Friedrich-Alexander-Universität Erlangen-Nürnberg
Technische Fakultät, Department Informatik

Katharina Kunz

MASTER THESIS

Developing a Domain Analysis Procedure based on Grounded Theory Method

Submitted on 28.05.2015

Supervisors: Prof. Dr. Dirk Riehle, M.B.A.
Andreas Kaufmann, M.Sc.
Professur für Open-Source-Software
Department Informatik, Technische Fakultät
Friedrich-Alexander University Erlangen-Nürnberg

Versicherung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Nürnberg, 28.05.2015

License

Nürnberg, 28.05.2015

Abstract

Domain analysis is the process of analyzing and modelling the domain in which a future software system is supposed to operate. It is an essential step in requirements engineering (RE) and therefore critical for the success of software development projects. However, common methods for deriving a domain model from natural language descriptions do not address the difficulties of abstracting a complex domain sufficiently and depend on the analyst's experience and expertise. Grounded theory method (GTM) offers a techniques for breaking up and abstracting qualitative data by developing and relating concepts. Its use can therefore improve the procedure of extracting the important entities of a domain and their relationships, while ensuring traceability between the data and the derived domain model. This thesis shows how GTM has to be adapted for its successful utilization in RE. For this purpose, we applied GTM to a domain analysis example and derived a systematic procedure for domain analysis.

Keywords: Domain Analysis, Domain Model, Requirements Engineering, Qualitative Data Analysis, Grounded Theory Method

Zusammenfassung

Der Prozess der Domänenanalyse dient der Analyse und Modellierung der Domäne, in welcher ein zukünftiges Software System angewendet werden soll. Sie ist ein wichtiger Schritt zur Definition von Anforderungen und trägt damit entscheidend zum Erfolg von Software Entwicklungsprojekten bei. Gängige Methoden zur Ableitung eines Domänenmodells aus natürlichsprachlichen Beschreibungen befassen sich jedoch nicht ausreichend mit den Schwierigkeiten, die eine Abstraktion einer komplexen Domäne mit sich bringt, wodurch das Ergebnis stark von der Kompetenz und Erfahrung des Analysten abhängt. Die Grounded Theory Methode bietet Techniken zur Auftrennung und systematischen Abstraktion qualitativer Daten durch die Identifikation und Entwicklung von Konzepten und Zusammenhängen. Die Anwendung dieser Methode in der Domänenanalyse kann daher das Extrahieren der wichtigen Entitäten einer Domäne und deren Beziehungen zueinander unterstützen und damit die Rückverfolgbarkeit von dem erstellten Domänenmodell zu den zugrundeliegenden Daten sicherstellen. Diese Arbeit zeigt notwendige Anpassungen der Grounded Theory Methode für eine erfolgreiche Nutzung bei der Definition von Anforderungen. Zu diesem Zweck wurde die Grounded Theory Methode auf eine beispielhafte Domänenanalyse angewendet und eine systematische Methode zur Domänenanalyse abgeleitet.

Contents

List of Figures

List of Tables

List of Abbreviations

GTM	Grounded Theory Method
HR	Human Resources
NFR	Non-Functional Requirements
OOAD	Object Oriented Analysis and Design
QDA	Qualitative Data Analysis
RE	Requirements Engineering
UML	Unified Modelling Language

1 Introduction

1.1 Original Thesis Goals

The original thesis goal was to produce a requirements specification and glossary using GTM in a real world industry RE project for a human resources (HR) software parallel to requirements analysts in order to compare the results and to show how GTM can be used to improve RE.

1.2 Changes to Thesis Goals

One month into the project, the cooperation with the original industry partner was canceled. We therefore decided to substitute the analysis of the industry project with an independent domain analysis within the same domain as the original project.

The goal of the independent analysis was to evaluate the use of GTM for the creation of a domain model, using in-depth interviews with experts working in the field of HR development as a data source. The challenge remained the application of the original qualitative research technique within a RE context, identifying similarities and differences, showing weaknesses and proposing possible adaptions of the analysis process that better assist in the creation of a domain model.

2 Research Chapter

2.1 Introduction

2.1.1 Domain Analysis

According to studies by Curtis, Krasner, and Iscoe (1988) and Hofmann and Lehner (2001), successful software development requires in-depth domain knowledge. A requirements engineer must understand the user's domain in order to determine the purpose of a software system and the respective requirements (Balzert, 2009; Robertson & Robertson, 2006). Thus, in the early stages of RE, the domain in which the software system is supposed to operate, is analyzed and a domain model is developed. This conceptual model is commonly represented as a Unified Modelling Language (UML) class diagram and describes the important entities of the domain and their structural relationships without considering the possible technical realization (Broy, 2013; Rumpe, 2011). The model provides a basis for later analysis and design models and can also be seen as a visual dictionary, as it visualizes and relates important terms of one domain (Larman, 2010; Rupp & SOPHISTen, 2014). With the help of a domain model, misunderstandings and synonyms or homonyms used by different stakeholders can be clarified and relationships between entities of the domain can be understood easily, facilitating better communication amongst stakeholders (Rupp & SOPHISTen, 2014). As software systems will be integrated into their problem domains even tighter in the future, domain analysis is an essential step for developing software systems of high quality (Broy, 2013). Therefore, new approaches to improve domain analysis are of particular interest.

2.1.2 Challenges in Domain Analysis

A common method to define conceptual classes and attributes for a domain model is to identify nouns from use cases or other natural language descriptions sentence by sentence to create a list of candidate classes. This list is then perused to check which of the candidate classes are relevant and correct. In an iterative process, attributes and associations are extracted from the descriptions to build a domain model (Larman, 2010; Rosenberg & Scott, 2001; Wazlawick, 2014). This approach is problematic because it does not give much guidance to novice analysts, thus the outcome depends on the analyst's experience in extracting conceptual classes. The domain a software system is supposed to operate in is often highly complex and the different needs and perspectives of all stakeholders need to be addressed (Browne & Rogich, 2001; Ebert, 2012; Rupp, Simon, & Hocker, 2009). This is even more difficult when developing a standard software, which has to address the needs of stakeholders in different businesses (Balzert, 2009). In addition, the gathered data might be unstructured, inconsistent and incomplete and words can be ambiguous or two noun phrases can represent the same conceptual class (Balzert, 2009; Larman, 2010; Rupp & SOPHISTen, 2014; Wazlawick, 2014). An analysis method must address these issues in order guide the analyst in developing a consistent and complete conceptual model of the domain (Browne & Rogich, 2001).

The analysis method must also ensure traceability in order to make later interrogations possible and to incorporate changes in requirements (Easterbrook & Nuseibeh, 2000; Rupp & SOPHISTen, 2014). This means that the origin and the relation to other RE artefacts of every domain model element can be retraced. Therefore, a well-documented systematic procedure for domain analysis is needed (Kleuker, 2013; Partsch, 2010).

2.1.3 Improvement Potential through Grounded Theory Method

GTM is a qualitative research technique, which provides a method of breaking up and abstracting data (Charmaz, 2014; Corbin & Strauss, 1996). The approach is aimed at providing a theoretical explanation of the phenomena under study which is grounded in empirical data (Kelle, 2010). This is achieved through coding, which is the process of deriving and developing concepts from data. A code is a conceptual label assigned to a unit of data (Corbin, 2008). Through several coding steps, data is conceptualized and reconnected to show interrelations (Bazeley, 2013). This suggests that even if the domain is described from a process view in the data, the important structural aspects of the domain can be identified and related to one another to be visualized with a UML class diagram. During the process of coding, inconsistencies are revealed and analyzed, thus a range of perspectives and contexts can be integrated (Hoda, Noble, & Marshall, 2012). Information gaps can be identified early in the process and can be closed through theoretical sampling, which is the sampling of new data based on the analysis of previously collected data (Corbin & Strauss, 1996). This can guide decisions about which stakeholders to interview and which questions to ask. Through the development of concepts and categories, the required level of abstraction is reached iteratively with possibilities for adaptions and refinement (Corbin & Strauss, 1990; Halaweh, 2012b). During this process, the analyst must apply theoretical sensitivity, which is the ability to interpret and reflect upon data with the help of theoretical terms in order to identify significant data and assign it a meaning. Although theoretical sensitivity stems from the analyst's experience and expertise, it also develops during the research process and can be enhanced using techniques for questioning and systematically analyzing the data (Corbin & Strauss, 1996).

