Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 192, Issue 35, June/July 2023, Pages 1–29
https://doi.org/10.7250/csimq.2023-35.01

Discovering and Assessing Enterprise Architecture Debts

Sara Daoudi[1], Malin Larsson[2], Simon Hacks[3]⋆, and Jürgen Jung[1]

[1]Frankfurt University of Applied Sciences, Frankfurt am Main, Germany
[2]KTH Royal Institute of Technology, Stockholm, Sweden
[3]Stockholm University, Stockholm, Sweden

sara.daoudi@outlook.de, malilars@kth.se, simon.hacks@dsv.su.se, jung.juergen@fb2.fra-uas.de

Abstract. The term Enterprise Architecture (EA) Debts has been coined to grasp the difference between the actual state of the EA and its hypothetical, optimal state. So far, different methods have been proposed to identify such EA Debts in organizations. However, these methods either are based on the transfer of known concepts from other domains to EA or are time and resource intensive. To overcome these shortcomings, we propose an approach that uses an interview format to identify EA Debts in enterprises and a method that allows a qualitative assessment of identified EA Debts. The proposed approach is supported by the designed framework that consists of an interview format and a process for determining thresholds of certain EA Smells.

Keywords: Enterprise Architecture, Quality Assessment, EA Debts Determination.

1 Introduction

One of the biggest challenges in digital transformation of enterprises is the alignment of business needs with information technology (IT) [1]. A holistic view is required to drive digital innovations as they significantly impact enterprise products, employees, or business models [2]. Having a misalignment usually results in inefficient business processes or poor business performance [3]. This is even more relevant when IT is becoming the driver of the business and its digital transformation strategy [4]. Digital trends, together with a change in organizational capabilities, build the foundation for developing new sources of value creation and improving speed and time of decision making [5]. Such objectives have traditionally been addressed by the Enterprise Architecture (EA) discipline [6]. EA provides methods and tools that aim to align business with IT, operationalize the business strategy, or even drive innovations inspired by a company's context [7].

EA aims at transparency by providing adequate visualizations of an organization or company. It encompasses business-related views (e.g., business processes, organization, business motivation) as well as abstractions for describing application landscapes together with required information

[⋆] Corresponding author

Reference: S. Daoudi, M. Larsson, S. Hacks, and J. Jung, "Discovering and Assessing Enterprise Architecture Debts", Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 35, pp. 1–29, 2023. Available: https://doi.org/10.7250/csimq.2023-35.01

Additional information. Authors ORCID iD: S. Hacks – https://orcid.org/0000-0003-0478-9347, J. Jung – https://orcid.org/0000-0002-8649-4827. PII S225599222300192X. Article received: 4 November 2022. Revised: 5 April 2023. Accepted: 1 May 2023. Available online: 31 July 2023.

technology [6], [8]. Enterprise Architecture Management (EAM) then uses such visualizations for planning and performing changes in the company. Such changes often affect software applications to be introduced, updated, or decommissioned, as IT departments usually drive EAM. Following a broader interpretation, EAM can also motivate changes in business processes, products, business models, and the entire enterprise [9]. In this case, EA is aligning IT with business needs and driving business innovations [7].

EAM has been introduced in organizations for many years, and the discipline has evolved over time, resulting in many frameworks, methods, and tools [10], [11]. Consequently, today's EAs reflect this evolution through many artifacts and systems being implemented in a corporate environment. Despite developing business-oriented IT, EAM is often perceived as bureaucratic, document-centric, and agility-hampering as it focuses on long-term effects [12], [13]. A study conducted by Kurnia et al. analyses the current state of EA in companies [14]. It identifies several blockers, like a negative perception by business stakeholders or a lack of communication skills. Furthermore, EAM tends to be IT-centric rather than involving business as a driver for IT-related decisions [15].

In such an environment, EAM consists of planning the organization's future application landscape and coordinating changes accordingly. Each change is conducted within an endeavor like IT projects, initiatives, or epics following business needs [16], [17]. There might be significant discrepancies between long-term EA objectives and individual projects. EA aims at finding an optimal overall application landscape while each project only focuses on the individual system. Furthermore, projects are driven by immediate business-related priorities, which can frequently change, while traditional EAM is too slow for adjusting plans [18], [19]. This may result in conflicts as the EA plans are not in line with business needs, and EA gets disconnected from relevant business needs. Typical consequences are, for instance:

- Complex application landscapes with legacy systems and redundancies,
- Out-dated or incomplete EA artifacts and documentation,
- Procedures and organizational units in EA management that hamper IT innovations.

These consequences are the results of decisions that have been made in the past. They might have been correct and justified at the corresponding point in time. However, the situation and context of an organization change, and these changes need to be incorporated into the design of the application landscape. But, existing legacy systems, historical procedures, and long-term planning processes slow the adoption of required changes – especially in agile markets and businesses.

Hacks et al. [20] shaped the term EA Debts in order to describe those results from past decisions that hamper changes in IT. Similar to the notion of technical debts, EA Debts represent blockers while moving from the current EA (as-is) towards a desired to-be-landscape. In contrast to technical debts, EA Debt encompasses technical systems, processes, organizational units, and regulations.

Based on this definition, a framework for managing EA Debts has been proposed [21] and will be further elaborated. One of the key questions in this framework consists of identifying EA Debts within a corporate environment. A catalog of typical EA Debts and EA smells (symptoms indicating the existence of a certain EA Debt) can help document common knowledge and typical EA Debts [22]. It serves as an analytical tool for identifying EA Debts that enterprise architects can use in an organization.

As a catalog can only document existing knowledge, a method for revealing new EA Debts is required. Such a method needs to involve enterprise architects together with relevant business representatives. An EA Debt workshop format has already been proposed and evaluated with several companies [23]. Such a workshop proved to be helpful in revealing EA Debts but is also very resource intensive as many people need to get involved (e.g., enterprise architects, managers from different business domains, EA Debt researchers, and facilitators). Therefore, we have developed an interview format to complement the workshops to identify IT-related issues and their alignment with the overarching business strategy presented in this work. An interview requires fewer people and can be done in a shorter time than a workshop. It is also more flexible as the interviewer can guide through the questions, while a workshop's dynamics are influenced by stakeholders' differing opinions and objectives.

The article is structured as follows. The research objectives and the corresponding research approach are presented in Section 2. Section 3 provides an overview on related research concerning the notion of enterprise architecture debts and related concepts. It will lead to a common understanding of the terminology used throughout the article. Section 4 then focuses on the first research objective – discovering EA Debts. An interview format has been used for discovering new debts. The section also lists results that have been validated. After providing an overview of debts discovered, a complementary method for assessing the threshold of EA Debts is presented in Section 5. The article closes with a summary and an outlook on future research in Section 6.

2 Research Objectives and Research Approach

2.1 Research Objectives

The impact of past EA decisions is imminent in various companies as legacy systems, infrastructure, and processes are often preventing the implementation of an ideal IT solution. A rigorous method for identifying such EA Debts is required as EA Debts, and especially their root causes, are not obvious to all corporate stakeholders. Such a method is presented in this article. The method is based on structured interviews for discovering and documenting EA Debts in an organization. The relevance/severity of each EA Debt will then be assessed with derived thresholds for the specific debt. The approach has already been evaluated with respect to applicability and relevance together with practitioners from different companies.

The following research objectives are addressed in this article:

RO1: Designing an interview format for discovering EA Debts in an organization.
RO2: Defining a method for assessing EA Debts in a corporate environment.
RO3: Building the foundation for a repository of EA Debts.

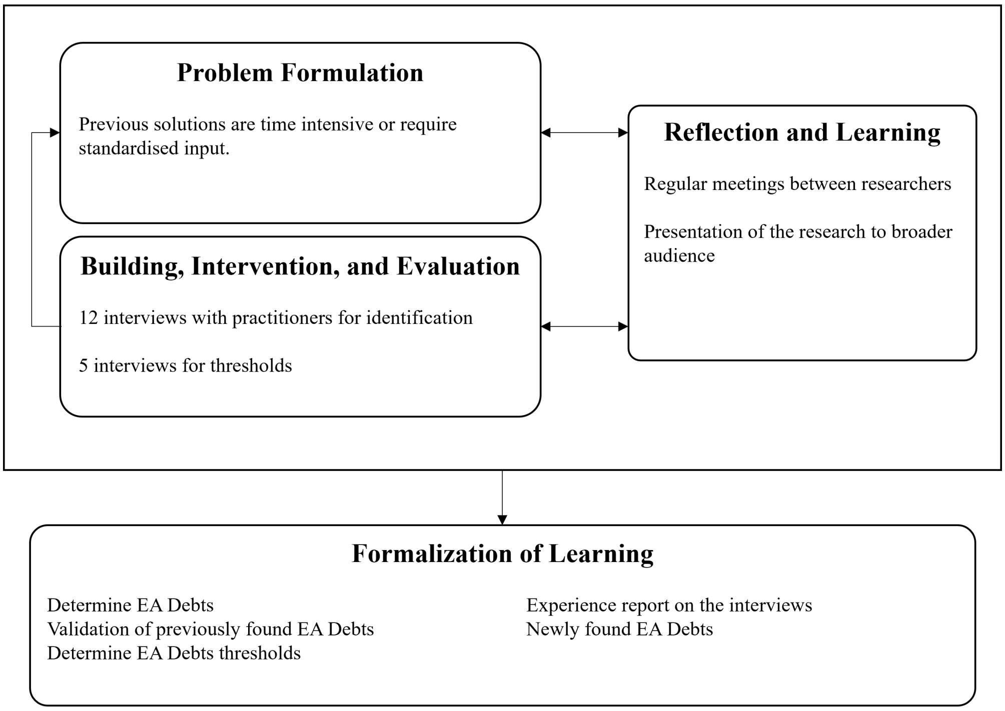
These research objectives are intended to substantiate an approach that can be applied by EA practitioners (e.g., enterprise architects, IT planners, consultants) in a company. These practitioners might not be aware of (all) EA Debts and require corresponding information from IT people and business experts. They perform structured interviews to get a list of potential EA Debts from the experts' experience. However, these candidates EA Debts are still subject to bias due to individual perceptions. The EA practitioner will then assess each item from the interviews' results together with corporate decision-makers. This allows for determining the relevance of each candidate and its impact on the overall organization. The EA practitioner then collects the EA Debts with their assessment in a central repository so that this information can be used to dismantle EA Debts in subsequent projects. The repository will eventually also serve as a knowledge base for future initiatives for identifying EA Debts in any organization.

The approach presented in this article does not cover EA Debt repayment (i.e., dismantling software systems or processes). EA Debt repayment is not completely excluded but will be subject to future research as a large body of EA Debts for analysis is required. Furthermore, the management of changes in an EA is already supported by EAM as well as related disciplines such as Application Portfolio Management (APM) [24] or Business Process Management (BPM) [25].

2.2 Research Approach

In this work, we design an artifact as a framework that supports enterprise architects to identify and assess EA Debts in their organization. This framework consists of an interview format and a process for determining thresholds of certain EA Smells. Accordingly, we rely on the principle of Design Science Research (DSR) [26]. To decide which concrete implementation of DSR we apply, we follow Venable et al. [27], who sketch a decision support that helps to choose the best fitting approach. They differentiate between objectivist, positivist methodologies and subjectivist, interpretive methodologies. They argue that if one expects the designed artifact to be the best solution for a generalized target group that behaves the same, one should opt for one of the objectivist, positivist methodologies, while a subjectivist, interpretive methodology is tailored to the needs of a certain problem.

In our case, we develop an artifact that is tailored to the needs of a certain problem, i.e., the identification of EA Debts in organizations. Thus, we opt for a subjectivist, interpretive methodology. Those methodologies are distinguished by the domains they address. Venable et al. [27] state: "If you have a single client that wants to engage in a research undertaking with you then choose ADR". As we are solely focusing on the domain of EA at the moment, we opt for Action Design Research (ADR) [28], while understanding the body of EA experts involved in our interviews as the organization we interact with. ADR comprises four stages, which are discussed subsequently and illustrated in Figure 1.



Problem Formulation

Previous solutions are time intensive or require standardised input.

