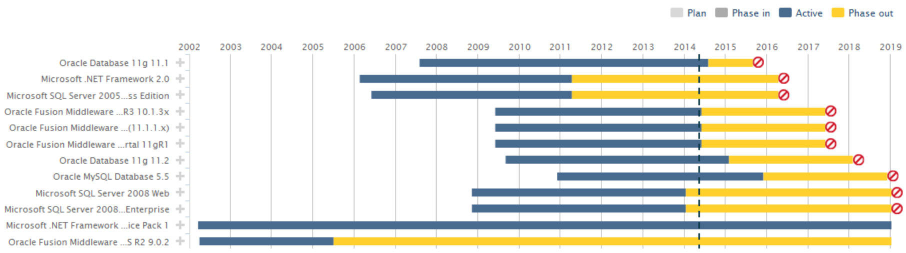

Figure 6.15.: LeanIX - IT component roadmap

6.2.9. Infrastructure Management

IT components in leanIX provide different attributes that support infrastructure management. They can be linked to a technical stack which can be regarded as an architectural blueprint. It can be stated which applications use the IT component. The attribute 'Technical fit' reflects the IT component's shape from a technical view. Moreover, a provider, lifecycle status and location can be set.

The provider cost pie chart (cf. Figure 6.13) can be used to see cost that infrastructure providers cause. The IT component roadmap (cf. Figure 6.15) is helpful see development over time. It can be analyzed when one component shall be replaced or when license support ends. Moreover, affected applications can be integrated.

7. Comparison of EAM Tools

The same scenario simulation described in detail for leanIX in chapter 6 has been performed for three additional EAM tools. In this chapter these evaluations are described and all four tools are compared regarding different aspects such as functionality and collaborative characteristics.

To select a set of evaluation objects a long list of existent EAM tools was created. As available resources are limited, this list was reduced until a reasonable amount of EAM tools remained. Another criterion for tool selection was diversity. Variety among the evaluated tools should be established to identify new findings. Therefore, tools from the Gartner Magic Quadrant (cf. Figure 2.8) were selected that are classified into different types. The last criterion why tools were dropped from the list was availability. Some tools could not be evaluated because vendors could not provide an evaluation version. At the end of this selection process three EAM tools remained for evaluation. These tools are:

- **Bizzdesign Architect**, an EAM tool developed by the Dutch company Bizzdesign. It is positioned as *visionary* in the Gartner Magic Quadrant. It implements the ArchiMate modeling language and is ArchiMate 2.1 certified. Bizzdesign Architect is provided through a fat client installation. An object repository can also be stored on a central server that is accessed by clients.

- **Iteraplan**, developed by the German company Iteratec, is called a *niche player* by Gartner. Because of its German company roots, iteraplan is mainly used in the German speaking market. It is available in two different versions: community and enterprise edition. The community edition is available as open-source software but only offers a limited set of features. The evaluation was performed using an online trial demo that offers most of the enterprise edition features and can be accessed through a web browser.

- **SAP Powerdesigner** is developed by Sybase, a company belonging to SAP. Gartner positions SAP Powerdesigner as a *leader* in its Magic Quadrant of EA. It is a modeling tool with differing available model types. Besides EA, other modeling tasks such as business process modeling or software design modeling can be performed. SAP Powerdesigner can be used via a local installation. A central repository can also be accessed over the network.

A summary of the scenario evaluations is provided in section 7.1. As this thesis focuses on collaborative EAM, a detailed analysis regarding collaborative EAM tool characteristics integrated into the evaluated tools is presented in section 7.2. The identified tool categories from sections 2.2.2 and 4.1 are reused to categorize the evaluated tools (cf. Section 7.3).

7.1. Summary of Scenario Evaluations

Documenting the evaluation results in the same extent as leanIX is evaluated in chapter 6 would go beyond the scope of this master's thesis. Therefore, the results are summarized in Table 7.1. Detailed evaluation results might be provided in future publications.

Harvey balls are used to visualize the degree of fulfillment. A quarter-full harvey ball (◔) says that only a small amount of the scenario's requirements are fulfilled and the majority of requirements is not realized. The half-full harvey ball (◑) denotes a partly implementation. Some requirements of the scenario are met while others cannot be accomplished. A full harvey ball (●) means that all requirements and sub-tasks of the scenario can be fulfilled in the tool.