During the whole research process, the researcher should write memos to protocol his thoughts (Corbin & Strauss, 1996). Due to this constant documentation, interim analysis findings and the reasoning behind decisions regarding the research procedure are comprehensible. This way, traceability between the domain model and the original data can be ensured and changes and their effects on other requirements or the domain model can be incorporated (Hughes & Wood-Harper, 1999). Code memos are attached to codes and contain the result of the coding process, which are the conceptual labels and their descriptions (Corbin & Strauss, 1996). This property of the research method lends itself well to RE, where the definition of domain terminology is a common task and results in an important artefact, the glossary, where all entries are directly traceable to the domain model when created through our method.

2.1.4 Research Contribution

This thesis makes the following contributions:

- We demonstrate how practices from qualitative research can be applied to the task of creating a structural model of a domain within a RE context.
- Through the application of GTM to a domain analysis example, we identify required adaptions to the method and present a meta model for code systems, which formalizes otherwise implicit knowledge of the analyst, and provides guidance for efficient coding towards the goal of creating a domain model.
- We validate the domain analysis results using a survey of domain experts.

2.2 Related Work

Qualitative data analysis (QDA) and GTM have been recognized as being applicable to the field of RE, and related fields such as knowledge engineering and process modelling (Carvalho, Scott, & Jeffery, 2005; Chakraborty & Dehlinger, 2009; Pidgeon, Turner, & Blockley, 1991).

Carvalho et al. (2005) tested the application of GTM on descriptive process modelling by having two process models produced: one by an experienced software engineer and one by an experienced qualitative data analyst. They found that using GTM procedures cannot compensate for the software engineers expertise and experience. However, its application can improve the modelling process, because it forces the analyst to explore the complexity of the data and to systematically abstract from it. Similar findings are described by Pidgeon et al. (1991), who applied GTM to knowledge elicitation. They add that GTM secures the traceability of a derived model back to original data sources through the documentation of the analysis process in codes and memos, but point out that the produced model is still an interpretation which needs to be validated. Both authors criticize the complex and labor intensive analysis process of GTM. Their findings can be transferred to the process of domain analysis, which also includes eliciting knowledge from domain experts and analyzing it to derive an abstract model (Cossick, Byrd, & Zmud, 1992).

Hughes and Wood-Harper (1999) express the need for addressing the organizational context during requirements determination and demonstrate the use of GTM to develop an abstract account of the organization with two case studies. They adapt GTM by using pre-defined categories to address time constraints. The requirements determined in the case studies cover mostly organizational aspects, such as high-level goals, constraints and aspects of change as opposed to specific requirements or structural elements of an organization. In addition, they do not describe the data analysis process in their case studies.

Chakraborty and Dehlinger (2009) explain how the coding procedure of GTM can be applied to determine enterprise system requirements and to derive UML diagrams, thus bridging between qualitative data and final system descriptions. They demonstrate their approach by deriving a diagram from a textual high-level description of a university support system. However, the developed class diagram is not very consistent. Features and information about the implementation are represented as classes and the relationships between classes are not specified. An important adaption in their procedure is that they added conjectural categories to their model, which were not derived from the data but based on the experience of the analysts. They discovered during their study that, apart from the advantage of traceability, the iterative process of GTM allows the analyst to discover and close information gaps early in the process.

In a recent study, Chakraborty, Rosenkranz, and Dehlinger (2015) propose a Grounded and Linguistic-Based Requirements Analysis Procedure for eliciting non-functional requirements (NFR). They argue that the application of constant comparison and theoretical sensitivity in the analysis process improves the requirements specification by facilitating the sense making of multiple viewpoints into a cohesive description. However, Chakraborty et al. (2015) also point out that RE differs from traditional theory development applications of GTM, making adaptions to the method necessary. Also, because system analysts are not familiar with GTM, they propose to support the analyst in developing theoretical sensitivity and identifying the important concepts by giving him guidance about the theoretical principles to apply. For eliciting NFR, pre-defined categories of NFR were used, which were related using Mylopoulos, Chung, and Nixon's (1992) NFR framework. Thomas, Bandara, Price, and Nuseibeh (2014) also use an analytical framework, including pre-defined thematic codes and extraction rules, to apply QDA for the determination of privacy requirements for mobile applications. They state that QDA improves requirements elicitation by accounting for contextual factors and securing traceability.

The use of GTM to model requirements is also investigated in Halaweh's studies (2012a; 2012b). He states that categories and their relationships derived from Corbin and Strauss' coding paradigm (1996) can be compared to classes and their relationships in class diagrams. Thus, the informal model resulting from GTM can easily be translated into a formal model such as a UML class diagram. Theoretical sampling can help to identify users for interviewing and theoretical saturation can be used as an indicator to stop requirements elicitation. Halaweh (2012a; 2012b) argues that by applying GTM, thus letting requirements emerge from the data, requirements are user driven, supporting user-centered design and satisfying user needs effectively. He points out that the analyst needs to apply theoretical sensitivity in order to produce relevant results. Another finding of his studies is that GTM can assist in identifying non-technical aspects regarding change due to the system's development and implementation, for example user's resistance to change. This might help to initiate pro-active measures for implementation and training to overcome organizational problems. In a case study he conducted and analyzed interviews and retrieved a class diagram. However, although he states equivalents of GTM and object oriented analysis and design (OOAD) elements, he does not explain why and how these can be transferred and does not present guidelines for coding and transferring the informal model to the class diagram.

2.3 Research Question

A comprehensible description of analyzing a domain and developing a domain model in form of a UML class diagram using GTM is still lacking in previous research. This thesis therefore aims at developing a systematic procedure for domain analysis based on GTM. The research questions are:

- How can domain analysis be improved by using GTM?
- How does GTM have to be adapted for an application in domain analysis?

2.4 Research Approach

To find out how GTM can improve RE and how the method has to be adapted, we investigated traditional GTM in regards to its utilization in RE by applying it to an example. Initially, we analyzed high level workshops in an already running software project. However, as these workshops followed a top-down approach, we could not sufficiently apply procedures of GTM. Therefore, we moved on to conducting independent interviews with domain experts and applying GTM to derive a domain model in order to analyze differences between traditional GTM and its application in domain analysis. These observations were incorporated into a code system meta model, which guides the analysis of a domain based on GTM.

2.4.1 Used Data Sources

For our domain analysis example, we used interviews as data sources, which is the most commonly used requirements elicitation technique (Browne & Rogich, 2001; Partsch, 2010; Rupp & SOPHISTen, 2014). We conducted six semi-structured interviews with four domain experts from different companies. All domain experts had high level management positions in HR and experience in HR development. The companies varied in size from a local company with 50 employees to an international corporation with over 100,000 employees worldwide and operated in the sectors IT and market research. The interview length varied between 15 and 60 minutes.