Building, Intervention, and Evaluation

12 interviews with practitioners for identification

5 interviews for thresholds

Reflection and Learning

Regular meetings between researchers

Presentation of the research to broader audience

Formalization of Learning

Determine EA Debts
Validation of previously found EA Debts
Determine EA Debts thresholds

Experience report on the interviews
Newly found EA Debts

Figure 1. Action Design Research with Main Contributions in this Work

Stage 1: Problem Formulation. We face the challenge of finding an approach to identifying EA Debts in organizations. Hitherto, previous approaches are either extremely time intensive [23] or require standardized inputs in the form of EA models [22], [29], which also are able to cover just a subset of possible EA Debts. Additionally, the latter is solely capable of identifying known EA Debts and provides no means to identify unknown EA Debts.

To sum up, we face the following challenges:

- Unknown types of EA Debts need to be identified in organizations.
- The time of practitioners is usually strongly constrained.
- Not all kinds of EA Debts can be identified in EA Models.
- Knowledge is spread out in organizations among many people.
- Practitioners can perceive EA Debts differently with respect to their role.

From these challenges, we deduct the following requirements:

1. The solution should be capable of identifying EA Debts that are not documented yet.
2. Compared to the previous research [23], the approach should require less time spent to grasp a larger set of input.
3. EA Debts should be assessed with respect to their relevance.

These requirements comply with the research objectives listed in Section 2.1.

Stage 2: Building, Intervention, and Evaluation. In stage 2, we follow an organization-dominant approach. We conducted 12 interviews with practitioners from the field. Firstly, we explained the terminology of EA Debts to them. Then, we asked about EA Debts that they perceive in their daily working life. Based on the practitioners' feedback, we developed our approach further to reflect the new knowledge and improve the artifact.

To identify EA Debts, we opted for a qualitative research approach, i.e., a combination of problem-centered and expert interviews. To address the problem-centered facet, we conducted a literature research [30, p. 230], and we designed a guideline based on the existing theoretical considerations [31, p. 364]. The guidelines ensure that the research questions are answered while leaving room for ad hoc questions [30, pp. 236–237] to clarify unclear aspects, following the 5-Why-Methodology [32, p. 8]. In addition, elements from pain point analysis have been incorporated (cf. [33, pp. 115]). On the other hand, the expert facet focuses on the expert with specialized knowledge [34, p. 9], i.e., process knowledge and interpretive knowledge. Process knowledge relates to situations in which the expert was involved, e.g., in a project. While interpretive knowledge is intended to convey the expert's assessment of the topic of EA Debts [34, p. 18–19].

Experts participating in interviews should have to have access to relevant information, be able to accurately reproduce it, and be willing to disclose this information [35, p. 117]. Therefore, we interviewed people involved in enterprise architecture, such as, IT architects, CIOs, or product owners. However, we assume that the experts will not be perfect in identifying EA Smells and EA Debts as the research on EA Debts is still in its infancy.

Similarly, we proceeded to design the process to determine the thresholds. We took inspiration from Saravia et al. [36] and refined the process in interaction with five participants that were distinct from the set of experts who facilitated the determination of EA Debts.

Stage 3: Reflection and Learning. While in stage 2, a specific solution for a certain problem is developed, stage 3 focuses on the reflection and learning process. Therefore, we conducted several meetings with the researchers to discuss the latest findings and determine their outcomes for our approach. Additionally, we organized several presentations in which we communicated our results to a broader audience and gathered feedback from outside our setting.

Stage 4: Formalization of Learning. The outcomes of stage 2 denote the following contributions of this work:

- An approach to determine EA Debts in different organizations.
- An experience report on conducting interviews to determine EA Debts.
- Support for EA Debts that were identified previously in other organizations.
- A set of EA Debts not reported yet.
- A process to determine EA Debt thresholds.

3 Related Research

3.1 Enterprise Architecture

IT is becoming more and more pervasive and increasingly important in organizations [37], [38]. At the same time, the integration of business requirements and IT functionality is also becoming increasingly important despite the business-IT alignment has been important for CIOs for a long time [39], [40]. To promote this business-IT alignment, more and more information systems (IS) change and development projects address the realization of technical solutions according to local business needs. EA is a widely accepted discipline [41], [42] to steer the local IS endeavors by means of fundamental structures, design, and evolution principles of the overall organization [43]. Therefore, EA focuses on aligning IS projects with enterprise-wide objectives of reducing complexity and integration efforts in the overall IS landscape [44], [45].

3.2 EA Debts

The increasing digitalization of organizations raises the application of agile methods. This creates new challenges for EA, as the EA conceptions phases to define proper target architectures become reduced [46]. One driver for this observation is the tendency of product owners to prefer short-term business value over solid architectural solutions, while effective means to propagate long-term architectural solutions are missing [47], [48].

To provide a means that support enterprise architects in arguing for more sustainable solutions, Hacks et al. [20] extend the concept of Technical Debts, which describes past technical shortcuts that hamper IT developments [49], [50], to the EA domain by suggesting a more holistic view on the entire organization.

The metaphor Technical Debt was first introduced by Cunningham [49] and refers to a concept that we understand today as “refactoring”. Thereby, Technical Debt refers to invisible properties of the software, i.e., qualitative aspects of the software that do not affect the features of the software but its maintainability [51]. Consequently, the concept of Technical Debt retrospectively reflects on the change in the environment, rapid success, or technological advancements as a possible cause for debt. However, a debt can be, depending on the situation, a good investment as long as the development team is aware of its imperative and the increased friction [51]. To support this awareness, tools have been developed that identify debt and its causes and enable the management of debt-related tasks [51].

Cunningham’s idea of not-quite-right code, which we postpone making it right, was extended [51], [52] to other kinds of debts or ills in software development, such as test and requirement debt. Moreover, different efforts have been invested to categorize these different technical debts. For example, McConnell [53] outlines business and technical aspects that can be highlighted, enhancing the communication of specific problems within a software. In other words, Technical Debt is a uniform communication tool to measure and keep track of debt while reflecting different stakeholders’ viewpoints to ease an effective collaboration [53], [54].

So far, Technical Debt has been able to demonstrate its benefits to estimating deficits in software, being a tool for decision making, and increasing the awareness [51], [52], [53], [55]. However, its focus was always in the technical aspects of a single system, thus missing the opportunity to be helpful for the entire enterprise and every scenario [56], [57].

This is where the concept of EA Dept comes into play, which Hacks et al. [20] define as ”the deviation of the currently present state of an enterprise from a hypothetical ideal state.” Such a deviation can be interpreted from two different perspectives: On the one hand, it can result from decisions that are expedient in the short term but cause future changes to be more costly. Therefore, EA Debts might hamper the implementation of better solutions in future. In other words, it is related to the planning of the EA toward its future evolutions (ex-post). On the other hand, the debt can describe a deviation in the actual EA that might have arisen due to changes in the valuation of the EA. In other words, when the decision was made, it was in line with the optimal EA, but over time, the strategic focus of the organization changed, leading to another optimal EA, and, thus, the former decision is now causing a debt (ex-ante).

However, it is in both cases a challenge to identify EA Debts in organizations, especially as it can be unclear what a ”hypothetical ideal state” could be. Our understanding of this ideal state refers to an EA that aligns exactly with the organization’s overall goals. This covers all parts of the EA, from business processes that work as they should to technical infrastructure that is in place according to the organization's strategy. To provide an easy way to assess the deviation between the actual and ideal states, Salentin and Hacks [22] suggest the concept of so-called EA smells that serve as potential indicators or symptoms within the EA. Moreover, they provide a first catalog of EA smells that help to identify possible flaws in EA models.

The new field of EA Debt-related research poses mainly two related streams of research [21]: On the one hand, it is research related to the technical aspects of EA Debt. On the other hand, it is research that elaborates on the socio-technical aspects of EA Debt.

Most of the research has been published on the technical aspects of EA Debts. As such, Salentin and Hacks [22] have not just published the first definition and set of EA Smells but also a prototype that was able to identify some of the smells in ArchiMate models. Where Salentin and Hacks [22] derived the smells from known code smells, Lehmann et al. [58] and Tieu and Hacks [59] continued in this line of research by relying on known anti-patterns from the BPM community and from Software Architecture Smells, respectively. To ease the identification of these smells, Smajevic et al. [29] developed related tool support that is not only able to identify EA Smells in an automatized way in EA models but also expands the identification from ArchiMate models to any EA model that has a graph-based representation.

Given an identified set of EA Debts, the question arises, which should be solved next. To address this question, Yeong et al. [60] propose to adapt portfolio theory and utility functions to prioritize the different debts in an organization's optimal manner. Next, the set of refactoring presented by Liss et al. [61] can be used to guide the removal of the respective debts.

To provide a frame for the beforehand presented technical measures, Alexander et al. [21] propose a process in which EA Debts are first identified and collected, then assessed and prioritized, and finally either removed or actively monitored. Here, the proposed workshop format of Jung et al. [23] fits in by providing a means to also identify EA Debts and EA Smells that cannot be detected by solely relying on EA models. However, the proposed workshop format is time-intensive and, hence, just limited and applicable on larger scales. This shortcoming is addressed in this article (cf. Section 4). Moreover, none of the known approaches answers the question of when an EA Smell is to be considered to have low quality and, thus, cause an issue for an organization. This question is addressed in Section 5 of this article.

4 Discovering EA Debts

This section provides an overview of the method for identifying EA Debts in organizations. An open interview approach was defined and used with 12 interviewees. Its structure is explained in section 4.1, and its evaluation is shown in subsequent sections.

4.1 Interview Structure

The structure of the interview (cf. Table 1) has been developed based on the method of problem-centered interviews and the root cause analysis. Only individual elements from the Root Cause Analysis are considered. Focus has been set on those that are important achieving research objectives. The ad hoc questions approach from the problem-centered interview is used for a detailed analysis of the problem. At the beginning of the interview, the topic is determined [31, p. 365] by introducing the term EA Debts and informal definitions of the terms EA Smells (Symptoms) and EA Debts. These terms are derived from Technical Debt and Code Smell and provide a first example of the idea behind EA Debts.

In the course of the interview, the experts thus have the opportunity to reflect on the terms, and they can provide their own understanding of the terms or agree to the ones provided by the interviewer. The informal terms are meant to provide first ideas, and the expert is encouraged to find their own definition. Furthermore, a real-world example of EA Debts will be introduced to the

Table 1. Final interview structure after pretest

Phase	Focus of primary questions
1	Collecting first impressions about the topic **Brainstorming**
2	Perception of EA Debts **Direct question**
3	Awareness of EA Debts **Direct question**
4	Understanding of problems and identify possible causes **Problem analysis, Pain Point-Analysis, 5-Why**
5	Identifying Causes **Symptom or Cause**
6	**Evaluation** of the expert interview (from the experts' perspective)

experts. The whole interview consists of six phases, each of which includes primary, secondary, and follow-up questions. The first question is, *"What comes to your mind spontaneously when you think of the terms EA Smells and EA Debts?"*

The first phase is a simplified brainstorming approach, which is also part of the Root Cause Analysis. A brainstorming approach helps identify potential causes in advance, which can be used for further analysis. Brainstorming also prevents the analysis from getting distracted from its objective [62, p. 44]. Potential candidates for EA Smells and EA Debts should already emerge in this context. Brainstorming is a good way to start with the topic since the expert can first think about it and get familiar with it. If the interviewer does not get enough information from these questions, then follow-up questions may support heading toward a more precise answer. Those questions are about awareness and perception of EA Debts. According to Hacks et al. [20, p. 13], awareness of difficult or invisible elements needs to be increased as they impose a higher risk for EA Debts.

In the second and third phases, the problems of the expert's organization should be determined. The method of pain point analysis is used to classify problems as pain points (cf. [33, pp. 115]). The Pain Point analysis suggests asking about the larger challenges to identify relevant problems. Direct questions about challenges can be seen as a start to further conversation, and the interviewee is enticed to talk. The question is also asked directly in general terms about the organization's challenges so that it is more concrete. It implies that a detailed definition of the problem should be made [32, p. 23]: *"A good problem definition can help make the diagnosis more focused and productive"*. Further investigation should be done according to the relevance of the problems. Various tools are also described for this purpose [32, pp. 23–24]. However, discussing them in detail would go beyond the scope of this article. In further investigation, several problems should be raised, and in addition, secondary questions in the form of ad hoc questions are to be used for a more in-depth analysis of the problem. Secondary questions will specifically ask for a more detailed explanation of the problem if the interviewer feels that the problem has only been briefly touched. Further examples of secondary questions from the literature were integrated into the guide [62, p. 55], which refer to asking about the problem in more detail if the expert does not provide sufficient information on his own.