Scenario	**Bizzdesign Architect**	**Iteraplan**	**leanIX**	**SAP Powerdesigner**
Importing, Editing, and Validating Model Data	◑	●	◑	◔
Creating Visualizations	●	●	◑	◔
Interacting with, Editing of, and Annotating Visualizations	●	◔	◔	●
Communication and Collaboration Support	◔	◑	●	◔
Flexibility of the Information Model	◑	◔	◔	◑
Support of large scale Data	◑	●	◔	◔
Reporting	●	●	◔	◑
Usability	◑	◑	●	◔
Landscape Management	●	◑	●	◑
Demand Management	●	◔	◑	◔
Project Portfolio Management	◑	●	●	◔
Synchronization Management	◔	◑	◔	◑
Strategies and Goals Management	●	◔	◔	◔
Business Object Management	◑	●	◑	◔
SOA Transformation	●	◔	◔	◑
IT Architecture Management	●	●	●	◑
Infrastructure Management	●	●	●	◑
◔ = barely implemented, ◑ = partly implemented, ● = fully implemented				

Table 7.1.: Summary of EAM tool evaluations

Bizzdesign Architect provides strong visualization and reporting capabilities as well as good landscape, SOA transformation, IT architecture and infrastructure management support. Moreover, ArchiMate's strategy and motivation extensions that are implemented in Bizzdesign Architect lead to proper strategy and demand management assistance.

Iteraplan offers good data import and editing features with mass-update functionality. A lot of visualization types can be created with many configuration possibilities. Additionally, advanced reporting features including a separate query language (iteraQL) and strong support for portfolio, IT architecture and infrastructure management are available.

Compared to the other three tools, **leanIX** convinces with its collaboration characteristics (cf. Section 7.2). The amount of available visualizations in leanIX and corresponding configuration options are limited. Nevertheless, landscapes, IT architecture and infrastructure can be managed sufficiently. Thanks to detailed attributes and visualizations about project state and development, project portfolio management is especially well supported.

SAP Powerdesigner is a tool solely for modeling. Models need to be drawn manually and thus, lots of editing and annotation options are offered. As this tool focuses on modeling, its support for functions not related to modeling is limited.

A more detailed analysis of the collaborative capabilities of each evaluated tool is performed in the following section.

7.2. Analysis of Collaborative EAM Characteristics

In section 4.1 collaborative EAM characteristics were identified. From a theoretical perspective EAM tools that integrate search, link, authoring, tag and signal functionality, and furthermore, provide a lean user interface can be categorized as collaborative EAM tools. These tools foster collaboration and can improve performance. It is now evaluated in detail which tools integrate these characteristics and hence, belong to this category.

LeanIX is accessed over the internet through a browser. A search box with instant search functionality allows to jump to objects quickly. Objects can be shared through URLs and relations between objects can be set with a supporting auto complete feature. Users can add content to objects by opening new discussion topics that other users can respond to. Moreover, they can attach documents or internet links through URLs and hence, refer to other contents. User-defined structures and categories can be created by applying the tagging functionality. Tags can be used for filtering in visualizations and lists. A newsfeed notifies users about recent changes. It is possible to filter for news about objects the user has subscribed to. Moreover, users can subscribe to RSS feeds that incorporate object changes. All functionalities can be accessed easily. Diagrams can be created quickly and contents is visualized without much configuration effort. Thus, the user interface of leanIX can be considered as lean.

Bizzdesign Architect offers the possibility to search for objects and visualizations. This feature is especially useful when lots of data is loaded into the system because keeping track of all objects becomes difficult. A central repository with version control functionality can be used to share models. Linking or sharing objects through URLs is not possible. A potential explanation is that Bizzdesign Architect is locally installed on clients and not accessed over the web. Authoring possibilities are not available. Manual annotations inside visualizations are the only option to provide feedback. Tagging or signal features are not integrated either. Bizzdesign Architect's user interface (cf. Figure 7.1) is rather complex with lots of configuration settings and many different views. It appears a bit bloated sometimes and thus, can not be considered lean.