The first interview was guided by 12 open questions, which aimed at gaining an overview over the domain as shown in appendix A. For the following interviews, analysis results determined the interview questions according to the principle of theoretical sampling. As we conducted semi-structured interviews, the prepared questions were used as a guideline and we adjusted to participant's answers (Corbin & Strauss, 1996; Myers & Newman, 2007). This was important because we wanted to capture the knowledge of the domain experts and not force preconceptions on the data (Corbin & Strauss, 1996; Nohl, 2013). To clarify inconsistencies, close information gaps and extract more detailed information, we conducted follow-up interviews with two of the domain experts.

As a secondary data source, literature on HR development (Achouri, 2015; Becker, 2013; Ryschka, Solga, & Mattenklott, 2011; Thom & Zaugg, 2008) was used to clarify the definitions of terms. Although literature research prior to or at the beginning of the research project is avoided in GTM, Corbin and Strauss believe that literature should be used to support the analysis as soon as the main categories of the theory have emerged (Gibson & Hartman, 2014).

2.4.2 Transcription

The interviews were audio recorded and then anonymized and transcribed manually. Corbin and Strauss (1996) advise to transcribe interviews fully at the beginning of the research project and in later stages only to transcribe those parts of an interview, which are important for the theory. To limit the risk of missing useful information due to our lack of experience in GTM, we transcribed the whole content, but left out introductory and closing conversations and defined a simplified transcription system (King & Horrocks, 2010). The speech parts of interviewees were transcribed word for word, including laughter. However, we did not include details such as accentuation or the lengths of breaks, because they are not relevant for the purpose of our research (Bazeley, 2013). For the speech parts of the interviewer, we left out parts which did not include any information, such as expressions of comprehension, because this would interrupt the information given by interviewees unnecessarily.

2.4.3 Coding

Coding involves breaking up the data, conceptualizing it and reassembling it in a new and more abstract way (Charmaz, 2014; Corbin & Strauss, 1996). Corbin and Strauss (1996) divide the process of coding into three steps, which are conducted alternately in an iterative process. Open coding is the process of identifying and comparing concepts shown in the data and grouping those concepts into categories. The developed concepts and categories are then set into relation during axial coding. In the final stage, selective coding, the categories are integrated around a core category to form a theory. We applied this technique to our independent domain analysis to evaluate its utilization in a RE context. The data analysis was performed in MAXQDA, a QDA software[1].

2.4.3.1 Open Coding

During open coding, incidents indicating a phenomenon are identified through looking at units of data which seem appropriate in size, ranging from one word to a whole paragraph or document. These units of data are coded (i.e. labelled) with a concept. A concept is a conceptual description of the event, the action/interaction, or the conceptual idea behind an incident (Bazeley, 2013; Corbin & Strauss, 1990). A concept can be a term used by the participant of the study, a so called in vivo code, or can be determined by the analyst (Charmaz, 2014). Similar incidents are labelled with the same concept. Concepts can then be grouped into categories, which are more abstract concepts representing a phenomenon central to the research. For these categories, properties are identified through comparing them in regards to their similarities and differences (Corbin & Strauss, 1990). The data units indicating the concept are assigned a code, thus become units of coding and can be further described in a code memo.

[1] http://www.maxqda.com/

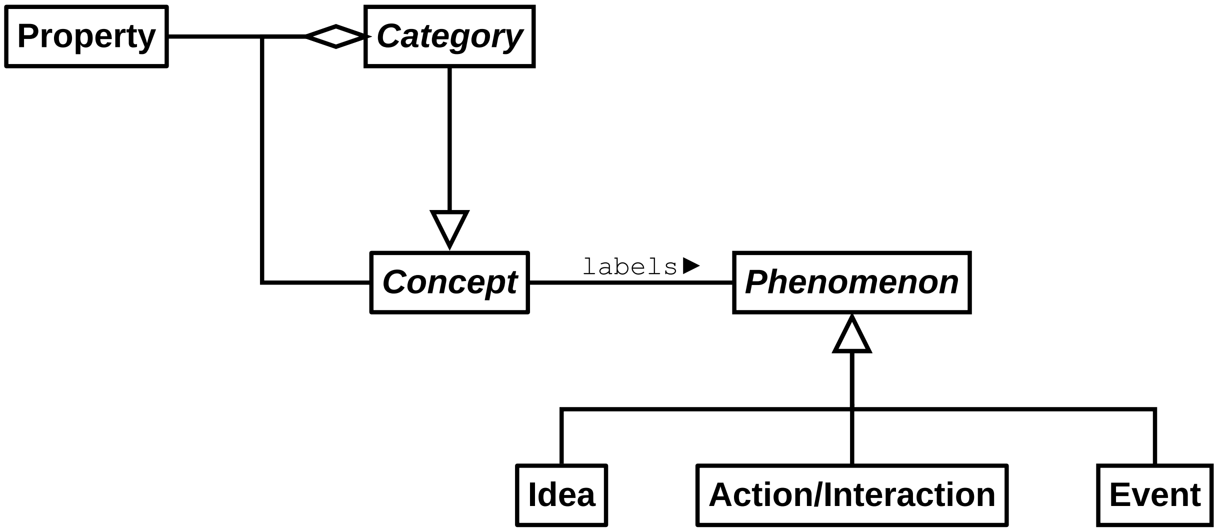


Figure 1: Conceptualization in GTM

When we applied GTM to our example, concepts emerged from the data during open coding as explained above. The coding process started after the first interview had been conducted and transcribed. Happenings or other aspects mentioned in the descriptions of domain experts were coded. In order to represent the domain terminology, mainly in vivo codes were used (Bazeley, 2013; Larman, 2010). Units of coding varied in size from one phrase to a whole paragraph. Coding a whole paragraph was sometimes necessary to preserve information about the relationships between concepts. The units of coding belonging to one concept were compared to investigate their differences and similarities and to guide the questions for the following interviews.

Usually, actors are not coded explicitly in GTM research projects, because they are intertwined with other concepts. For example, a study investigating how patients deal with pain includes concepts such as "experiencing pain" or "pain", but no concept "patient" (Charmaz, 2014; Corbin & Strauss, 1996). However, actors, including external systems and organizational units, need to be represented in a domain model (Larman, 2010; Rosenberg & Stephens, 2007; Wazlawick, 2014). For the domain of HR development, for example, "employee" is a central entity. The same is the case for objects and places, which are normally not investigated explicitly during GTM research. Therefore, actors, places and objects, which includes tangible and intangible objects and the concept type "idea" of GTM, need to be coded as well.

Because domain models represent the entities of a domain, these are the phenomena we want to study and were therefore developed into categories. Concepts which seemed to belong to the same aspect were grouped into categories. For example "giving feedback", "feedback survey", "360-degree feedback" and "evaluating feedback" were grouped under "feedback".

During our analysis, we also coded background information, such as the position of the interviewee and the current systems in use, and information about the purpose of HR development. Although these codes should be clearly distinguished, such information should be captured and kept in mind during the analysis, as it might be the reason for differences between incidents and contain important information for later design decisions.

2.4.3.2 Axial Coding

During axial coding, a higher abstraction level is reached through comparing categories, identifying their properties and grouping categories which seem similar into a more abstract category (Corbin & Strauss, 1996). Thus, a hierarchical structure of categories and subcategories emerges in which subcategories represent specializations of a category. For example, target agreements which were set for an employee at the beginning of a year were an aspect which often recurred in the data. When we compared the data fragments in which targets were mentioned, we discovered that there were two different kinds of targets: performance targets which were quantitative targets and development targets which were qualitative targets to improve the employee's competencies. Therefore, "performance target agreement" and "development target agreement" became subcategories of the category "target agreement" and were related to it with an "is a" relationship.

Subcategories and concepts can also describe a category further, for example by defining the conditions leading to a phenomenon. To describe a category fully, Corbin and Strauss (1996) propose the following coding paradigm.