In the fourth phase, the 5-Why-Method is used to get closer to the root cause(s). With this approach, symptoms, physical and system causes can be investigated [32, p. 16]. This method is based on the assumption that people tend not to think of the real root cause immediately. They might provide reasons, but those reasons are usually a result of the root cause behind them. An example is provided by a hypothetical chain of raising the question "Why?" again and again as in Figure 2. Given the problem that a business process is not working properly, the first valid reason might be a bad choice for the software system. This is, obviously, a reason, and we could fix the problem by changing the system. However, we should also understand why there has been a bad choice, which is why we ask "Why?" again. Doing this repeatedly leads to the real root cause, a missing responsibility for creating a business process for communicating IT knowledge properly.

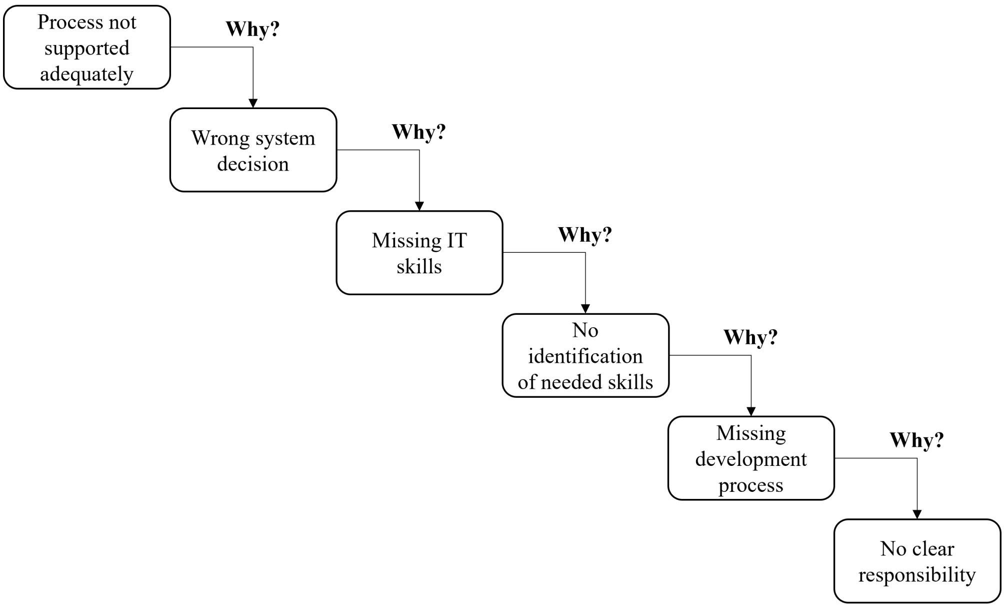

Process not supported adequately
Why?
Wrong system decision
Why?
Missing IT skills
Why?
No identification of needed skills
Why?
Missing development process
Why?
No clear responsibility

Figure 2. Five Why Example

The 5-Why-method is only used to find causes and symptoms throughout the interviewing process. Therefore, there is no differentiation between physical and system causes. The following hypothetical deliberation was made on how the 5-Why method can work concretely. For this purpose, the following example is described: A payment system has been introduced, but it is not the proper system because it cannot technically process all the technically required payments. Figure 2 shows an example of how the 5-Why can proceed. The answers are hypothetical examples and can also be done differently. It is a deliberation of how to proceed. In addition, as many why questions as possible are asked so that as many causes as possible can be identified. Questions can also be asked here if something is incomprehensible.

In the fifth phase of the interview structure, all the points mentioned by the expert that could be potential candidates for an EA Debt or EA Smell are interactively noted with the expert in a table. The interviewer repeats the points mentioned in the first and fourth phase questions. In addition, points from the second and third phase questions can be considered as well since the expert might also have mentioned concrete examples of smells and debts for the second and third main questions. Then the expert should be asked whether it is an EA Smell or EA Debt. The expert should make a classification based on his practical experience. In the Root Cause Analysis, the third step is to find out which factors are less likely to be the cause of the problem and which factors are more likely to be the cause of the problem [32, pp. 49–51]. In this phase of questions, the output, according to the experts' assessments, should result in a list containing potential causes [32, pp. 179–181]. The list also includes the potential symptoms of EA Smells. The results are presented in the form of keywords in a list; thus the result is qualitative text data [32, p. 69].

Finally, in the sixth phase, the expert evaluates the interview. To contribute to EA Debts research, the expert should assess what effect the questions had on them and whether they were complete and purposeful in relation to the topic of EA Debts. In addition, the expert should be given the opportunity to mention further points in relation to the topic of EA Debts. Based on an interview example, individual concluding questions can be designed [63, p. 410].

4.2 Pretest

A pretest has been conducted to verify the interview guide, as the object of investigation is very complex [64, p. 46]. It is recommended to choose an expert who would also have belonged to the interviewed group [65, pp. 69–70]. The pretest was conducted with a researcher (Expert 1) in the topics of EA Debts and EA Smells. Expert 1 also works for an energy supplier as an enterprise architect and thus brings a certain experience from practice. He was able to provide information from both (short- and long-time) perspectives for the final design of the interview guide. The interview was conducted accordingly, with one exception that the third phase was excluded from the interview, as the interview had already exceeded the planned time frame of approximately 60 minutes, and it was not clear whether the question of the third phase had possibly already been answered by the expert in the second phase. All questions were read out verbally to the expert. Follow-up questions were asked if necessary. Occasionally, 5-Why questions could be asked. These were asked directly in the fourth phase after the expert had named and explained the problems. In the final evaluation of the interview, the expert recommended some improvements. In the context of this article, only one pretest was conducted to primarily check the understanding of the questions.

It can be assumed that a single pretest is not very meaningful to optimally test the interview guide. Thus, this point must be considered critically within our research. Overall, the pretest showed that no major changes were necessary. The rough structure was kept as large amount of information on the questions could be collected, and the questions were designed open enough to give the expert the opportunity to report in detail on the topic. After considering the results, the question of the third phase of the interview was finally included so that further answers from the expert could be considered. The question of the fifth phase was integrated into the question of the fourth phase, as it is easier to ask "Why?" directly after analyzing the individual problems. This arose spontaneously during the interview, and it was possible to react directly to the problems with the "Why?" questions.

Finally, questions have been grouped into six phases. The questions were basically easy to understand. Only one question (*"What are the current problems or challenges in your organization or company?"*) should focus on EA. Expert 1 pointed out that this question should be adapted accordingly to directly exclude far-reaching topics that do not belong to the EA. The closed primary question from the second phase could not be answered directly by the expert with ”Yes” or ”No”; he answered the question with "Partly". "Partly" was thus included as an answer option, and the question was kept in the questionnaire to reflect the expert's initial opinion and to test the effect of a closed question. If the expert is undecided, they can also select "Partly" as an answer option.

One challenge identified during the pretest is following the narration and checking whether all questions (including follow-up questions) have been answered. The time frame for the interview was also refined. The interviews should last about 90 minutes and can also be extended if necessary and possible. Experience in the related work has shown that the best results come from interviews that take place in a time frame of 90 to 120 minutes, as it is only then possible to analyze the problem in depth. Thus, the guide should also provide enough material to be able to use the time meaningfully [65, pp. 52–53]. The pretest also showed that a detailed conversation could provide many symptoms and causes.

4.3 Conducting the Interviews

After the pretest, the interviews had to be scheduled. Various enterprise architects were contacted via email for an interview. In the end, eleven experts were interviewed after the pretest (cf. Table 2); almost all of them were architects in one form or another. Only expert twelve was an IT consultant

Table 2. Interviewees' demographic data

Expert	Duration	Business Sector	Business Role
1 (Pretest)	1:45 h	Energy	Enterprise Architect
2	2:30 h	Finance	Program Manager
3	2 h	Energy	Enterprise Architect
4	2 h	Finance	Information Architect
5	1:15 h	Logistics	Enterprise Architect
6	1:45 h	Finance	Enterprise Architect
7	2 h	Finance	Enterprise Architect
8 (2 Interviews)	1:30 h 0:45 h	Finance	Enterprise Architect
9	1:45 h	Transportation	Enterprise Architect
10	1:15 h	Finance	Enterprise Architect
11	1:30 h	Retail	Enterprise Architect
12	1:30 h	Consulting	Consultant

for EA topics. However, all experts had many years of experience in the field of EA. Six of the experts interviewed were from the financial sector, while the others were from the postal and parcel delivery sector, retail, a transport company, an energy supplier, and a management consultancy.

4.4 Qualitative Text Analysis

The interview results have been analyzed using Lamnek's approach for qualitative text analysis. Systematic interpretative procedures can be used to analyze the interview material for qualitative expert interviews [65, p. 3]. This was conducted based on transcripts from audio recordings. An exact or unambiguous interpretation of the interview is impossible since diverse interpretations can be derived from what is said. An analysis of language use, repetitions, peculiarities, and new insights must be considered for the analysis [66, pp. 25–26].

According to Lamnek, various possibilities exist for analyzing the material of qualitative interviews. However, he describes a general procedure for analyzing the interviews, which can be deducted in four phases. This procedure is modifiable and can be adapted accordingly. The following four phases are described [31, p. 402]: (1) transcription, (2) individual analysis, (3) generalizing analysis, and (4) control phase. Transcription of the entire audio recording does not usually occur and is an exception [67, p. 83].

Most of the content was transcribed as verbatim as possible and individual keywords such as personal or company names were anonymized accordingly. The transcription was also smoothed for a better reading flow, and the questions regarding the evaluation were mostly paraphrased. The transcripts were generally kept verbatim and were shown in italics. For better readability, the interviewer's content was not presented in italics. The individual analysis described by Lamnek is deducted through the inductive category formation of qualitative content analysis. For this purpose, each individual interview is examined, and categories are built sequentially [68, p. 63].

In summary, central passages from the interview material are highlighted in the form of inductive categories. These are then subjected to a content analysis. The result is a compressed material. The compressed material can be used to characterize the interview [69, pp. 403–404]. This procedure was partly modified. No characteristics were defined for the individual interviews. However, the material was analyzed individually with regard to the characterization of the categories. In the following, a generalizing analysis of the interview material will take place. Lamnek describes the generalizing analysis as the next general step to reach general conclusions or findings beyond the individual interviews [31, p. 404]. According to Kuckartz, this is done by creating two topic matrices. Here, the categories represent one dimension, and the other dimension is represented by the interviews of the experts [68, p. 74].

The qualitative text analysis of the interview material resulted in 15 categories. These categories were subdivided according to the terms Symptom or Debt. The categories are described in order to provide an idea of the research results.

Integration. Problems are generally described with regard to the integration of systems and processes that were not carried out successfully. In addition, EA documentation is not integrated into the business process flow. In general, the integration should be done properly and the right tools should be used for the integration. Expert 2 describes this as a *”failed integration policy”*. Also, Expert 10 refers to this: *“...[They] do not have enough budget or just or so they did not look deep enough and where they then also neglected integration, which then just leads to big costs afterwards...”*

Software. This category describes problems that arise through the use and handling of systems, for instance, through the use of redundant systems and functions. All of the experts from the interviews reported problems relating to this category. A high number of problems were described in this category compared to the other categories. In general, legacy systems were frequently reported to cause difficulties. Dependence on the vendor of the technological product is another problem *“...but for the company, that just means more licensing costs and another vendor lock-in with Oracle, who can then also turn the cost screw...”* (Expert 6). Expert 7 points to the *“reuse”* possibility of applications, which has not been specifically considered. Expert 10 reports that many applications are hardly used in the company. Many systems (e.g., applications, data warehouses) also increase the application landscape’s complexity.