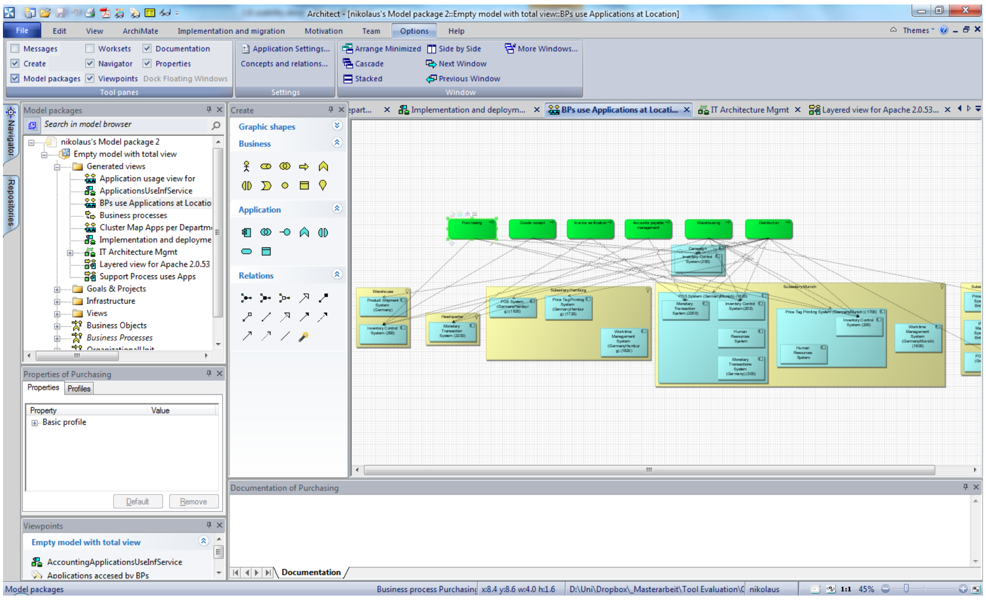

Figure 7.1.: Bizzdesign Architect - User interface

Iteraplan is used through a web browser and all data is stored in a central database. A search box is implemented where full text searches can be performed. Objects can be linked with each other on object property pages. Users can share and access all objects through URLs. Queries can be saved and executed by other users. Authoring features to comment, foster discussions or create blogs as well as tag services are not implemented. Users can subscribe to objects and thus get notified via e-mail when objects are changed. The user interface provides a good overview and all functionalities can be accessed quickly. A snapshot of the user interface is depicted in Figure 7.2. On the other hand, setting up visualizations require lots of configuration effort. Data and visualization settings need to be set up first before visualizations can be generated.

Similar to Bizzdesign Architect, **SAP Powerdesigner** has an integrated search for objects and visualizations. Objects can be stored in a central repository with version control to avoid conflicts. As SAP Powerdesigner is installed locally, users cannot share object links. Authoring, tagging and signal features are missing as well. The user interface does not provide a lean impression (cf. Figure 7.3). Many user actions are necessary to create models. Additionally, some functionalities are implemented insufficiently: the automatic object alignment feature showed inexplicable behavior during the evaluation.

An overview of the evaluated EAM tools and their collaborative EAM characteristics is presented in table 7.2.

In summary, leanIX is the only EAM solution that integrates all collaborative EAM tool characteristics. Therefore, it can be categorized as a collaborative EAM tool. Iteraplan brings some collaborative options and has the potential to be considered a collaborative EAM tool if some features are added in the future. The other two evaluated tools, Bizzdesign Architect