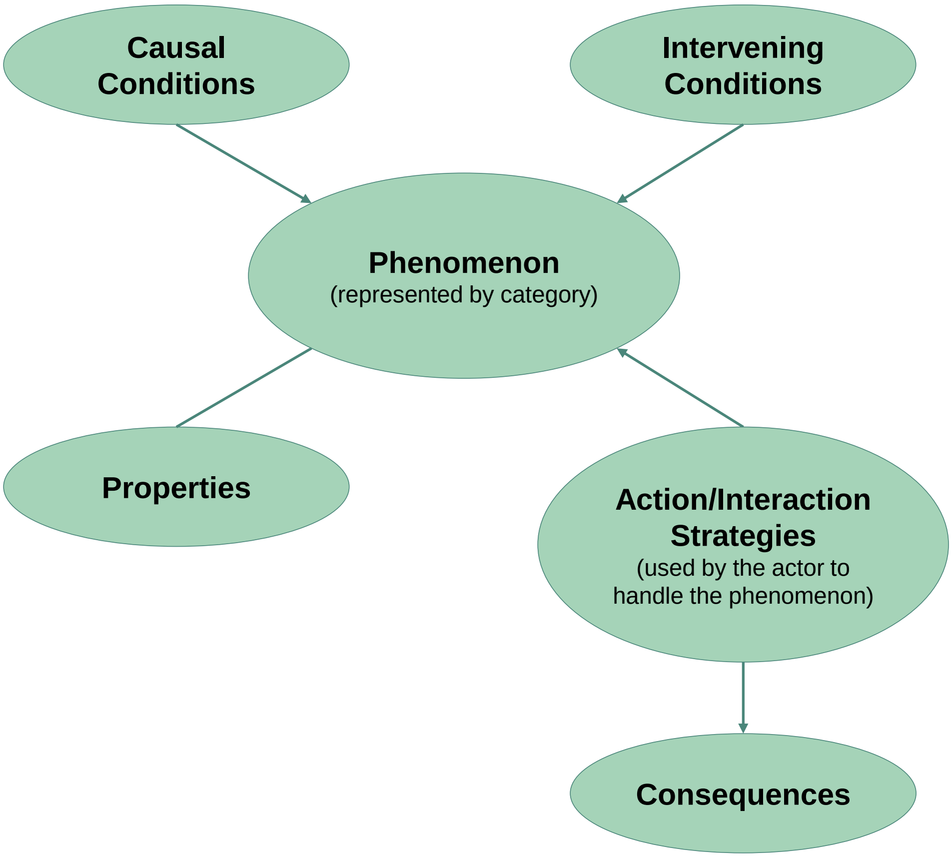


Figure 2: Coding Paradigm

Using the coding paradigm leads to the following relationship types between concepts. The relationship type “affects” includes both conditions influencing a phenomenon and action and interactions strategies being directed at a phenomenon.

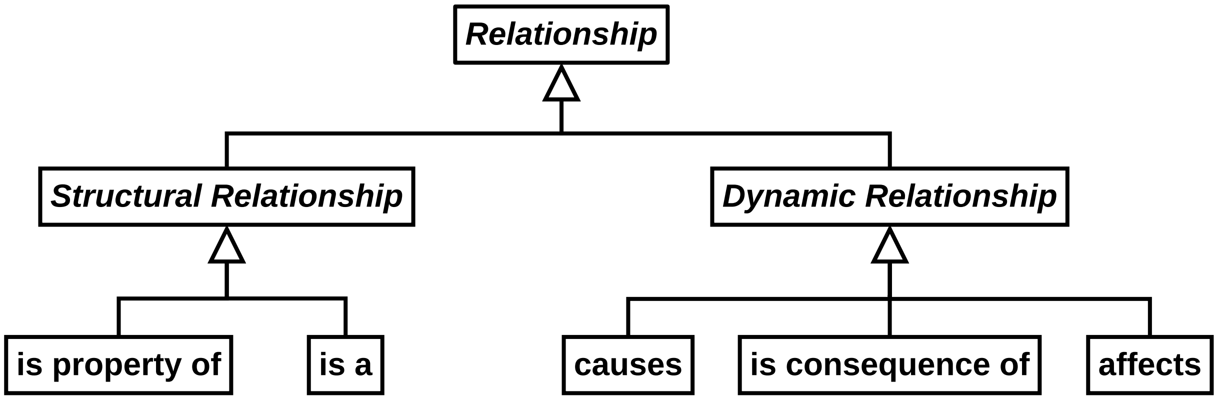


Figure 3: Relationship Types in GTM

According to Corbin and Strauss (1996), the analyst records his thoughts about relationships between concepts in memos. In order to derive a domain model, we needed to document the relationships explicitly. Activities, including actions and interactions, are performed by an actor and are directed at a phenomenon, thus affect a category. Causal or intervening conditions can be any phenomenon, for example an event. The fact that they represent a condition for something is shown through the relationship “causes” or “affects”. In our interviews, the domain experts also mentioned influencing aspects, for example the current market situation, which influences the company targets and the competencies required of employees. Such aspects describe a state, which affects another concept.

As the focus of domain analysis lies on investigating the structure of a domain, additional structural types of relationships apart from “is a” and “is property of” are needed (Daoust, 2012). Glaser (1978) proposes a more flexible method for relating different concepts of a theory by using theoretical codes, which he divides into coding families. Charmaz (2014) criticizes that Glaser does not provide a comprehensive model and that some of the theoretical codes overlap and seem random. Their use is therefore difficult for a requirements engineer who is a novice at GTM (Chakraborty et al., 2015; Charmaz, 2014). However, we investigated Glaser’s coding families (1978) and found two theoretical codes which are relevant for deriving a structural description of a domain. These codes are “part” from the dimension family and “type” from the type family. A “type”, which has an “is a” relationship, already follows from developing a hierarchy of categories, but “is part of” is an important relationship for describing the structure of a domain. We found that the remaining structural relationships cannot be grouped, because this would limit the meaningfulness of a model. For this reason, structural relationships which cannot be allocated to one of the other relationship types are of the type “is related to” and can be specified with a specific relationship name.

2.4.3.3 Selective Coding

During the selective coding phase, one or a few core categories are selected for representing the central idea of the theory. In this phase, categories which are not well developed need to be completed, which means going back to open and axial coding and sampling new data. This is done until theoretical saturation is reached, which means that new data does not bring any important new information regarding the categories (Corbin & Strauss, 1996; Hughes & Wood-Harper, 1999). Collecting a single core category did not make sense for the purpose of domain analysis, as a domain model should give a complete representation of the domain (Bolloju & Leung, 2006). The phenomena, i.e. entities, which are central to the domain have already been identified as being important by developing them into categories.

2.4.4 Domain Modelling

The domain analysis results are visualized with a domain model in form of a UML class diagram. To facilitate communication with stakeholders, the model should represent the domain in an easily understandable way. Therefore, only basic UML notation elements should be used (Kecher, 2011; Rupp & SOPHISTen, 2014). For deriving a domain model from the code system, the relevant concepts and relationships had to be identified and transferred according to the domain meta model shown in figure 4.