Governance. This category describes problems in the corporate processes that set standards and guidelines. Poor or non-optimal process management is described here. For instance, no processes were set up in the past with regard to data management. There is a fundamental lack of data governance for this. In general, there is a lack of concrete instructions and standards and the processes are insufficiently defined, outdated, and partly overgeneralized. The processes lead to additional work, *“... that’s why we need stronger IT support and the other thing, [...] is that the processes if they are only manual, there are also expensive ...”* (Expert 11). Another example of no guidelines is described as follows: *“...in the end, everyone can do what they want, there are no binding rules anymore and one person just does Java, the other does C#, the next maybe does C++, ...”* (Expert 12). In addition, there is a lack of harmonious collaboration between IT and business departments, which is not present among business departments. The architecture is not included in decisions at an early stage. The processes are also lacking here.

People. This category refers to the behavior, attitudes, and expertise of all company employees. Problems arise here because the people do not have certain know-how, e.g., about technological topics. The experts also observed a certain dismissive behavior about new or innovative technologies, in that the specialist department does not get involved with these technologies. It was also described that project managers do not work transparently. Another problem is that communication, teamwork, and a common understanding are not present: *“... an understanding in management that if I am serious about business architecture, that I actually then also, accordingly, that I make at least half a staff capacity available ...”* (Expert 7). Another example is: *“...the departments say everything worked out until now, why should I do more now ?...”* (Expert 8).

Responsibility. The category describes problems that arise when responsibilities are not clearly defined. In general, there is a lack of concrete instructions, e.g., from management, that responsibility is also taken for concrete problems. Responsibilities are not taken seriously, and it is assumed that someone will take responsibility for it on their own: *"... that there is actually no responsibility for certain aspects, that this is the kind of thing that just, well, you think like, well, someone will take care of it, but it doesn't take place ..."* (Expert 4).

Information. This category describes the handling of information in the company. Technical information and its meaning are described. The information or information flows cannot be assigned, are incomprehensible, or concerns wrong information. The demarcation to the category data is that it is not about the attitude of the information itself, but about the meaning of this, *"... how do I interpret this information at all, what does it mean, if in the field 24 there is suddenly a 3..."* (Expert 2). Also, Expert 4 illustrates an example: *"...then a field is called in our Warehouse, then any 3...4 letters lined up and if someone has built that somehow 10 years ago and has baptized it, even then this person knows what these 3...4 letters mean, but if one from the specialist area comes, which sees then all at once these 3 letters or 4 for him the combination does not make sense at all..."*

Transparency. This category is not described superficially by the experts. It describes fundamental aspects of the enterprise architecture where transparency is lacking. For instance, a lack of transparency is described in connection with data, information, decisions, the status of a project, costs, and the application landscape: *"... fewer people [...] have an overview of what capabilities are available at all, and the more they simply say, yes, we need something new, so we need something new"* (Expert 10).

Documentation. This category is mentioned by a few experts. Missing and insufficient documentation is described here. It refers to all areas in which, for instance, a data flow is not documented, and this would lead to an additional effort: *"... to maintain or further expand this data flow or something likely, but if no documentation has taken place at an earlier point in time, then it is incredibly difficult to make the first changes to such an information flow..."* (Expert 4). Another example is: *"... I hardly know any company that has documented anything properly, unless it must do so, [..] like in the military or in the banking sector. In the whole core area, maybe it is done, but as you have seen with Wirecard, even in this kind of environment it is not the case ..."* (Expert 3).

Requirements. This category refers to requirements that were not clearly defined at the beginning. Basically, trade-offs are described where risks are taken for implementation. Compromises are made and alternative or intermediate solutions are implemented *"... then in the end, of course, it's often a missing front-end thing, [...], projects just start running always with the same time-to-market argument, [...], they say we have to start, otherwise we won't be ready in time..."* (Expert 5). Another example describes it also clearly: *"...sometimes a conscious decision also includes a certain amount of debt, because it is simply a throw-away solution for a certain period of time or something similar, or because the second point, which is often also known, is because time is pressing and you first test something on the market and then it makes it better or right, ..."* (Expert 9).

Architecture. This category refers to problems in the whole system landscape. The infrastructure is unsuitable, interfaces are not maintained properly, or the necessary interfaces are not built. The application landscape is too complex and too large: *"..., if I now have 100 workflows here in my* *folders, then that is also a code smell and architecture smell for me, so to speak, the thing is now simply too large [...] that we say, okay, we are now splitting this up ...”* (Expert 6).

Time. This category encompasses EA Debts related to temporal aspects and scheduling. These can occur in both directions. For example, certain processes can run faster or slower. In concrete terms, aspects are described where something occurs with a delay. In addition, the timing of e.g., components or system products, are described: *“... your Oracle database is going out of support, when do you want to update it”* (Expert 8).

4.5 Characterizing EA Debts

The fifth phase of the interview aims to classify the results as either EA Debt or a symptom for detecting such a debt. Experts could exclude each aspect if they were unsure whether to classify it as debt or symptom. In fact, it was repeatedly noted that it was not always easy for the experts to categorize the results as symptoms and debt. Both terms should be refined with respect to an unambiguous interpretation. An interpretative approach will be taken to make an approach to sharpen the term EA Debts since the results in themselves do not achieve a clear sharpening of the term EA Debts. However, as a preliminary step, all aspects of EA Debts are first classified as Debt. For the symptoms, it is checked which aspects have been consistently defined as symptoms by the experts. Accordingly, these are classified as symptoms. Some exceptions of the interview material were removed. These exceptions represent individual cases and are therefore not considered in more detail. The classification of a symptom or Debt was not always clear. The interviews show that some experts classified an aspect as a symptom when they were unsure. The following exemplary excerpts from the interviews will demonstrate this:

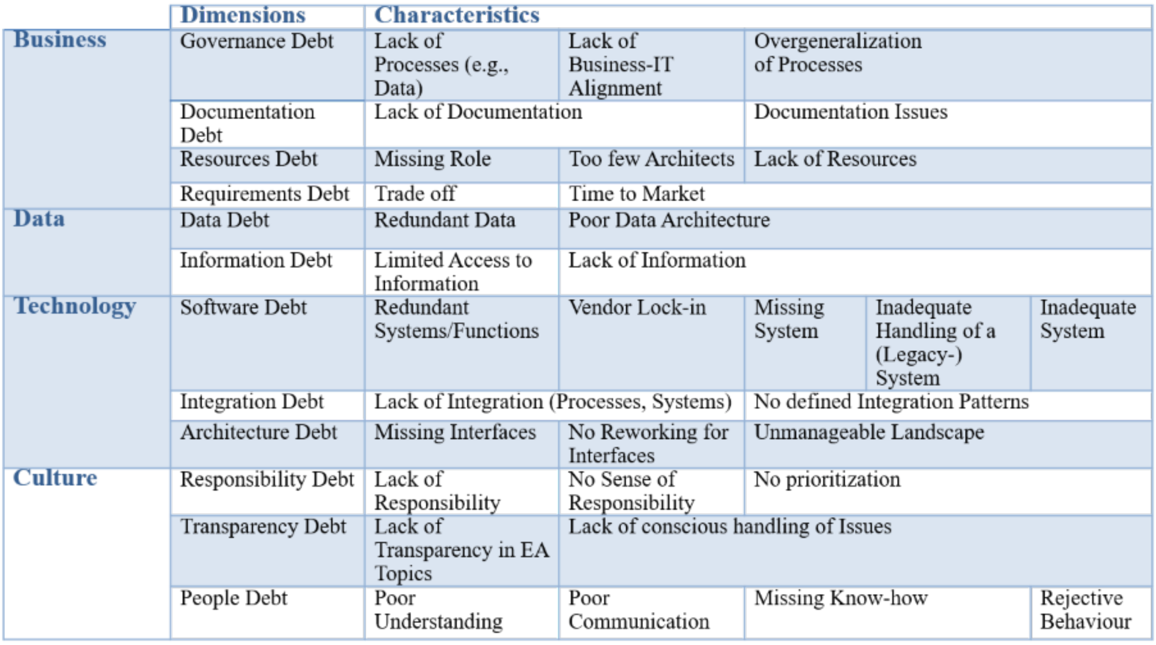
> Expert 6: *“... but those would be for me now two examples of Smells, which are also quite clearly debts ...”*
> Expert 7: *“Now that’s a good question, is that an indicator, it may even be both in that case, an indicator I think in any case ...”*

4.6 Taxonomie of EA Debts

The main contribution of this work is the developed interview format to ease the identification of EA Debts and EA Smells in organizations. However, during conducting the interviews to evaluate the format, we also identified some EA Debts and EA Smells that we would like to report. To represent the results, we use the taxonomy-building method of Nickerson et al. [70] for the graphical representation of EA Debts and EA Smells. The methodological approach was not used because a qualitative text analysis had already been conducted. The methodological approach requires that the meta-characteristics are defined first. Then end conditions are determined, which, for instance, differentiate the dimensions and characteristics uniformly from each other. Subsequently, the elements and characteristics of the objects are determined, which are repeatedly checked in an iterative process about the fulfillment of the end conditions. A distinction is made between a deductive (conceptual-to-empirical) and an inductive (empirical-to-conceptual) approach.

In the example of Nickerson et al., the objects to be considered would thus be the EA Smells and EA Debts identified by the experts from the empirical material. Based on the results of the fifth phase of the interview, a sharpening of both terms was interpreted. The categories were assigned accordingly.

For EA Debts, additional meta-characteristics were added afterward. These result from the levels of EA, where EA Debt was classified by Hacks et al. EA Debts should therefore be found on all levels [20], [71]. The Technology and Application levels were combined and defined as a Technology meta-characteristic since a Software Debt category had already been formed. The cultural aspect, which also plays a role in EA [6, p. 7], was added as a further meta-characteristic. This was also mentioned by Expert 3 and Expert 7 in the context of EA Smells and EA Debts. According to Nickerson et al. [70] the meta-characteristics must be defined in advance. The taxonomy only serves as a way of representation. The illustration is complete without the meta-characteristics and results from the material (cf. Figure 3). Only the differentiation of the terms EA Smells and EA Debts was conducted by an interpretative approach.



	Dimensions	Characteristics				
Business	Governance Debt	Lack of Processes (e.g., Data)	Lack of Business-IT Alignment	Overgeneralization of Processes		
	Documentation Debt	Lack of Documentation		Documentation Issues		
	Resources Debt	Missing Role	Too few Architects	Lack of Resources		
	Requirements Debt	Trade off	Time to Market			
Data	Data Debt	Redundant Data	Poor Data Architecture			
	Information Debt	Limited Access to Information	Lack of Information			
Technology	Software Debt	Redundant Systems/Functions	Vendor Lock-in	Missing System	Inadequate Handling of a (Legacy-) System	Inadequate System
	Integration Debt	Lack of Integration (Processes, Systems)		No defined Integration Patterns		
	Architecture Debt	Missing Interfaces	No Reworking for Interfaces	Unmanageable Landscape		
Culture	Responsibility Debt	Lack of Responsibility	No Sense of Responsibility	No prioritization		
	Transparency Debt	Lack of Transparency in EA Topics	Lack of conscious handling of Issues			
	People Debt	Poor Understanding	Poor Communication	Missing Know-how		Rejective Behaviour

Figure 3. EA Debts' Taxonomie

4.7 Reflection on the Interviews

The results of the interviews show an approach to characterize the term EA Debts. The methodological approach of Root Cause Analysis, especially the 5-Why method, should have identified more causes than symptoms. However, the results clearly show that many more symptoms than causes were identified by the experts. The interpretive approach that there was some uncertainty among some experts regarding the definition and classification of the terms could be a reason for this. In total, 15 categories could be formed that include potential EA Debts. The categories of Governance and Software could be substantiated with examples by each expert. These are the strongest categories of the material. This could be due to the importance of business and IT in the context of EA. Accordingly, the essence of a business and IT is captured by EA [6, p. 3]. These two categories seem to be very important according to the experts and show that there are many debts to find. Many of the described problems of the categories for EA Smells and EA Debts are also found in the challenges of EA. A first example is that companies are not informed about their products, services, etc. [72, p. 7].