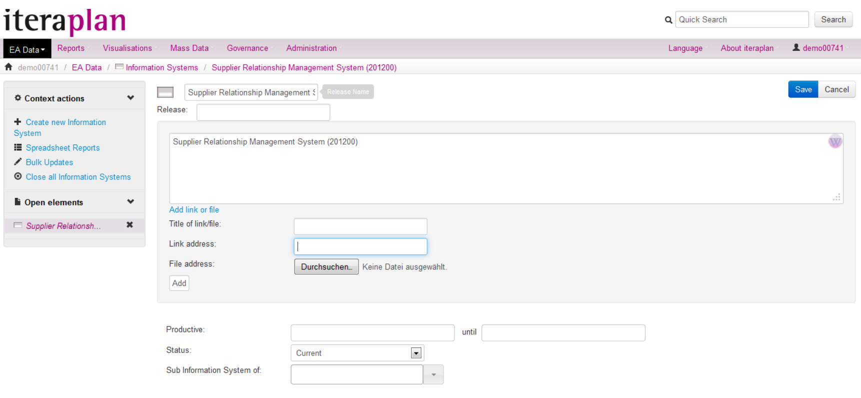

Figure 7.2.: Iteraplan - User interface

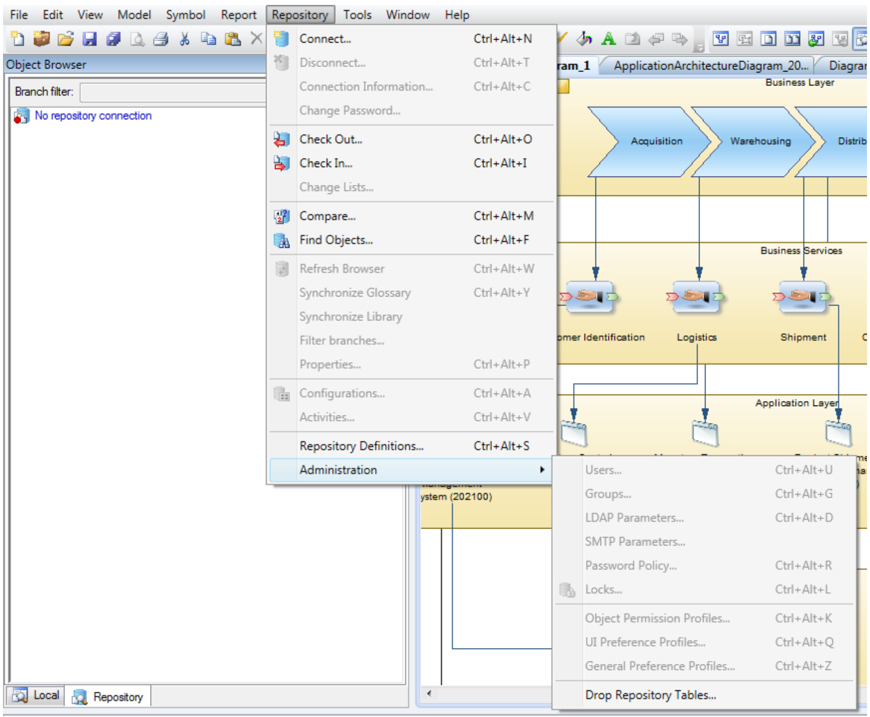

Figure 7.3.: SAP Powerdesigner - User interface

Collaborative EAM Characteristics	Bizzdesign Architect	Iteraplan	leanIX	SAP Powerdesigner
Search	✓	✓	✓	✓
Links		✓	✓	
Authoring possibility			✓	
Tag feature			✓	
Signal feature		✓	✓	
Lean User Interface		(✓)	✓	

Table 7.2.: Collaborative characteristics of evaluated EAM tools

and SAP Powerdesigner, only implement a search feature to improve usability. Apparently, their vendors focus on other characteristics and functionality.

7.3. Categorization of EAM Tools

Besides the summarized scenario evaluation and the analysis on collaborative characteristics, the tools are now put into the categories identified in sections 2.2.2 and 4.1.

The three theoretical dimensions introduced by Keuntje et al. [Ke10] are extended by the dimension *EAM Type*. This dimension includes the categories *traditional EAM* and *collaborative EAM*. Tools that implement the collaborative EAM characteristics explained in section 4.1 are positioned into this category. On the other hand, tools that not include these characteristics but focus on meeting traditional EAM requirements can be called traditional EAM tools. As with the other three dimensions, a clear distinction is not always possible. Tools can include collaborative EAM characteristics and fulfill traditional requirements and thus, can be positioned into both categories.