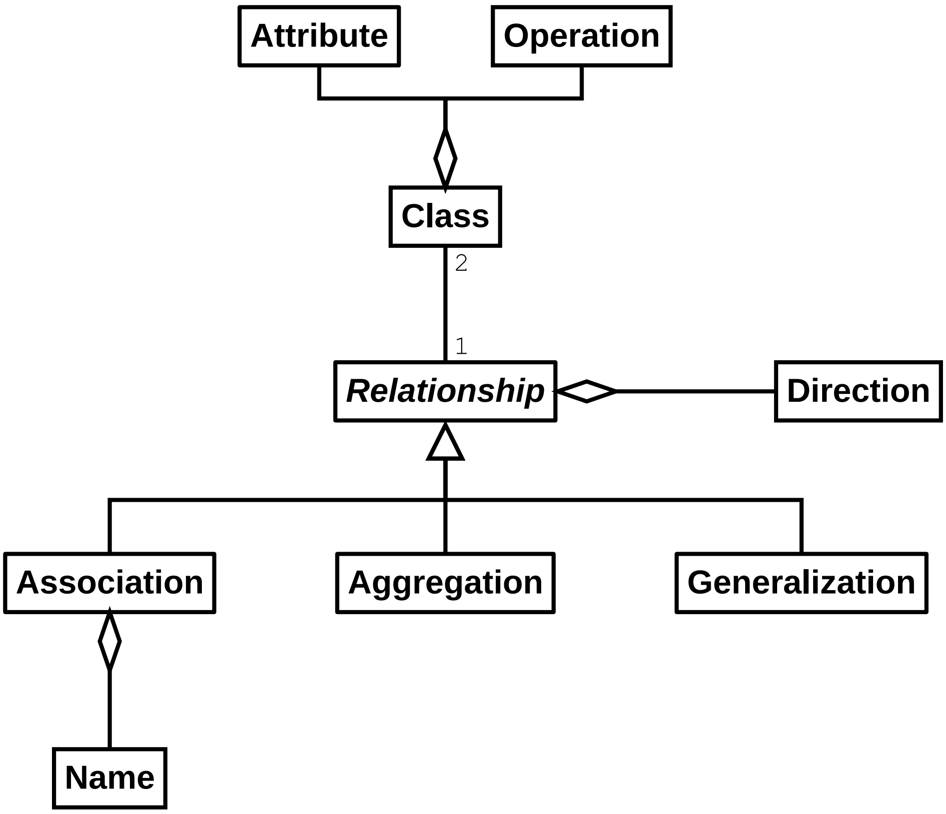


Figure 4: Domain Meta Model

Normally, operations are not included in domain models, but represented in dynamic models such as use case diagrams (Larman, 2010; Rosenberg & Scott, 2001; Rupp & Queins, 2012; Wazlawick, 2014). However, as our analysis method provides both structural and dynamic information about the domain, operations can also be extracted from the code system if the analyst wants to include them in the domain model. For example, many activities affected the category “development measure”, such as “proposing development measure”.

2.5 Research Results

2.5.1 Code System Meta Model

During our case study, some important differences between traditional GTM and its application to domain analysis became apparent. The most significant difference is the focus on either dynamics or structure. GTM focuses on interaction systems and therefore mainly uses concepts to describe dynamic aspects of a research area, i.e. happenings and actions which indicate a phenomenon (Corbin & Strauss, 1990). A domain model however describes the important entities of a problem domain and their structural relationships (Broy, 2013). Although the data sources used for domain analysis, such as domain expert knowledge, contain mostly dynamic descriptions of the domain, an analysis method must provide a way to extract structural entities (Rupp & Queins, 2012; Wazlawick, 2014). These structural aspects need to be described with their attributes and related to each other, the same way phenomena under study in GTM are represented with categories and properties. We therefore adapted the coding procedure and developed a code system meta model. By coding transcripts of interviews with domain experts according to our meta model, structural and dynamic aspects of the domain and their relationships are identified. From the developed code system, RE artefacts such as a domain model can be derived.

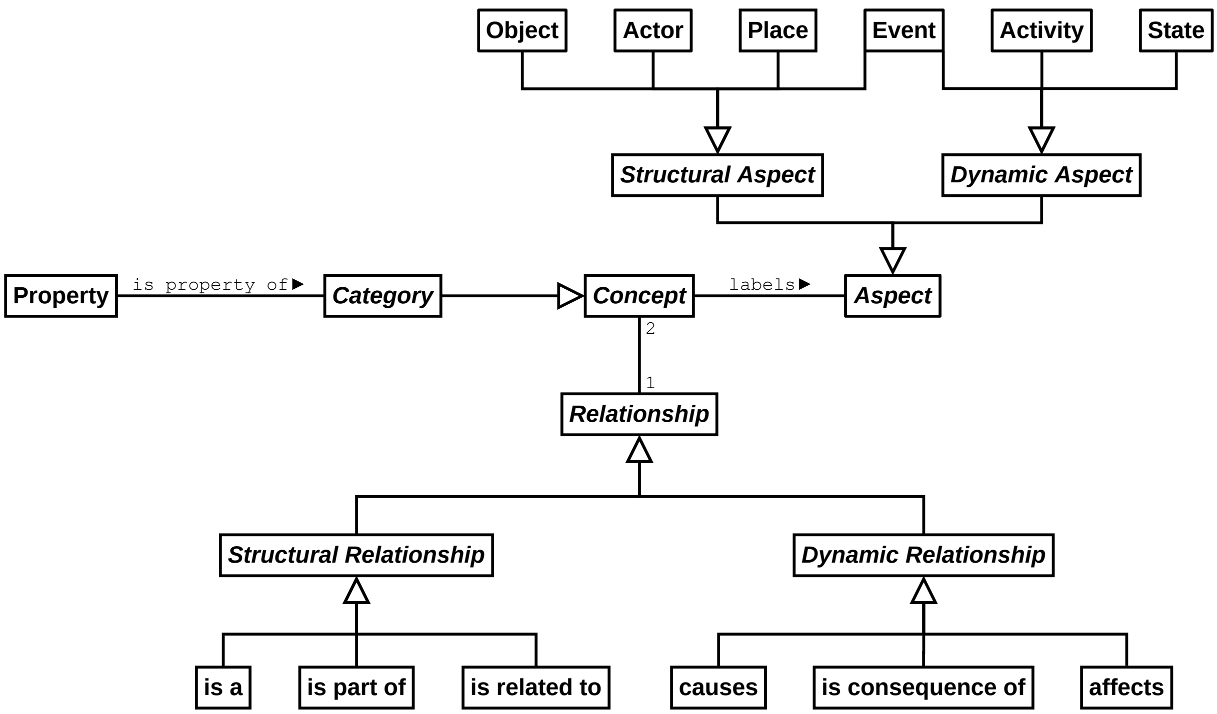


Figure 5: Code System Meta Model

During open coding, data fragments which represent structural and dynamic aspects of the domain are coded. As a domain model should represent the domain terminology (Larman, 2010), mainly in vivo codes should be used. If there are several terms used for the same concept, the synonyms should be documented in the code memos as shown in chapter 2.5.1.1. We advise to first generate specific concepts for smaller units of coding and then to combine them during the abstraction process. It is important to make all aspects explicit. A code “employee attends development measure” is not correct, because it includes several aspects: the actor “employee”, the activity “attending development measure“ and the event “development measure”. In addition, the analyst should be careful to describe activities with verbs and not with nouns in order to distinguish them from events.

Concepts are then grouped into categories, which represent the aspects central to the domain and are described further in regards to their properties and context through constant comparison and questioning. The data fragments indicating the properties should also be coded. Both structural and dynamic aspects can be developed into categories. However, if the purpose of the analysis is clear, such as the extraction of a domain model, the analyst should focus on aspects which are central for the analysis and investigate these first.

Concepts which are grouped into a category can have a structural or dynamic relationship with this category, which needs to be defined during axial coding, as do relationships between categories. The relationship type “affects” includes the following cases:

- An actor performs an activity.
- An activity is directed at a category.
- A state influences a concept.

As there is no feature available in QDA software for recording relationships other than a hierarchical structure, and coding them explicitly would overload the code system, they are documented in code memos. In the third stage of GTM, selective coding, categories which are not fully developed are completed and refined.