Expert 7 illustrated this very clearly with the *"Reuse"* example: *"... the topic of reuse, where I already have powerful platforms in-house [...], that people simply say, yes, no, I don't know any tool that we have in-house, and I will have a look on the market later, there are some, but I particularly like the one, by the way, I have already invited the software manufacturer ..."*. This example clearly shows that there is no overview of the existing platforms. Another example is an unclear responsibility in the company [72, p. 7]: *"In my concrete environment, we have that quite a lot, that you, that there is actually no responsibility for certain aspects, that such things just, well, you think so, well, someone will take care of it ..."* (Expert 4).

In addition, further challenges are the lack of understanding and the lack of common control of the most important data sources [72, p. 7]. This becomes clear by the following example of Expert 8: *"... topic data quality, that is yes, a big complex [...], which damage can develop there, and that it is meaningful to invest purely there because they say, the specialist areas, yep, had so far everything worked out, why should I now make there still more ..."*. Here it becomes clear that the department does not want to take care of the topic of data quality because it has not yet recognized the dangers. Thus, a certain understanding of the topic is missing here.

A final example is redundancies in terms of resources and technologies [72, p. 7]. Expert 9 clarifies these challenges about technologies as follows: *"... that we have different departments that acquire software solutions that functionally overlap each other in part, and where then such things as synergies are not used at all, so the IT landscape becomes a real zoo ..."* Here he describes software solutions that functionally overlap, and thus there are redundancies in the systems so that synergies cannot be used in the company and the IT landscape becomes too chaotic. This should be an exemplary presentation; of course, there are further examples from the interviews for the challenges already mentioned or also for further challenges from the literature. In this article, not all challenges are specifically addressed. EA challenges have a significant meaning in relation to EA Debts. Thus, these should continue to be explored more specifically in relation to EA Debts theory.

A final important point, which is probably also not visible at first, is the connection between the categories. For this, the category Software and Data could be considered. Expert 7 also provides an interesting example for this *"... Data quality can have many sources or many causes, but one of them could also be that my application landscape is complex or simply that I have a redundant IT system that holds and maintains similar or the same data..."*. Accordingly, a hypothetical assumption regarding a connection between the categories of software and data would be that if the department introduces a new redundant system with redundant functions, this represents a potential EA Debt, according to the experts. This system could possibly also result in redundant data as a debt. In particular, the question arises as to what other invisible debts are also caused by consciously incurring a debt, such as introducing a redundant system. Expert 12 describes an important condition, which he determined by observations in various enterprises: *"[...] from my perception, it is actually the case that there are masses of them, yes, they are not all transparent, but they do exist, and they are not becoming fewer, but rather more..."* It becomes clear that the amount of debt in the company is high. Some experts reported that debts are evaluated and noted. The extent to which all debts are considered should be investigated further, as possibly many debts are interrelated and thus remain hidden. Accordingly, it was suggested that the interrelationships between the categories be examined quantitatively in further research, as it is significant which debts have an impact on other debts. Thus, hypotheses could be derived from the categories. A possible hypothesis would be that if software debts exist, data debts exist as well.

5 Assessing EA Debts

After having identified a set of EA Debts, it is important to follow up on how they develop to see if the determined measures are effective or not. Alternatively, already existing EA Debts might be found in the organization by other means, requiring an assessment of their severity. However, assessing the severity (or relevance) of an EA Debt is subjective and depends on the organizational context. Consequently, a threshold for distinguishing between EA Debts and acceptable issues is required, together with a process that allows stakeholders to make such a decision for each EA Debt. *"Redundant Data"* mentioned in the previous section might be a severe issue for one company, while it is still bearable for another as redundancy is part of business continuity. There is no general threshold for this distinction, and we, therefore, defined a process that supports corporate stakeholders in determining their individual threshold.

Values describing the severity of a Technical Debt, code smell, or other software metrics are a common concept for prioritization. By assigning the metrics thresholds, they will get a clearer meaning and help provide guidance in decision making [36]. In a study, developers were shown code and asked if it contained any design problems, and if so, they were asked to describe the nature and severity of them. It was discovered that some smells were not perceived as design problems by developers and that some smells were only seen as problems depending on their magnitude. Palomba et al. [73] also found strong implications that severity measures or thresholds are important when it comes to detecting smells.

The predefined values for software metric thresholds are few in the literature. Though, there are several suggested approaches for deriving and defining such thresholds [36]. Common thresholds describing the quality of code smells are defined in a similar way of benchmarking and were calculated through extensive analysis of a large number of different software systems [74]. Much research is conducted in the area of automating technical debt measures, but the developed tools are seldom based on earlier studies and their outcomes. These tools are often difficult for users to adopt and require a complex setup. Hence, Khomyakov et al. [75] concluded, after conducting an extensive systematic literature review, that the area of automated technical debt measures is still immature.

When looking at specific thresholds, they might differ in various systems. Depending on the complexity of the system, the requirements and what is considered acceptable or risk can differ [76]. When Sharma and Spinelli [77] investigated automatic approaches for detecting code smells, it was concluded that the investigated approaches could not define, capture, or specify the context of a code smell. Eisenberg [78] concluded that some thresholds for TD must be set differently in a system depending on the priority of the components. Also, new code or code under development was suggested to have other thresholds.

In a study conducted by Spadini et al. [79], severity thresholds for the branch of code smells, called test smells, were investigated. One of the most important parameters was experts' opinions regarding the derived thresholds. Mantyla and Lassenius [80] could see that the developers' perception of the code smell tended to be more aligned when less complex smells were evaluated; when the complexity increased, the deviation of perception also increased. Their suggestion is to evaluate thresholds with an approach that considers both qualitative subjective and quantitative source code metrics. Also, Eisenberg [78] emphasized that thresholds for Technical Debts must be adjusted and configured with user tests and analysis.

5.1 Process for Defining Thresholds

Salentin and Hacks [22] resonate about several similarities between software metrics, such as code smells and EA Debt. The fact that the research field of EA Debt is still new and quite immature, combined with the fact that most of the data concerning EA Debt models are confidential and not available for outsiders [81], [82] made an absence of data inevitable when conducting this study. Several researchers have suggested that users, developers, or experts should be involved when analyzing software metrics and setting thresholds for them [78], [79], [80]. Saravia et al. [36] presented a systematic process for defining expert-driven software metrics thresholds. A motivator for this process is that it can be used in the absence of data which is a common problem also regarding software metrics. The similarities of EA Debt and software metrics combined with the fact that it is suggested that "experts" should be involved when analyzing software metrics and that there is an absence of data in the field have been motivators for adopting Saravia et al. [36] systematic process for defining expert-driven software metrics thresholds. They argue that this process has an advantage when data is lacking. Hence, in this article, the process proposed in [36] has been tuned and adjusted to be used in the context of defining EA Debt thresholds.

The process builds on fuzzy logic, which is about applying formal logic in colloquial and non-formal language. Fogarty mentions four principles describing the properties of fuzzy logics [83]: *"Everything is, or is allowed to be, partial, i.e., a matter of degree, Everything is, or is allowed to be, imprecise (approximate), Everything is, or is allowed to be, granular (linguistic), Everything is, or is allowed to be, perception based."*

When defining a threshold for EA Debt, the three-phase process (cf. Figure 4) can be used to elicit data from one or many domain experts. A threshold designer leads the session and talks to the domain expert through the process. Before the process starts, the EA Debt or EA smell that is about to be evaluated is described for the domain expert to ensure that s/he is familiar with the concept and that there is an alignment regarding description, consequences, cause, and detection of the EA Debt. When testing the process during this study, the EA smell *"Hub-Like Modularization"* was chosen to define thresholds. The domain expert is supposed to be not only an expert on the EA domain but also on the specific system that the debt is about to be evaluated for. When conducting this study, a fictive case was presented to all the domain experts to make it possible to compare their perceptions of the process. For this reason, they were not experts on the specific system when conducting the study.

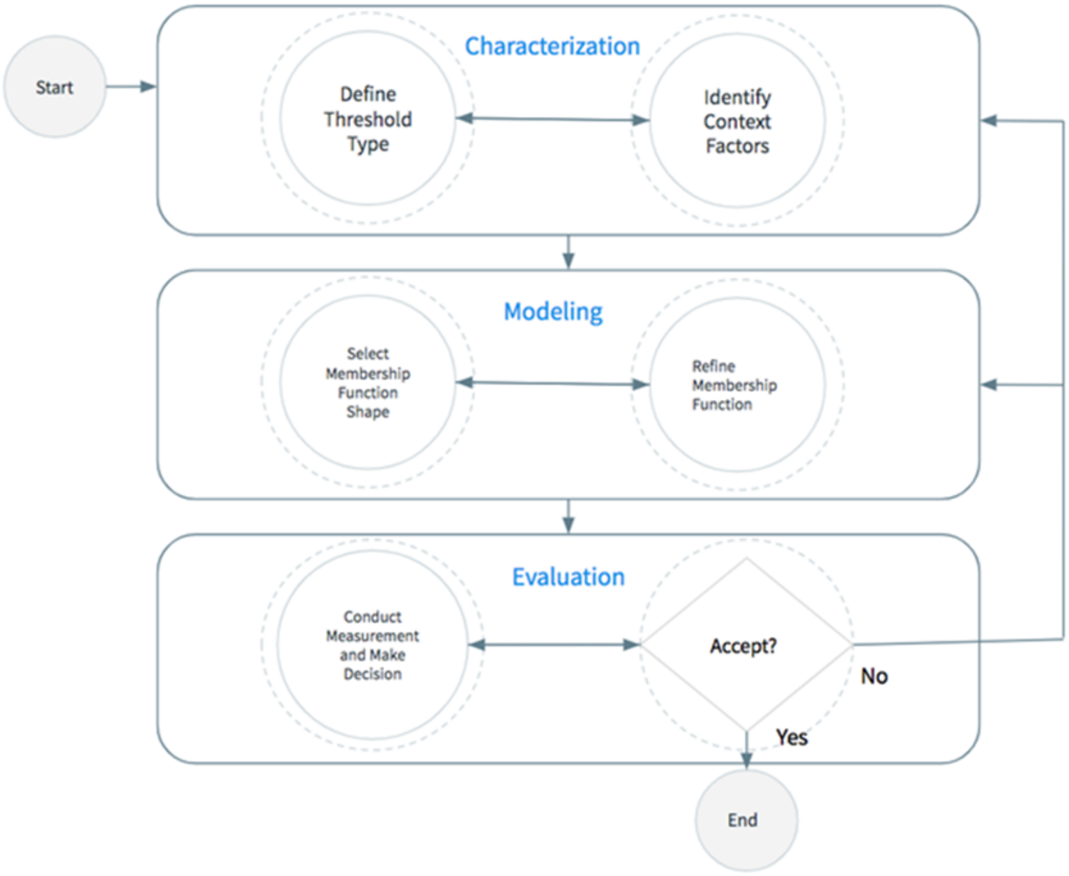


Figure 4. The process for defining a threshold for EA Debt

The characterization phase of the process includes the first four steps of the process where the two main activities are defining the threshold type (step 2) and identifying context factors (step 3):

1. First, the metric's semantic scale is defined. This scale represents how the thresholds are presented. The metric semantic scale can either be Boolean (e.g., OK and Not OK) or ordinal (e.g., Bad, Moderate and Good). The domain expert chooses what s/he believes suits the specific EA Debt threshold best.
2. The type of threshold should be identified in the next step, it can be either marginal (HigherThan(Θ) $\wedge$ LowerThan(Θ), where Θ is the reference value) or interval (HigherThan(α)

$\wedge$ LowerThan(β), where α is the lower bound and β is the upper bound). The domain expert chooses a suitable type depending on how s/he believes that different scenarios affect the threshold.

3. In the third step, context factors should be identified. These are attributes from key entities of the process that might influence the thresholds. In some situations, a specific number might have different severity depending on the context, and these factors are to be considered when deciding which measures to take. These can be diminishing or enhancing factors meaning that some factors make a scenario increase or decrease in severity due to the nature of the context. The domain expert is asked which context factors s/he can think of related to the specific debt and system that the thresholds should be defined for.
4. If multiple domain experts are involved in the process, steps 1–3 must be performed independently of each other with each expert to avoid bias. When this is done, there is an evaluation process called Planning Poker with all the domain experts involved. Planning Poker is a form of consensus-based voting often applied in the domain of software development[4]. It is important that the domain experts have the same perception regarding the characterization of thresholds. In this study, this step was not conducted during the testing due to time constraints from domain experts.