In table 7.3 the evaluated EAM tools are divided into corresponding categories.

The evaluation of **Bizzdesign Architect** shows that its information model is rather static but lots of configuration and visualization options are available. Thus, it is positioned as a method-centered tool. It is not extendable and can be directly used after installation. Hence, it is an off the shelf solution. Automatic integration of data sources is not possible and data need to be imported manually. Most of the collaborative characteristics are missing. Therefore, it is categorized as a traditional EAM tool.

Iteraplan includes a static best practice information model which can be extended by attributes. Configuration options are available for visualizations and reports. Furthermore, some EAM processes are predefined. Therefore, in the dimension flexibility of the information model it belongs to two categories, method-centered and process-centered tools. Iteraplan can be used off the shelf but also extended and customized thanks to its open

Dimension	Category	Bizzdesign Architect	Iteraplan	leanIX	SAP Powerdesigner
Degree of flexibility	Information model-centered				(✓)
	Method-centered	✓	✓		
	Process-centered		✓	✓	
Customization effort	EAM off the shelf solution	✓	✓	✓	✓
	EAM platform		✓		
Integration approach	Automatic integration				
	Manual import	✓	✓	✓	✓
EAM Type	Traditional EAM	✓	✓	✓	✓
	Collaborative EAM		(✓)	✓	

Table 7.3.: EAM tools positioned into categories

source characteristic. Hence, it is an off the shelf solution and an EAM platform. It belongs to the category manual import as automatic data integration is not implemented. It meets requirements of traditional EAM tools and additionally, includes some collaborative EAM characteristics.

Configuration options in **leanIX** are limited. It is process-centered. LeanIX can be used off the shelf and is not customizable. A manual data import is necessary. This tool fulfills traditional EAM tasks and includes all derived collaborative characteristics. Thus, it can be categorized as a traditional and as a collaborative EAM tool.

There is no underlying information model in **SAP Powerdesigner**. There are just constraints implemented that control which object types may be used in which kind of visualization. New object types can be created. So it is neither method- nor process-centered but offers some flexibility to be categorized as information model-centered. Additionally, the tool has few customization options. It is an off the shelf solution. Data needs to be manually imported. It can be categorized as a traditional EAM tool but apart from a search box, it does not implement any collaborative functionality .

The categorization (cf. Table 7.3) shows that fully flexible EAM tools are rare. Most of the evaluated solutions are method- or process-centered. Every tool that has been looked at comes as an off the shelf solution. Additionally, Iteraplan can be customized. No tool is capable of integrating third party data sources automatically. Every evaluated tool is able to fulfill traditional EAM requirements while only leanIX integrates all collaborative EAM characteristics. With some limitations (cf. Section 7.2) Iteraplan can also be characterized as a collaborative EAM tool.

Part IV.

Conclusions

8. Conclusions and Outlook

This chapter recaps the results of this thesis and summarizes the concept of collaborative EAM tools. The relevance of the findings is discussed and implications are outlined. Finally, an outlook on future research in the area of collaborative EAM tools is presented.

8.1. Summary

The main goals of this thesis were to elaborate collaborative characteristics that can be integrated into EAM tools, and to evaluate existing EAM tools with respect to their collaborative aspects. After demonstrating the importance and current problems in this field, fundamental EA terms and frameworks were introduced. Furthermore, it was presented how software tools can support EAM. The first part of this work finished with a description of the Enterprise 2.0 approach.

In the beginning of the second part shortcomings of traditional EAM were highlighted. The collaborative EAM approach that can reduce complexity, eliminate bureaucracy and improve efficiency, was explained. Specific examples how lean and agile principles as well as Enterprise 2.0 technologies can be integrated into EAM were shown. It was presented to which degree collaborative principles are applied by EA practitioners today. Moreover, it was examined how EAM tools can integrate collaborative aspects. Collaborative characteristics of EAM tools were derived and a new tool category, called collaborative EAM tools, was identified. Afterwards, potential collaborative EAM tools were listed. At the end of the second part a scenario evaluation approach for EAM tools was introduced.