During the whole coding process, questions need to be asked about the concepts and categories which identify their properties and relationships. These questions may lead to further investigations through theoretical sampling. If a concept or category only appears once in the data or cannot be related to other categories, it might be irrelevant for the domain, but needs to be further investigated by the analyst before being discharged (Pidgeon et al., 1991).

2.5.1.1 Code Memo

To provide a thorough understanding of the concepts and categories of a domain, they need to be described in a glossary (Rupp & SOPHISTen, 2014). By documenting this information in code memos as shown in figure 6, term descriptions are directly linked to the domain model elements, but can also be exported as a separate glossary.

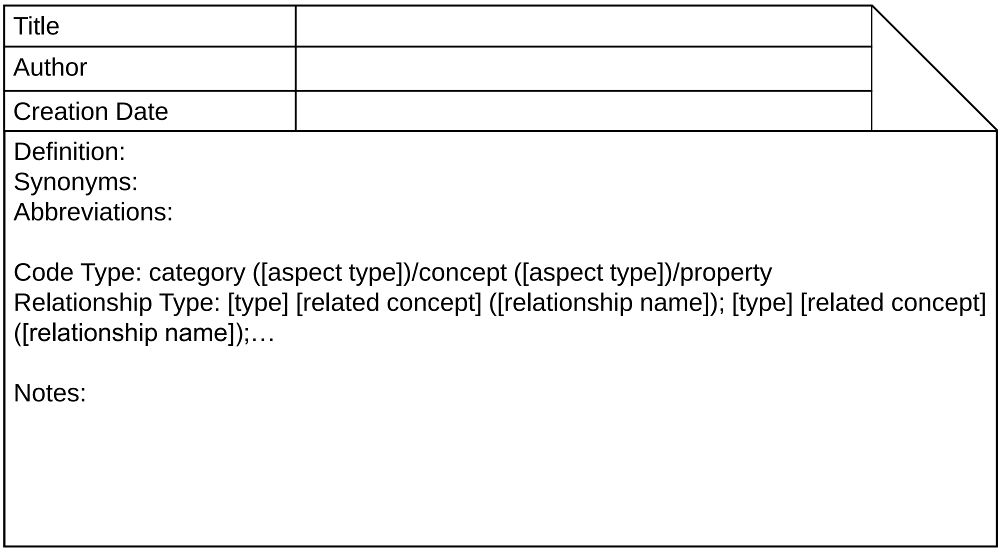


Figure 6: Code Memo Stencil

2.5.2 Transferring the Code System to a Domain Model

The following table shows which elements of the code system can be transferred to a domain model.

Table 1: Code System Meta Model Elements transferred to Domain Meta Model Elements

Code System Meta Model	**Domain Meta Model**
Structural category	Class
Property	Attribute
Structural concept	n/a
Dynamic category	n/a
Activity	Operation candidate/association candidate
State	n/a
Is property of	Is attribute of
Is a	Generalization
Is part of	Aggregation
Is related to	Association
Causes	n/a
Is consequence of	n/a
Affects	Indicates operation/indicates association

Activities and “affects” relationships are candidates for or indicate operations and associations, but need to be investigated by the analyst, as shown in table 2.

Table 2: Representation of Activities and "affects" Relationships in a Domain Model

Code System	**Domain Model**
Activity affects category	Activity is operation of class
Actor affects activity, which affects category	Activity is represented as an association between actor class and category class

After transferring the code system to a domain model, the analyst needs to review the domain model and adjust it to his needs. He may decide on a higher abstraction level for communicating with stakeholders, for example by leaving out attributes and operations or by reducing the number of associations shown (Larman, 2010).

2.5.3 Observations

During our study, we found that the coding procedure supported the structuring and analysis of qualitative data. Important concepts became apparent already early in the coding process. Our hypothesis that structural elements and relationships can be extracted from a process description has also been confirmed. The participating domain experts primarily gave an account of their domain from a process point of view. Through the development of concepts and categories, the structural aspects emerged and could be further investigated through theoretical sampling. Inconsistencies could be investigated through comparing the respective data fragments and notes could be taken in code memos about questions which need to be asked in the next interview and about the different options of interpretation. This was especially important for integrating company-specific descriptions of HR development into a consistent domain model.

Although the systematic coding procedure and the writing of memos make the process of domain analysis traceable, coding and modelling decisions are still interpretive and therefore depend on the analyst’s experience and expertise. What to code and how to develop concepts into categories is a difficult task for which there is not one solution. We found that abstracting too early in the process or focusing too much on the domain model while coding can make later changes more difficult. However, the analysis method provides more guidance to a novice analyst for extracting a domain model than the method described in the introduction. Systematic coding helps the analyst to engage with the domain to be analyzed. The application of constant comparison, theoretical sensitivity and questioning of the data can also help to prevent experienced analysts from prejudiced misconceptions. On the other hand we experienced the coding process as very time consuming and requiring a high cognitive effort, like many authors of related work also state.

2.6 Results Discussion

2.6.1 Limitations

In our example, theoretical sampling could not be fully applied due to limited access to interview partners. This meant that we applied theoretical sampling mainly to the choice of questions and not the choice of interview partners. Because we interviewed domain experts from different companies, we had to start each interview with basic questions to understand how HR development was conducted in their company. Thus, the interviews provided rather high level information. We conducted follow-up interviews with two domain experts to retrieve more detailed information. However, theoretical saturation could not be reached due to availability constraints of domain experts.

2.6.2 Validation of Results

To validate our proposed method, it will have to be applied to other examples. These examples should also attempt to derive RE artefacts which address dynamic aspects of a domain in order to ensure that the code system meta model allows a holistic analysis.

The domain model created through our method was evaluated in regard to the following quality aspects proposed by Bolloju and Leung (2006):

- Syntactic quality: The domain model adheres to the modelling language.
- Semantic quality: The domain model represents the reality correctly and completely.
- Pragmatic quality: The domain model is easy to understand from the stakeholders' perspective.

We used basic notation elements of UML class diagrams in accordance with the UML specification of the Object Management Group, Inc. (2013). Adherence to the syntax was ensured by using tool support for domain modelling. To assess the perceived semantic and pragmatic quality, we conducted a qualitative survey of the participating domain experts. The evaluation of semantic quality was completed by comparing our domain model with an existing knowledge representation of the domain to assess the congruence of identified concepts with established research.

2.6.2.1 Survey of Domain Experts

For our written survey (adapted from Poels, Maes, Gailly, & Paemeleire, 2005), we received answers from three of the four participating domain experts as shown in table 3.

Table 3: Evaluation of Domain Model by Domain Experts

Question	Disagree	Rather disagree	Undecided	Rather agree	Agree
It was easy for me to understand what the model was trying to model.			1	1	1
The model represents the domain correctly.				3	
The model is a realistic representation of the domain.			1	2	
All the elements in the model are relevant for the representation of the domain.			2	1	
The model gives a complete representation of the domain.				3	
The model contains contradicting elements.		2	1		
The model contains the following inconsistencies.	"Performance assessment does not necessarily evaluate target agreements"				
The following elements are missing from the domain model.	"Criteria of potential"				

In general we received positive feedback. The domain model was evaluated to give a rather complete, realistic and correct representation of the domain. The only concept which was identified as missing was "criteria of potential". Within the interviews we conducted, the topic of potential was only mentioned once as being currently in discussion for implementation, thus did not show to be relevant according to the data. However, as saturation could not be reached, this concept might appear during further analysis. The only inconsistency which was reported, was that performance assessment did not necessarily evaluate target agreements. Domain experts' descriptions of the relationship between competency, performance, employee assessment and target agreements were inconsistent and imprecise. Their statements were therefore compared and further investigated in interviews, which resulted in the distinction between competency and performance assessment and the defined relationship between performance evaluation and target agreements. However, the inconsistencies and imprecisions in the data were not completely resolved because saturation could not be reached and would need to be investigated further with additional interviews. The received feedback suggests that regular validation of analysis results should be part of the domain analysis process to improve the quality of the domain model.