In the second phase, the thresholds are modeled; the main goal of this phase is to fuzzify the crisp values of a metric into fuzzy linguistic terms. Instead of numbers, they are expressed with words and sentences – as a linguistic variable. When doing this, the metric semantic scale (selected in the characterization phase step 1) is used. In the modeling phase, which includes the next three steps of the process (step 5 to step 7), the membership function is selected, and then it is refined:

5. The membership function illustrates how the domain experts believe that the threshold will behave in different scenarios. One membership function should be defined for each term in the linguistic variable (e.g., two membership functions for a Boolean linguistic variable). For instance, this means that one function will represent the probability of Not Ok, and the other will represent the probability of Ok in different scenarios. The threshold designer shows the experts possible shapes for the membership functions to give them guidance. On the x-axis, the scenarios are represented, and on the y-axis, the probability. For instance, if the threshold is marginal, the experts could choose a z-shape(B), sigmoid(D), or s-shape(E) as a reference. This is because if the threshold is believed to be marginal, the probability will, at some point, descend or increase and remain in that direction. If the threshold is interval, the experts could choose triangular(A), trapezoidal(C) or Gaussian(F) because the threshold would be a state in between two scenarios where the probability descends or increases (see Figure 5).
6. What-if scenarios are defined to configure the function and plot them. As a foundation of this process, the domain expert utilizes experience from earlier projects. To represent different what-if scenarios, an increasing scale was used. The domain experts were asked if the scale was perceived as sufficient for the evaluation. The probability was then mapped. To each scenario, for each term in the metric semantic scale, the domain expert assigned a probability, and the verbal probability scale presented by Renooij and Witterman [84] (see Figure 6), was used for this purpose. An example from the tests showing how the probability was mapped for different scenarios is illustrated in Table 3.
7. Given the numerical representation of the probabilities, the algorithm Akima Spline was used to fit the data in an appropriate distribution. The experts analyzed the distribution visually and judge if it reflects their intuition. An example of plotted values from one of the testing sessions is illustrated in Figure 7.

[4] https://www.mountaingoatsoftware.com/agile/planning-poker

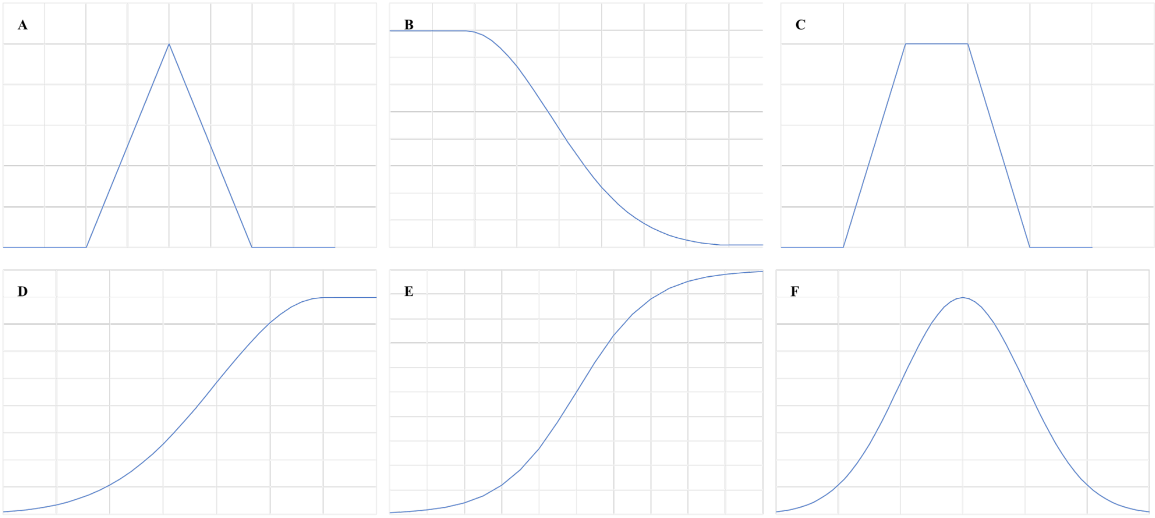


Figure 5. Six types of fuzzy membership functions: Triangular (A), Z-shape (B), Trapezoidal (C), Sigmoid (D), S-shape (E) and Gaussian (F)

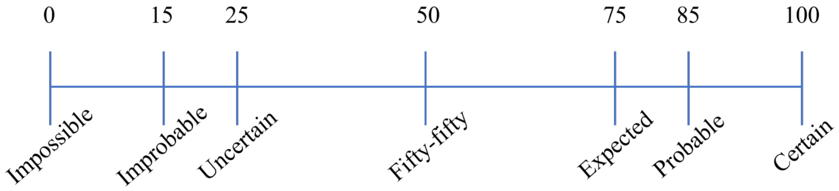


Figure 6. A verbal probability scale

Table 3. Probability and scenario for Bad, Moderate, and Good (Respondent 3)

Hub-Like Modularization	Bad	Moderate	Good
2	0	0	100
3	0	0	100
5	0	15	85
8	15	15	70
12	15	50	35
17	50	50	0
23	50	50	0
29	85	15	0
37	100	0	0
46	100	0	0

The final phase of the process is the evaluation phase. During this study, this phase was not tested and evaluated as suggested if this process were used in the industry. When applied in its true context, out in the industry, the evaluation phase would consist of a meeting initiated by the threshold designer. The results from implementing and using the defined thresholds should then be discussed and evaluated among the domain experts. This step was not possible to conduct in this study because EA Debt and EA smell are not yet something that EA practitioners are working with. But when the EA Debt consideration is spread and adopted, in the future, this step will be necessary to evaluate and be able to adjust and iterate the process back to characterization or modeling. In this

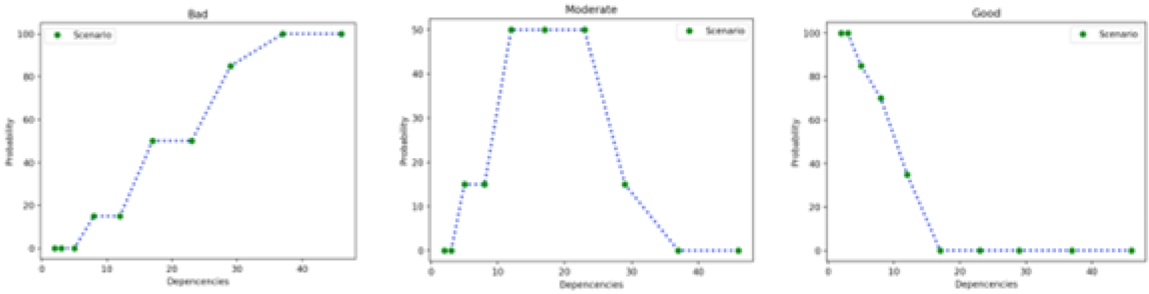


Figure 7. Plotted what-if scenarios for each term in the metric semantic scale (Respondent 3)

Table 4. Domain experts' feedback on the process

Common Topics	Respondent 1	Respondent 2	Respondent 3	Respondent 4	Respondent 5
Clearer process description needed	Agree	Agree	N/A	Agree	Agree
Context Factors are of high importance	Agree	Agree	Agree	Agree	Agree
Unnecessary to select membership functions	Agree	Agree	Agree	N/A	Disagree
Real-world EA model would increase understanding	Agree	Agree	Agree	Agree	N/A
Facilitates useful communication	Agree	N/A	Agree	Agree and Disagree	Agree and Disagree

study, the evaluation phase consisted of an interview after finishing the testing session, to collect the domain expert's perception of the process and be able to evaluate the process in that way.

5.2 Perceptions of the Process

When conducting the testing sessions of the process, five recurrent topics could be derived from the feedback. These could be used in the future when a process is tuned and refined to be used and implemented for defining EA Debt thresholds (see Table 4).

Four out of five domain experts stated that a clearer process description was needed for the process. The process was sometimes difficult for the participants to communicate, grasp and fully understand. A suggestion was to outline the objective and the way of reaching it before starting to define thresholds with the process.

There was a consensus among the domain experts that the context factors are of high importance. A suggestion was to focus even more on contextual factors to make the process more useful. The context factors could also be an important parameter to foster communication among stakeholders, which is much what EA and EA Debt are all about. When evaluating debt, dependencies, and context are necessary to be considered.

What was also a common topic was part of the process where a membership function was selected. Three respondents thought this was an unnecessary part of defining thresholds for EA Debt; it was seen as difficult to grasp how the function should be selected. In contrast, one of the respondents mentioned the membership function as one thing that made perfect sense. It was described that selecting a membership function before giving each scenario a probability forced the domain expert to reflect upon the probability selection in the later step of the process.

Four out of five domain experts thought that a realistic EA model would increase the understanding of the process and the scenarios for defining thresholds.

The last topic that came up among the domain experts was how the process could facilitate useful communication. Two of the experts thought that it could, and it was mentioned that it would be fun, to define thresholds with this process and that EA is very much about communication. It was also mentioned that the communication part regarding the context factors could be very useful but that the right incentives for realizing the adoption of the process are needed. The idea of the process was seen as positive, but it was questioned if it would be possible to realize it.

5.3 Drivers for EA Debt Thresholds

When conducting the interviews, five different themes for drivers were found. These are cost and time, responsibility and engagement, context, important areas, and machine learning. Below the respondent perceptions are summarised.

Cost and Time. In the long run, thresholds for EA Debt could help save time, a common problem is that EA today is prioritized too late according to four of five respondents. There is a focus on activities that bring in money and since EA is no such activity it is instead seen as a cost. What often has a high priority is time to market, and EA is prioritized when major problems occur, and the prioritization is inevitable. Incentives and arguments for change happen easier in domains that drive revenue. When prioritizing EA, time is a factor, and since EA rarely is included in a project plan, it can be difficult to extend the time plan and the budget. Therefore, it would have to be included from the beginning in order to be motivated. If EA would be a part of building an organization from the beginning, this would not be an issue; but since EA is often applied when an organization grows large, and the complexity increases, the cost and scope also increase. When an organization sees the need for EA they reach a more mature state, and it is then easier to prioritize it. When a mature state is reached there usually is a common understanding that EA is a long-term investment. Therefore, it is important to enlighten architecture and manage it at an early stage. Putting time and effort into analyzing EA Debt could be helpful and prevent organizations from making the same mistakes again. A common problem is that mistakes tend to repeat themselves.

Some respondents state that a process for defining EA Debt thresholds could eliminate unnecessary discussions and, in that way, save time for the organization. However, to do so EA Debt thresholds must be communicated, augmented, and motivated for decision makers to make them understand the benefits of EA Debt thresholds. That the process of defining thresholds possibly is time-consuming and involves several stakeholders could make it difficult to realize and implement it. It can also be a challenge for stakeholders to agree upon the thresholds. Having a collection of thresholds could be helpful for this purpose. On the other hand, the communication part and the discussion can be seen as more important than the threshold itself. On the more technical levels, an automatic approach might be possible to implement, this would save time, but also the art is lost. Experience is an important factor, and if an automatic approach is applied, there would have to be manual checks as well.

Responsibility and Engagement. How EA is communicated and the dynamics in the organization, due to the fact that EA is under IT in the hierarchy, is mentioned by 4 of 5 respondents. How EA is communicated to stakeholders impacts how they will assimilate its meaning and objectives of it. Often a pedagogic way of explaining EA to stakeholders is lacking in organizations, and this could be an obstacle when motivating EA. Since EA is seen as an IT-related matter, it is often left to be handled and solved by IT departments. The relationship between EA practitioners and other stakeholders must build on trust and be nurtured. Communication is often the key, and EA is a long and slow negotiation where no unnecessary friction should be created. It is a problem that EA is organized under IT. If it would be higher up in the hierarchy, under business operations, architecture should stern the development. It is a problem that departments that have budget and project responsibilities do not understand nor prioritize EA.