This evaluation approach was used to assess four EAM tools. In part three the evaluation results of the EAM tool leanIX were presented in detail. Moreover, the evaluations of three additional EAM tools, Bizzdesign Architect, Iteraplan and SAP Powerdesigner, were summarized. A comparison of all evaluated tools was performed. Furthermore, collaborative characteristics of the evaluated tools were analyzed. There is only one tool, leanIX, that implements all previously defined collaborative characteristics. Especially authoring and tagging capabilities as well as lean user interfaces are missing in other tools. Finally, these results were taken to position the tools into categories of four different dimensions.

8.2. Discussion

Previous research in this field concentrated on organizational and structural questions regarding the integration of collaborative principles into EAM. Theoretical thoughts and concepts based on practical experience led to collaborative EAM guidelines. Additionally, specific process structures that integrate collaboration into established EAM approaches existed. This work complements existing research by taking into account software tool support for collaborative EAM.

- The identified collaborative tool characteristics can improve EAM performance and, thanks to enhanced acceptance among stakeholders, increase utilization of EAM tools. EAM tool users can benefit from these collaborative functionalities and vendors can use the derived characteristics to identify future tool features.
- The evaluation results indicate that EAM tools offer similar functionalities in supporting traditional EAM tasks. Social software functionality and lean user interfaces are not implemented in most of the tools. These findings suggest that EAM tool vendors focus on core functionality and that collaboration features are not perceived as being essential.
- The integration of collaborative characteristics can make EAM tools distinguishable from others. This finding implies the existence of new market opportunities for EAM tool vendors. In particular, small vendors that are able to adapt and enhance their products quickly can take on this chance and thereby gain market share.
- The evaluation of the collaborative EAM tool leanIX showed that its configuration and adaption possibilities are limited. A possible explanation for this result may be the aim to reduce tool's complexity. It seems to be difficult to unite the two aims *allow user-specific customization* and *reduce complexity*. Therefore, one limitation of collaborative EAM tools may be reduced customization and configuration capabilities.

8.3. Outlook

In this work collaborative EAM characteristics were identified and a selection of EAM solutions were evaluated to analyze the current state of collaboration functionality. Further research can be performed with regards to the following aspects.

- The SLATES functionalities were mapped to EAM tasks in this thesis to verify potential performance impacts. Different collaborative technologies could be used as technology tasks in the applied research process. By utilizing other collaborative functionality, additional task-technology fits could be identified and new collaborative characteristics could be discovered.
- The performed evaluation identified only one EAM tool that integrates all collaborative characteristics. Further EAM products might be available that also integrate these functionalities. By evaluating additional EAM solutions, more collaborative EAM tools may be identified. The tools listed in section 4.2 are potential candidates for this category.
- Limitations regarding customization and configuration capabilities of collaborative EAM tools were identified during the tool evaluation. Further research should be conducted to verify this limitation. Additional collaborative EAM tools could be analyzed regarding customization capabilities. Furthermore, the potential limitation raises the question to which degree customization of EAM tools is necessary and requested from practice.

- The application of collaborative EAM tools in practice has not been examined yet. Empirical studies on the application and usage patterns of collaborative EAM tool functionality in practice could provide further insights. Theoretically derived benefits could be verified, and additional collaborative characteristics and usage scenarios could be revealed.

Bibliography

[AGW11] Aier, S.; Gleichauf, B.; Winter, R.: *Understanding Enterprise Architecture Management Design – An Empirical Analysis*. In (Bernstein, A.; Schwabe, G., Ed.): *Proceedings of the 10th International Conference on Wirtschaftsinformatik WI 2011*. pages 645–654. 2011.

[Am14] Ambler, S.: *Agile and Enterprise Architecture*. `https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/entry/agile_and_enterprise_architecture?lang=en`. 2010, Last checked: 10.06.2014.

[ARW08] Aier, S.; Riege, C.; Winter, R.: *Unternehmensarchitektur – Literaturüberblick und Stand der Praxis*. *Wirtschaftsinformatik*. 4(50):292–304. 2008.

[BBL12] Bente, S.; Bombosch, U.; Langade, S.: *Collaborative enterprise architecture: Enriching EA with lean, agile, and enterprise 2.0 practices*. Morgan Kaufmann. Waltham, MA, USA. 2012.