The domain experts were undecided if all elements in the domain model were relevant for the representation of the domain. This was to be expected as the evaluation of relevance depends on the purpose of the domain model and the desired level of abstraction.

Answers in regards to the perceived pragmatic quality varied. The domain model was perceived as confusing by some of the domain experts. This might be attributed to our limited experience in domain modelling and presents an opportunity for improvement in regard to the design of the domain model. In addition, it should be investigated if clearly defined abstraction levels in the code system can help to improve the clarity of the domain model.

2.6.2.2 Comparison with existing Knowledge Representation of the Domain

To assess the congruence of identified concepts, we compared our domain model with Schmidt and Kunzmann's (2006) competency-based ontology of HR development. While the ontology only covers HR development in regard to competency management, all participating domain experts stated that performance management was also a part of HR development and our analysis showed a close interrelationship between these two sub-domains as shown in appendix E. Thus, our domain model provides a more holistic representation of the domain. In comparison, our domain model covers 70% of the concepts from the ontology, while 50% of the competency-related classes (excluding sub-classes) from our domain model are represented in the ontology. However, the identification of equivalent concepts was based on our interpretation, because Schmidt and Kunzmann (2006) do not provide definitions of their concepts. This shows the value of creating a glossary to provide a thorough understanding of the identified concepts. Using our method, concepts and their definitions are developed simultaneously and directly linked, which ensures consistency between the domain model and the glossary.

2.7 Conclusions

In this thesis we applied GTM to a domain analysis example and identified similarities and differences as well as weaknesses of using this QDA technique in a RE context. Based on these observations we proposed adaptions, which led to a code system meta model for domain analysis based on GTM and a procedure for deriving a domain model.

We showed that by applying our method to domain analysis, structural elements and relationships needed to derive a UML class diagram can be extracted from interviews with domain experts. Constant comparison and theoretical sampling assist well in integrating differing domain descriptions into an abstract model. While the analysis process still includes interpretations and modelling decisions, our method provides more guidance than existing domain analysis approaches and a thorough documentation of these decisions. In addition, codes and memos ensure traceability between the original data and the derived model and assist in connecting several RE artefacts.

Although our method will have to be validated through further research projects, we are convinced that its application can improve the process as well as the outcome of domain analysis and RE, thus contribute to the success of software development projects.

3 Elaboration of Research Chapter

3.1 Requirements Engineering

Requirements describe properties of a software system that provide value to a stakeholder, for example by helping a user to solve a problem (Wiegers, 2003). They are the basis for design, implementation and testing of a software system and are therefore critical for the success of a software development project (Berntsson-Svensson & Aurum, 2006; Hruschka, 2014; McManus & Wood-Harper, 2007). Schmidt, Lyytinen, Keil, and Cule (2001) identified „misunderstanding the requirements“ as one of the top three risk factors faced by software project managers. A US survey (The Standish Group International, Inc., 1995) found poor requirements to be one of the main problem sources in software development projects and the results of a European study (Ibáñez, Kugler, & Rementeria, 1996) also showed that software organizations perceive RE as the greatest problem area in software projects. According to Hamill and Goseva-Popstojanova (2009), requirement faults account for 33% of system failures. Errors made during requirements activities increase development costs because they make rework necessary (Leffingwell, 1997). The later these errors are discovered in the software development process, the higher are the costs for correcting them (Boehm, 1981; Grady, 1999). Therefore, new approaches for requirements activities are of high relevance.

3.1.1 Requirements Activities

As shown in figure 7, the terminology describing requirements activities is inconsistent. Most commonly, they are summarized under the topic of RE (Balzert, 2009; Ebert, 2012; Pohl & Rupp, 2011). During RE, information needs to be elicited from stakeholders and other sources. This information has to be analyzed and documented in order to specify requirements, which then have to be validated to ensure the necessary level of quality. The defined requirements have to be managed during RE and throughout the remaining software development process (Pohl & Rupp, 2011). This is important because requirements are constantly changing and traceability needs to be ensured (Pohl & Rupp, 2011; Wiegers, 2003).

The RE activities are not clearly separable but form an interwoven and iterative process, in which requirements are determined and revised (Wiegers, 2003). The first set of activities is referred to as requirements development (Wiegers, 2003), requirements determination (Browne & Rogich, 2001; Pitts & Browne, 2007), requirements discovery (Robertson & Robertson, 2006), or requirements analysis (Hruschka, 2014). Although the various definitions of all these terms are not congruent, they identify the initial processes within RE as identifying the stakeholders and their needs, thus the purpose of the envisioned software, and documenting them as a basis for communication, design, implementation, and testing (Easterbrook & Nuseibeh, 2000; Robertson & Robertson, 2006).

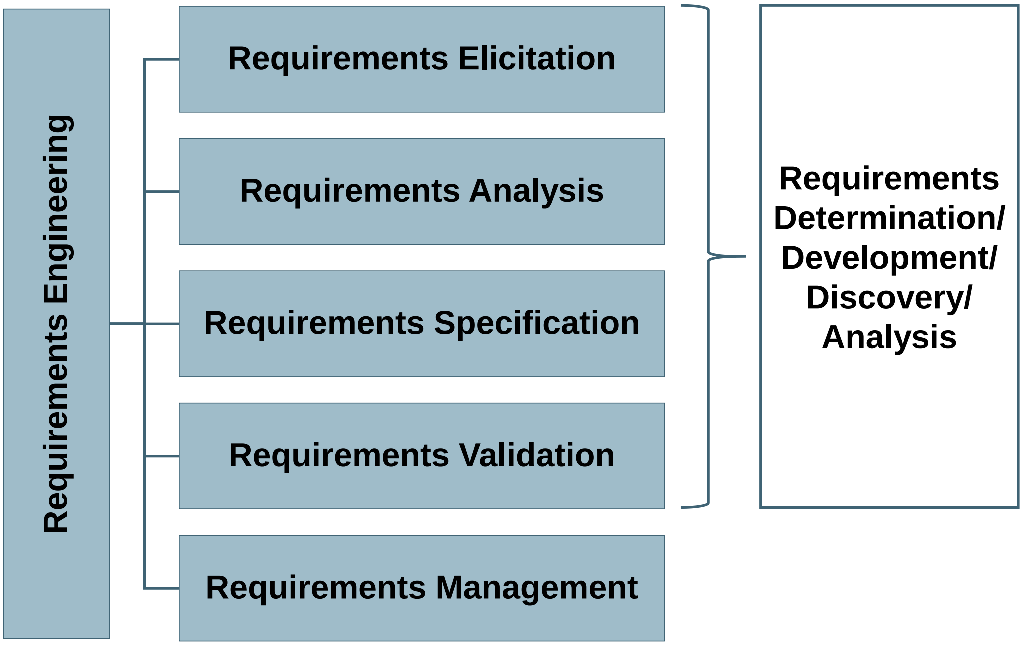


Figure 7: Requirements Activities

Requirements elicitation includes stakeholder analysis, which is the process of identifying stakeholders and other information sources. Stakeholders are people or groups of people who directly or indirectly influence the requirements of a system, such as users, developers, customers or testers (Pohl & Rupp, 2011; Sharp, Finkelstein, & Galal, 1999). Further it encompasses the identification of their needs as well as gathering information about the context, goals, and constraints of a planned software system (Easterbrook & Nuseibeh, 2000). There are many methods available for collecting data from stakeholders, such as interviews, observations, or card sorting, but interviews are the most commonly used technique (Dieste, Juristo, & Shull, 2008; Partsch, 2010; Rupp & SOPHISTen, 2014).