Several respondents mentioned that the adoption of EA Debt in the industry is important and complex. For EA Debt thresholds to be applied in the industry, EA Debt first would have to be adopted. On the other hand, EA Debt thresholds could be a way of communicating with stakeholders and provide a more concrete way of talking about EA and EA Debt. Measuring EA Debt might also be of interest to practitioners since it would raise awareness and bring transparency to the actual problems. Even if numbers might not always be sufficient, they can be used as indicators and start a discussion. Presenting, for instance, the top 10 EA Debts would make it easier for stakeholders to take in the information and listen and understand. The technical aspects are often easier to measure but in this field knowledge already exists and therefore it is less interesting to do so. IT departments often have a strong focus on technology but less on other businesses which makes their point more difficult to prove. Competence for driving the EA project and holistic business competence is missing. To be able to communicate architectural changes and maintenance, the question must be discussed on the top executive level. Thresholds could help to motivate when it is time to refactor or invest in new projects.

Context. 5 of 5 respondents brought up that context was a significant matter when talking about EA Debt and its motivators. When working with EA compromises are necessary to make, often due to other projects and dependencies. Some EA Debts might be more important and sufficient in certain organizations. There is no size that fits all and reality is much more complex and diverse than what a certain framework can predict and plan. There can be different requirements for different models depending on the context. If the reason for a certain debt can be justified it might not have to be adjusted.

It might be difficult to set thresholds beforehand, but if thresholds would have a default value they could later be adjusted. Then they could be standardized and generalized to be used as indicators. The size of the company or how many systems there are could then be parameters for deciding on which default thresholds to apply. But the thresholds are often unique in their context in an organization since requirements are different depending on the context. The ones that are similar in different organizations are often not the ones that are more difficult to set and predict. When there are more than two architects, a dialogue or checklist could be useful in order to align the work and reduce discussions. Definitions of complexity regarding EA and requirements are needed to realize EA Debt and to find thresholds.

6 Summary and Future Work

Within this work, we aimed to solve three research objectives: the developing of an interview format to discover EA Debts in organizations; defining a method to assess these EA Debts; and building the foundation for a repository of EA Debts.

To address the first research objective, we designed an interview guideline comprised of six phases inspired by pain point analysis and the 5-Why method. We tried the guideline in 12 different interviews with experts from different domains. Besides the confirmation of our proposed interview based approach, we were also able to identify a set of 15 categories of EA Debts, that arose regularly in the different organizations. To ease the representation of these categories and to address the third research objective, we structured the identified categories into a taxonomy. This taxonomy can serve as a future starting point to identify known and unknown EA Debts in organizations. Finally, we proposed and demonstrated a process to determine thresholds for found EA Debts that classify the EA Debt's quality (i.e., is it "Good" or "Bad").

Our research leaves different directions for future research open. Firstly, the found categories of EA Debts should be further refined and complemented by conducting more interviews with different organizations. Secondly, it is interesting to elaborate on the perception of the EA Debts. We assume that different stakeholders perceive EA Debts differently, based on the expectations related to their roles. This should also manifest within different thresholds to be found among those stakeholders.

Thirdly, all our findings need further evaluation as the number of conducted interviews is rather low for both the interview format and the threshold determination. The latter has been proven to be applicable in the field of software engineering, but further research should be invested to align it more with the needs of EA stakeholders. Moreover, the usefulness of the entire concept of EA Debts is so far solely anecdotally, and the more formal proof is missing.

References

[1] B. Tabrizi, E. Lam, K. Girard, and V. Irvin, "Digital Transformation Is Not About Technology," *Harvard Business Review*, pp. 2–7, 2019. [Online]. Available: https://hbr.org/2019/03/digital-transformation-is-not-about-technology

[2] R. Morakanyane, A. Grace, and P. O'Reilly, "Conceptualizing digital transformation in business organizations: A systematic review of literature," in *30th Bled eConference: Digital Transformation – From Connecting Things to Transforming our Lives, BLED 2017*, 2017, pp. 427–444. [Online]. Available: https://doi.org/10.18690/978-961-286-043-1.30

[3] D. Ori, "Towards detecting misalignment symptoms: An alignment perspective-driven architecture-matching framework (extended abstract)," *CEUR Workshop Proceedings*, vol. 1701, pp. 32–35, 2016.

[4] C. Matt, T. Hess, and A. Benlian, "Digital Transformation Strategies," *Business and Information Systems Engineering*, vol. 57, no. 5, pp. 339–343, 2015. [Online]. Available: https://doi.org/10.1007/s12599-015-0401-5

[5] A. Bharadwaj, O. A. El Sawy, P. A. Pavlou, and N. Venkatraman, "Digital business strategy: Toward a next generation of insights," *MIS Quarterly: Management Information Systems*, vol. 37, no. 2, pp. 471–482, 2013. [Online]. Available: https://doi.org/10.25300/MISQ/2013/37:2.3

[6] M. Lankhorst, *Enterprise Architecture at Work*. Berlin, Heidelberg: Springer, 2009. [Online]. Available: https://doi.org/10.1007/978-3-642-01310-2

[7] J. Lapalme, "Three Schools of Thought on Enterprise Architecture," *IT Professional*, vol. 14, no. 6, pp. 37–43, 2012. [Online]. Available: https://doi.org/10.1109/MITP.2011.109

[8] G. Wierda, *Mastering ArchiMate*, edition ii ed. R&A, 2017.

[9] T. Tamm, P. B. Seddon, G. Shanks, and P. Reynolds, "How does enterprise architecture add value to organisations?" *Communications of the Association for Information Systems*, vol. 28, no. 1, pp. 141–168, 2011. [Online]. Available: https://doi.org/10.17705/1CAIS.02810

[10] S. Kaisler and F. Armour, "15 Years of Enterprise Architecting at HICSS : Revisiting the Critical Problems," in *Proceedings of the 50th Hawaii International Conference on System Sciences 2017*, 2017, pp. 4807–4816. [Online]. Available: https://doi.org/10.24251/HICSS.2017.585

[11] S. Kotusev, "Enterprise Architecture: What Did We Study?" *International Journal of Cooperative Information Systems*, vol. 26, no. 4, 2017. [Online]. Available: https://doi.org/10.1142/S0218843017300029

[12] G. Wierda, *Chess and the Art of Enterprise Architecure*. R&A, 2015.

[13] S. Bente, U. Bombosch, and S. Langade, *Collaborative Enterprise Architecture: Enriching EA with Lean, Agile, and Enterprise 2.0 Practices*. Morgan Kaufmann, 2012. [Online]. Available: https://doi.org/10.1016/C2011-0-69682-6

[14] S. Kurnia, S. Kotusev, R. Dilnutt, P. Taylor, G. Shanks, and S. Milton, "Artifacts, activities, benefits and blockers: Exploring enterprise architecture practice in depth," *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2020-Janua, no. January, pp. 5583–5592, 2020. [Online]. Available: https://doi.org/10.24251/HICSS.2020.687

[15] M. Lange, J. Mendling, and J. Recker, "An empirical analysis of the factors and measures of Enterprise Architecture Management success," *European Journal of Information Systems*, vol. 25, no. 5, pp. 411–431, 2016. [Online]. Available: https://doi.org/10.1057/ejis.2014.39

[16] D. Pascot, F. Bouslama, and S. Mellouli, "Architecturing large integrated complex information systems: An application to healthcare," *Knowledge and Information Systems*, vol. 27, no. 1, pp. 115–140, 2011. [Online]. Available: https://doi.org/10.1007/s10115-010-0292-1

[17] J. L. Williams and J. T. Stracener, "First Steps in the Development of a Program Organizational Architectural Framework (POAF)," *Systems Engineering*, vol. 16, no. 1, p. 45–70, 2013. [Online]. Available: https://doi.org/10.1002/sys.21218

[18] J. J. Korhonen, J. Lapalme, D. McDavid, and A. Q. Gill, "Adaptive Enterprise Architecture for the Future: Towards a Reconceptualization of EA," *Proceedings - CBI 2016: 18th IEEE Conference on Business Informatics*, vol. 1, no. February 2018, pp. 272–281, 2016. [Online]. Available: https://doi.org/10.1109/CBI.2016.38

[19] D. Wissal, D. Karim, and K. Laila, "Adaptive Enterprise Architecture: Initiatives and Criteria," *7th International Conference on Control, Decision and Information Technologies, CoDIT 2020*, pp. 557–562, 2020. [Online]. Available: https://doi.org/10.1109/CoDIT49905.2020.9263891

[20] S. Hacks, H. Hofert, J. Salentin, Y. C. Yeong, and H. Lichter, "Towards the Definition of Enterprise Architecture Debts," in *2019 IEEE 23rd EDOCW*. IEEE, 10 2019, pp. 9–16. [Online]. Available: https://doi.org/10.1109/EDOCW.2019.00016

[21] P. Alexander, S. Hacks, J. Jung, H. Lichter, U. Steffens, and Ö. Uludağ, "A framework for managing enterprise architecture debts – Outline and research directions," in *CEUR Workshop Proceedings*, vol. 2628, 2020.

[22] J. Salentin and S. Hacks, "Towards a Catalog of Enterprise Architecture Smells," in *WI2020 Community Tracks*. GITO Verlag, 3 2020, pp. 276–290. [Online]. Available: https://doi.org/10.30844/wi_2020_y1-salentin

[23] J. Jung, S. Hacks, T. De Gooijer, M. Kinnunen, and K. Rehring, "Revealing Common Enterprise Architecture Debts: Conceptualization and Critical Reflection on a Workshop Format Industry Experience Report," *Proceedings – IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, pp. 271–278, 2021. [Online]. Available: https://doi.org/10.1109/EDOCW52865.2021.00058

[24] M. M. Lankhorst, D. A. Quartel, and M. W. Steen, "Architecture-based IT portfolio valuation," *Lecture Notes in Business Information Processing*, vol. 69 LNBIP, pp. 78–106, 2010. [Online]. Available: https://doi.org/10.1007/978-3-642-16770-6_4

[25] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Heidelberg, New York, Dordrecht, London: Springer, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-33143-5

[26] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004. [Online]. Available: https://doi.org/10.2307/25148625

[27] J. R. Venable, J. Pries-Heje, and R. L. Baskerville, "Choosing a Design Science Research Methodology," in *ACIS 2017 Proceedings*, 2017, p. 112.

[28] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, "Action design research," *MIS quarterly*, vol. 35, no. 1, pp. 37–56, 3 2011. [Online]. Available: https://doi.org/10.2307/23043488

[29] M. Smajevic, S. Hacks, and D. Bork, "Using Knowledge Graphs to Detect Enterprise Architecture Smells," in *The Practice of Enterprise Modeling. PoEM 2021. Lecture Notes in Business Information Processing*, Serral E., Stirna J., Ralyté J., and Grabis J., Eds. Springer, Cham, 2021, vol. 432, pp. 48–63. [Online]. Available: https://doi.org/10.1007/978-3-030-91279-6_4

[30] A. Witzel, *Das problemzentrierte interview*. Beltz, 1985 (in German).

[31] S. Lamnek and C. Krell, *Qualitative sozialforschung*. Psychologie Verlags Union München, 2005 (in German).

[32] D. Okes, *Root cause analysis: The core of problem solving and corrective action*. Quality Press, 2019.

[33] P. H. Pendse, *Business Analysis: Solving Business Problems by Visualizing Effective Processes and IT Solutions*. PHI Learning, 2015.

[34] A. Bogner, B. Littig, and W. Menz, *Interviews mit Experten: eine praxisorientierte Einführung*. Springer-Verlag, 2014. [Online]. Available: https://doi.org/10.1007/978-3-531-19416-5

[35] J. Gläser and G. Laudel, *Experteninterviews und qualitative Inhaltsanalyse: als Instrumente rekonstruierender Untersuchungen*. Springer-Verlag, 2009. [Online]. Available: https://doi.org/10.1007/978-3-531-91538-8

[36] R. Saraiva, M. Perkusich, H. Almeida, and A. Perkusich, "A systematic process to define expert-driven software metrics thresholds," in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, vol. 2019-July. Knowledge Systems Institute Graduate School, 2019, pp. 171–176. [Online]. Available: https://doi.org/10.18293/SEKE2019-217

[37] S. D. Oliner and D. E. Sichel, "The resurgence of growth in the late 1990s: Is information technology the story?" *SSRN*, 2000. [Online]. Available: https://doi.org/10.2139/ssrn.233139

[38] S. Vodanovich, D. Sundaram, and M. Myers, "Research commentary—digital natives and ubiquitous information systems," *Information Systems Research*, vol. 21, pp. 711–723, 2010. [Online]. Available: https://doi.org/10.1287/isre.1100.0324

[39] J. N. Luftman, "Key issues for IT executives 2004," *MIS Quarterly Executive*, vol. 7, 2005.