[Be14] Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J. et al.: *Manifesto for Agile Software Development*. `http://agilemanifesto.org/`. 2001, Last checked: 24.05.2014.

[Bu10] Buckl, S.; Matthes, F.; Roth, S.; Schulz, C.; Schweda, C. M.: *A Conceptual Framework for Enterprise Architecture Design*. In (Proper, E.; Lankhorst, M. M.; Schönherr, M.; Barjis, J.; Overbeek, S., Ed.): *Trends in Enterprise Architecture Research*. volume 70 of *Lecture Notes in Business Information Processing*. pages 44–56. Springer. Berlin and Heidelberg, Germany. 2010.

[Bu11] Buckl, S.; Matthes, F.; Monahov, I.; Roth, S.; Schulz, C.: *Towards an agile design of the enterprise architecture management function*. In *Trends in Enterprise Architecture Research (TEAR) - 6th International Workshop*. 2011.

[Bu12] Buschle, M.; Grunow, S.; Hauder, M., Matthes; F., Roth, S: *Automating Enterprise Architecture Documentation using an Enterprise Service Bus*. In *Americas Conference on Information Systems (AMCIS 2012)*. Seattle, Washington, USA. 2012.

[Co10] Cohn, M.: *Succeeding with Agile: Software development using Scrum*. The Addison-Wesley signature series. Addison-Wesley. Upper Saddle River, NJ, USA and München, Germany. 2010.

[Er06] Ernst, A. M.; Lankes, J.; Schweda, C. M.; Wittenburg, A.: *Tool Support for Enterprise Architecture Management - Strengths and Weaknesses*. In *10th IEEE International Enterprise Distributed Object Computing Conference, EDOC '06*. 2006.

[Ga14] Gartner: *Magic Quadrant for Enterprise Architecture Tools*. https://www.gartner.com/doc/2601526. 2013, Last checked: 01.06.2014.

[Gl10] Gloger, B.: *Scrum: Der Pradigmenwechsel im Projekt- und Produktmanagement – Eine Einführung*. *Informatik-Spektrum*. 33(2):195–200. 2010.

[GT95] Goodhue, D. L.; Thompson, R. L.: *Task-technology fit and individual performance*. *MIS quarterly*. 19(2):213–236. 1995.

[Ha13] Hauder, M.; Roth, S.; Schulz, C.; Matthes, F.: *Organizational Factors Influencing Enterprise Architecture Management Challenges*. In *ECIS 2013 Completed Research*. 2013.

[Ha14] Hauder, M.; Roth, S.; Schulz, C.; Matthes, F.: *Agile Enterprise Architecture Management - An Analysis on the Application of Agile Principles*. In *4th International Symposium on Business Modeling and Software Design (BMSD 2014)*. Grand Duchy of Luxembourg, Luxembourg. 2014.

[Ho09] Hoogervorst, Jan A. P.: *Enterprise governance and enterprise engineering*. The enterprise engineering series. Springer. Berlin, Germany. 2009.

[IE07] IEEE: *ISO/IEC Standard for Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems: ISO/IEC 42010 IEEE Std 1471-2000*. *ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15*. 2007.

[IS14] ISO/IEC/IEEE 42010: *A Conceptual Model of Architecture Description*. http://www.iso-architecture.org/ieee-1471/cm/. 2014, Last checked: 02.06.2014.

[KAV05] Kaisler, S. H.; Armour, F.; Valivullah, M.: *Enterprise Architecting: Critical Problems*. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS '05*. page 224b. 2005.

[Ke10] Keuntje, J. H.; Matthes, F.; Buckl, S.; Schweda, C. M.: *EAM-Werkzeuge*. In (Keuntje, J. H.; Barkow, R., Ed.): *Enterprise Architecture Management in der Praxis*. pages 189–213. Symposion. Düsseldorf, Germany. 2010.

[Ke12] Keller, W.: *IT-Unternehmensarchitektur: Von der Geschäftsstrategie zur optimalen IT-Unterstützung*. dpunkt.verlag. Heidelberg, Germany. 2 edition. 2012.

[La05] Lankhorst, M.: *Enterprise architecture at work: Modelling, communication, and analysis*. Springer. Berlin, Germany. 2005.

[Le99] Lee, Y. T.: *Information modeling: From design to implementation*. *Second World Manufacturing Congress*. pages 315–321. 1999.

[Le14] Leffingwell, L.: *Scaled Agile Framework*. http://scaledagileframework.com/. 2014, Last checked: 13.06.2014.

[Ma08] Matthes, F.; Buckl, S.; Leitel, J.; Schweda, C. M.: *Enterprise Architecture Management Tool Survey 2008*. Techn. Univ. München. München, Germany. 2008.

[Ma11] Matthes, D.: *Enterprise Architecture Frameworks Kompendium: Über 50 Rahmenwerke fur das IT-Management*. Xpert.press. Springer. Heidelberg, Germany. 2011.

[Mc14] McAffee, A.: *Enterprise 2.0, version 2.0*. `http://andrewmcafee.org/2006/05/enterprise_20_version_20/`. 2006, Last checked: 02.06.2014.

[O'07] O'Reilly, T.: *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. *Communications and Strategies*. (65):17–37. 2007.

[PP03] Poppendieck, M.; Poppendieck, T.: *Lean Software Development: An Agile Toolkit*. Addison-Wesley Longman Publishing Co., Inc. 2003.

[PP06] Poppendieck, M.; Poppendieck, T. D.: *Implementing lean software development: From concept to cash.* Addison-Wesley. Upper Saddle River, NJ, USA. 1 edition. 2006.

[Pr14] Pragmatic EA Ltd: *Tool Evaluation - Product: Model: Tool: Scores Version 2.2*. `http://www.pragmaticea.com/downloads.asp`. 2013, Last checked: 01.06.2014.

[RH13] Roth, S.; Hauder, M., Farwick, M., Matthes, F., Breu, R.: *Enterprise Architecture Documentation: Current Practices and Future Directions*. In *11th International Conference on Wirtschaftsinformatik (WI)*. 2013.

[RWR06] Ross, J. W.; Weill, P.; Robertson, D. C.: *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard Business School Press. Boston, MA, USA. 4 edition. 2006.

[RZM14] Roth, S.; Zec, M.; Matthes, F.: *Enterprise Architecture Visualization Tool Survey 2014: State-of-the-Art and Future Development*. `https://wwwmatthes.in.tum.de/pages/o790x7rho1te/EAVTS-2014-Final-Report`. 2014, Last checked: 25.05.2014.

[Sc09] Schöenherr, M.: *Towards a Common Terminology in the Discipline of Enterprise Architecture*. In (Feuerlicht, G.; Lamersdorf, W., Ed.): *Service-Oriented Computing – ICSOC 2008 Workshops*. volume 5472 of *Lecture Notes in Computer Science*. pages 400–413. Springer. Berlin and Heidelberg, Germany. 2009.

[Sc14] Schekkerman, J.: *Enterprise Architecture Tool Selection Guide Version 6.3*. `http://www.enterprise-architecture.info/Images/EA%20Tools/Enterprise%20Architecture%20Tool%20Selection%20Guide%20v6.3.pdf`. 2013, Last checked: 01.06.2014.

[Th14a] The Open Group: *TOGAF 9.1 Documentation*. `http://pubs.opengroup.org/architecture/togaf9-doc/arch/`. 2011, Last checked: 19.05.2014.

[Th14b] The Open Group: *ArchiMate 2.1 Specification*. `http://pubs.opengroup.org/architecture/archimate2-doc/`. 2013, Last checked: 20.05.2014.

[Th14c] The Open Group: *The Open Group - ArchiMate*. `http://www.opengroup.org/subjectareas/enterprise/archimate`. 2014, Last checked: 20.05.2014.

[Wh14] Whitehouse Office of Management and Budget: *A Common Approach to Federal Enterprise Architecture*. http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/common_approach_to_federal_ea.pdf. 2012, Last checked: 17.05.2014.

[Za14] Zachman International Inc.: *About the Zachman Framework*. http://www.zachman.com/about-the-zachman-framework. 2008, Last checked: 17.05.2014.