[40] J. N. Luftman and T. Ben-Zvi, "Key issues for IT executives 2010: Judicious IT investments continue post-recession," *MIS Quarterly Executive*, vol. 9, 2010.

[41] P. Saint-Louis and J. Lapalme, "Investigation of the lack of common understanding in the discipline of enterprise architecture: A systematic mapping study," in *20th International Enterprise Distributed Object Computing Workshop (EDOCW)*, U. Franke, J. Lapalme, and P. Johnson, Eds., 2016, pp. 1–9. [Online]. Available: https://doi.org/10.1109/EDOCW.2016.7584364

[42] D. Simon, K. Fischbach, and D. Schoder, "An exploration of enterprise architecture research," *Communications of the Association for Information Systems*, vol. 32, pp. 1–72, 2013. [Online]. Available: https://doi.org/10.17705/1CAIS.03201

[43] W. F. Boh and D. Yellin, "Using enterprise architecture standards in managing information technology," *Journal of Management Information Systems*, vol. 23, pp. 163–207, 2006. [Online]. Available: https://doi.org/10.2753/MIS0742-1222230307

[44] S. Aier and R. Winter, "Virtual decoupling for IT/business alignment – conceptual foundations, architecture design and implementation example," *Business & Information Systems Engineering*, vol. 1, pp. 150–163, 2009. [Online]. Available: https://doi.org/10.1007/s12599-008-0010-7

[45] L. Plazaola, J. Flores, E. Silva, N. Vargas, and M. Ekstedt, "An approach to associate strategic business-IT alignment assessment to enterprise architecture," 2007.

[46] Ö. Uludağ, M. Kleehaus, X. Xu, and F. Matthes, "Investigating the role of architects in scaling agile frameworks," in *2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE, 2017, pp. 123–132. [Online]. Available: https://doi.org/10.1109/EDOC.2017.25

[47] Ö. Uludag, N. Reiter, and F. Matthes, "What to Expect from Enterprise Architects in Large-Scale Agile Development? A Multiple-Case Study," in *25th AMCIS*, 2019.

[48] F. Gampfer, A. Jürgens, M. Müller, and R. Buchkremer, "Past, current and future trends in enterprise architecture—A view beyond the horizon," *Computers in Industry*, vol. 100, pp. 70–84, 9 2018. [Online]. Available: https://doi.org/10.1016/j.compind.2018.03.006

[49] W. Cunningham, "The WyCash portfolio management system," *ACM SIGPLAN OOPS Messenger*, vol. 4, no. 2, pp. 29–30, 1993. [Online]. Available: https://doi.org/10.1145/157710.157715

[50] Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," *Journal of Systems and Software*, vol. 101, pp. 193–220, 2015. [Online]. Available: https://doi.org/10.1016/j.jss.2014.12.027

[51] P. Kruchten, R. L. Nord, and I. Ozkaya, "Technical Debt: From Metaphor to Theory and Practice," *IEEE Software*, vol. 29, no. 6, pp. 18–21, 2012. [Online]. Available: https://doi.org/10.1109/MS.2012.167

[52] C. Seaman and Y. Guo, "Measuring and monitoring technical debt," *Advances in Computers*, vol. 82, pp. 25–46, 2011. [Online]. Available: https://doi.org/10.1016/B978-0-12-385512-1.00002-5

[53] S. McConnell, "Managing Technical Debt Slides," 2007.

[54] T. Theodoropoulos, M. Hofberg, and D. Kern, "Technical Debt from the Stakeholder Perspective," in *Proceedings of the 2Nd Workshop on Managing Technical Debt*, ser. MTD '11. New York, NY, USA: ACM, 2011, pp. 43–46. [Online]. Available: https://doi.org/10.1145/1985362.1985373

[55] T. Klinger, P. Tarr, P. Wagstrom, and C. Williams, "An Enterprise Perspective on Technical Debt," in *Proceedings of the 2Nd Workshop on Managing Technical Debt*, ser. MTD '11. New York, NY, USA: ACM, 2011, pp. 35–38. [Online]. Available: https://doi.org/10.1145/1985362.1985371

[56] B. Curtis, J. Sappidi, and A. Szynkarski, "Estimating the Principal of an Application's Technical Debt," *IEEE Software*, vol. 29, no. 6, pp. 34–42, 2012. [Online]. Available: https://doi.org/10.1109/MS.2012.156

[57] R. L. Nord, I. Ozkaya, P. Kruchten, and M. Gonzalez-Rojas, "In search of a metric for managing architectural technical debt," in *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*, 2012, pp. 91–100. [Online]. Available: https://doi.org/10.1109/WICSA-ECSA.212.17

[58] B.-D. Lehmann, P. Alexander, H. Lichter, and S. Hacks, "Towards the identification of process anti-patterns in enterprise architecture models," vol. 2767. CEUR-WS, 2020, pp. 47–54. [Online]. Available: http://ceur-ws.org/Vol-2767/06-QuASoQ-2020.pdf

[59] B. Tieu and S. Hacks, "Determining enterprise architecture smells from software architecture smells," vol. 02. IEEE, 2021, pp. 134–142. [Online]. Available: https://doi.org/10.1109/CBI52690.2021.10064

[60] Y. Yeong, S. Hacks, and H. Lichter, "Prioritization of ea debts facilitating portfolio theory," vol. 2511, 2019.

[61] L. Liss, H. Kämmerling, P. Alexander, and H. Lichter, "Towards a catalog of refactoring solutions for enterprise architecture smells," B. Gan, E. L. Ouh, B. Wadhwa, S. Chawla, H. Lichter, S. Aydin, T. Sunetnanta, and T. Anwar, Eds., vol. 3062. CEUR-WS.org, 2021, pp. 60–69. [Online]. Available: http://ceur-ws.org/Vol-3062/Paper09_QuASoQ.pdf

[62] B. Andersen and T. N. Fagerhaug, "The ASQ Pocket Guide to Root Cause Analysis," Tech. Rep. [Online]. Available: http://qualitypress.asq.org.

[63] J. Loheide, *Finanzmarkt ohne Grenzen?* Springer, 2008 (in German). [Online]. Available: https://doi.org/10.1007/978-3-531-90869-4

[64] E. Mehrmann, *Vom Konzept zum Interview: Informationsgespräche richtig planen und führen*. Econ-Taschenbuch-Verlag, 1995 (in German).

[65] R. Kaiser, "Qualitative experteninterviews: Konzeptionelle grundlagen und praktische durchführung (2., aktualisierte auflage)," *Elemente der Politik*, 2021 (in German). [Online]. Available: https://doi.org/10.1007/978-3-658-30255-9

[66] H. O. Mayer, *Interview und schriftliche Befragung: Grundlagen und Methoden empirischer Sozialforschung*. Walter de Gruyter, 2012 (in German). [Online]. Available: https://doi.org/10.1524/9783486717624

[67] M. Meuser and U. Nagel, "Expertinneninterviews—vielfach erprobt, wenig bedacht," in *Das Experteninterview*. Springer, 2002, pp. 71–93 (in German). [Online]. Available: https://doi.org/10.1007/978-3-322-93270-9_3

[68] U. Kuckartz, *Qualitative inhaltsanalyse*. Beltz Juventa, 2012 (in German).

[69] A. Diekmann, "Empirische sozialforschung," *Grundlagen, Methoden, Anwendungen*, vol. 18, 2007 (in German).

[70] R. C. Nickerson, U. Varshney, and J. Muntermann, "A method for taxonomy development and its application in information systems," *European Journal of Information Systems*, vol. 22, no. 3, pp. 336–359, 2013. [Online]. Available: https://doi.org/10.1057/ejis.2012.26

[71] S. Hacks and J. Jung, "Enterprise architecture debts–a concept to manage EA evolution?" in *ICIS 2020 TREOs*, 2020.

[72] M. Op't Land, E. Proper, M. Waage, J. Cloo, and C. Steghuis, *Enterprise architecture: creating value by informed governance*. Springer Science & Business Media, 2008. [Online]. Available: https://doi.org/10.1007/978-3-540-85232-2

[73] F. Palomba, G. Bavota, M. D. Penta, R. Oliveto, and A. D. Lucia, "Do they really smell bad? a study on developers' perception of bad code smells." Institute of Electrical and Electronics Engineers Inc., 2014, pp. 101–110. [Online]. Available: https://doi.org/10.1109/ICSME.2014.32

[74] M. Aniche, C. Treude, A. Zaidman, A. Van Deursen, and M. A. Gerosa, "Satt: Tailoring code metric thresholds for different software architectures," in *2016 IEEE 16th international working conference on source code analysis and manipulation (SCAM)*. IEEE, 2016, pp. 41–50. [Online]. Available: https://doi.org/10.1109/SCAM.2016.19

[75] I. Khomyakov, Z. Makhmutov, R. Mirgalimova, and A. Sillitti, "An analysis of automated technical debt measurement," in *International Conference on Enterprise Information Systems*. Springer, 2019, pp. 250–273. [Online]. Available: https://doi.org/10.1007/978-3-030-40783-4_12

[76] G. Suryanarayana, G. Samarthyam, and T. Sharma, *Refactoring for software design smells: managing technical debt*. Morgan Kaufmann, 2014. [Online]. Available: https://doi.org/10.1016/B978-0-12-801397-7.00001-1

[77] T. Sharma and D. Spinellis, “A survey on software smells,” *Journal of Systems and Software*, vol. 138, pp. 158–173, 2018. [Online]. Available: https://doi.org/10.1016/j.jss.2017.12.034

[78] R. J. Eisenberg, “A Threshold Based Approach to Technical Debt,” *SIGSOFT Softw. Eng. Notes*, vol. 37, no. 2, pp. 1–6, 4 2012. [Online]. Available: https://doi.org/10.1145/2108144.2108151

[79] D. Spadini, M. Schvarcbacher, A.-M. Oprescu, M. Bruntink, and A. Bacchelli, “Investigating severity thresholds for test smells,” in *Proceedings of the 17th International Conference on Mining Software Repositories*, 2020, pp. 311–321. [Online]. Available: https://doi.org/10.1145/3379597.3387453

[80] M. V. Mäntylä and C. Lassenius, “Subjective evaluation of software evolvability using code smells: An empirical study,” *Empirical Software Engineering*, vol. 11, no. 3, pp. 395–431, 2006. [Online]. Available: https://doi.org/10.1007/s10664-006-9002-8

[81] A. Barbosa, A. Santana, S. Hacks, and N. v. Stein, “A taxonomy for enterprise architecture analysis research,” in *21st International Conference on Enterprise Information Systems*, vol. 2. SciTePress, 2019, pp. 493–504. [Online]. Available: https://doi.org/10.5220/0007692304930504

[82] I. Gmati, I. Rychkova, and S. Nurcan, “On the way from research innovations to practical utility in enterprise architecture: The build-up process,” *International Journal of Information System Modeling and Design (IJISMD)*, vol. 1, no. 3, pp. 20–44, 2010. [Online]. Available: https://doi.org/10.4018/jismd.2010070102

[83] P. Fogarty, “Fuzzy logic, a logician’s perspective,” in *Fuzzy Logic*. Springer, 2021, pp. 1–9. [Online]. Available: https://doi.org/10.1007/978-3-030-66474-9_1

[84] S. Renooij and C. Witteman, “Talking probabilities: communicating probabilistic information with words and numbers,” *International Journal of Approximate Reasoning*, vol. 22, no. 3, pp. 169–194, 1999. [Online]. Available: https://doi.org/10.1016/S0888-613X(99)00027-4