The elicited data needs to be thoroughly analyzed to document and specify requirements. During analysis, the requirements engineer needs to uncover the business problem, which is supposed to be solved with a software system (Robertson & Robertson, 2006). He therefore needs to understand what stakeholders mean and interpret the data accordingly (Robertson & Robertson, 2006; Rupp & SOPHISTen, 2014). As the information gathered is often inconsistent and incomplete, the requirements engineer needs to find and fill information gaps and abstract the information to develop a consistent requirements specification (Balzert, 2009; Rupp & SOPHISTen, 2014).

Because requirements are the basis for communicating, designing, implementing as well as testing the desired software system, the defined requirements and all changes in requirements need to be documented neatly to secure traceability. Requirements traceability is “the ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases)” (Gotel & Finkelstein, 1996, p. 167). Pre-requirements traceability thereby focuses on requirements development and specification, documenting how stakeholders’ needs were discovered and how these needs were integrated in the requirements specification (Gotel & Finkelstein, 1996). Documentation is necessary, because undocumented knowledge diffuses over time, for example because of personnel fluctuations, and changes in requirements need to be incorporated. Requirement engineers also need to document and visualize their findings in order to keep an overview and to communicate with stakeholders without misunderstandings. In addition, clear and comprehensible documentation of the domain and the requirements offers the opportunity for reusing this knowledge in later projects, thus preventing redundant work (Broy, 2013; Easterbrook & Nuseibeh, 2000; Rupp & SOPHISTen, 2014). Requirements can be documented in written form, but can also be visualized or completed with models. Models represent “a view of a system”, thus “an abstraction of the system, with a certain purpose” (Rupp & SOPHISTen, 2014). They play an important role in RE in order to organize information, uncover information gaps and inconsistencies and to facilitate communication with stakeholders (Hickey & Davis, 2003).

The final specification of requirements consists at least of a formal requirements specification document (or user stories in agile software development), in which requirements are documented according to a requirements template, and a glossary with a definition of all used terms. However, clear documentation is important throughout the whole RE process (Balzert, 2009).

3.1.2 Domain Model

One important artefact of RE is the domain model, which describes the important entities of the domain in which the envisioned system is supposed to operate and their structural relationships. This model is created in the early stages of RE and is used to represent the problem domain without considering possible technical realizations (Broy, 2013). Usually, the basic notation elements of a UML class diagram are used. Classes represent a set of objects which have the same attributes and operations (Podeswa, 2010). Attributes describe the information the business needs to capture about them and operations define the actions performed on them (Daoust, 2012). Classes are related to each other through an association, aggregation, or generalization. An association is a relationship between classes, which needs to be further defined with a relationship name and the direction of the relationship (Seidl, Brandsteidl, Huemer, & Kappel, 2012). An aggregation is a "part of" relationship between a component and the aggregate. A generalization is a relationship between a more specific and a more general class. An object which belongs to the specific class also belongs to the general class and inherits all features of the general class (Podeswa, 2010). For a comprehensible description of the domain, some authors advise against using operations (Rupp & SOPHISTen, 2014; Wazlawick, 2014) or against distinguishing aggregation or composition relationships and instead recommend to name the association accordingly (Daoust, 2012). The terms visualized in a domain model need to be described further in a glossary, including the definition, abbreviation and synonyms of each term (Rupp & SOPHISTen, 2014).

3.2 Grounded Theory Method

Qualitative research involves analysis of data with non-mathematical techniques. This data can be quantitative or qualitative in itself, but the most common type of data are interview transcripts. During research, analytical or interpretive techniques are used in order to gain insights. (Corbin & Strauss, 1996)

One well established method of qualitative research is GTM, which was developed by Glaser and Strauss (1999) in need of a new method to construct theories in social science (Bryant & Charmaz, 2010b). The motivation for creating this method was to provide strategies for qualitative analysis for producing reliable and valid results with equal significance as statistical-quantitative methods (Bryant & Charmaz, 2010a). Rather than starting with a hypothesis and conducting research to test it (called hypothetico-deductive approach), the approach is aimed at providing a theoretical explanation of the phenomena under study which is grounded in empirical data using a systematic, inductive and comparative method (Kelle, 2010). It includes the conceptualization of data, called coding, non-statistical theoretical sampling, memo writing and the development of theories about conceptual relationships, which are often visualized in diagrams (Corbin & Strauss, 1996).

3.2.1 Different Approaches

There are numerous methods which have been adapted according to the field of research or the interpretations of researchers and therefore represent variations of GTM. However, two main approaches can be distinguished, following the different paths the original founders took after their initial collaboration.

Glaser (1978) focuses on the interpretive and inductive aspect of GTM, in which the researcher approaches his field without any specific research questions in mind and without consulting related literature prior to analysis (Titscher, 1998). This should prevent the analyst from forcing preconceived ideas on the data instead of letting categories emerge during the analysis (Hoda et al., 2012). Glaser differentiates between substantive and theoretical codes. Substantive codes refer to the empirical substance of the research domain and are developed during open coding. Theoretical codes are terms which describe possible relations between substantive codes, for example causes or consequences (Glaser, 1978; Kelle, 2010). For this purpose Glaser provides a list of coding families, which can be used to integrate the theory. An excerpt from this list is given below.

- The Six C's: causes, contexts, contingencies, consequences, covariances and conditions
- Causal: sources, reasons, explanations, accountings or anticipated consequences
- Process: stages, staging, phases, phasings, progressions, passages, gradations, transitions, steps, ranks, careers, orderings, trajectories, chains, sequencings, temporaling, shaping and cycling
- The Degree Family: limit, range, intensity, extent, amount, polarity, extreme, boundary, rank, grades, continuum, probability, possibility, level, cutting points, critical juncture, statistical average (mean, medium, mode), deviation, standard deviation, exemplar, modicum, full, partial, almost, half and so forth
- The Dimension Family: dimensions, elements, division, piece of, properties of, facet, slice, sector, portion, segment, part, aspect, section
- Type Family: type, form, kinds, styles, classes, genre
- The Strategy Family: strategies, tactics, mechanisms, managed, way, manipulation, maneuverings, dealing with, handling, techniques, ploys, means, goals, arrangements, dominating, positioning

These coding families overlap and are not complete, and an analyst may think of many additional theoretical codes. Several families might fit the same data, thus the choice of theoretical codes depends on the research focus and needs to be grounded in the data. The purpose of the coding families is to show possibilities of integrating substantive codes and to increase the analyst's theoretical sensitivity. (Glaser, 1978)

Corbin and Strauss (1996) present more guidance in their approach in regard to the research process and provide criteria for evaluating a grounded theory. They believe that an open research question and a critical literature research do not conflict with the goal of constructing a theory which is grounded in empirical data. For relating categories, they propose a coding paradigm. The paradigm includes some theoretical terms which are also included in Glaser's coding families, but connects them in a general model of action to determine the interaction strategies of actors.

It identifies the following:

- The investigated phenomenon
- Causal conditions
- Attributes of the context of the investigated phenomenon (properties)
- Additional intervening conditions by which the investigated phenomenon is influenced
- Action and interaction strategies the actor uses to handle the phenomenon
- The consequences of their actions and interactions

Our approach builds on the method defined by Corbin and Strauss, because it offers a more systematic and focused process. This makes it more applicable for the purpose of domain analysis and provides more guidance for a novice analyst. In order to clearly identify the differences which occur when applying GTM to domain analysis, we concentrated on the literature by the original authors and applied their method to our example.

3.2.2 Key Elements

During the research process, data is collected, analyzed and the essential findings about the area under study are abstractly represented in a theory. The process of GTM is iterative and the different phases cannot be clearly separated, but are highly interwoven (Corbin & Strauss, 1996). The key elements of this process are explained below.

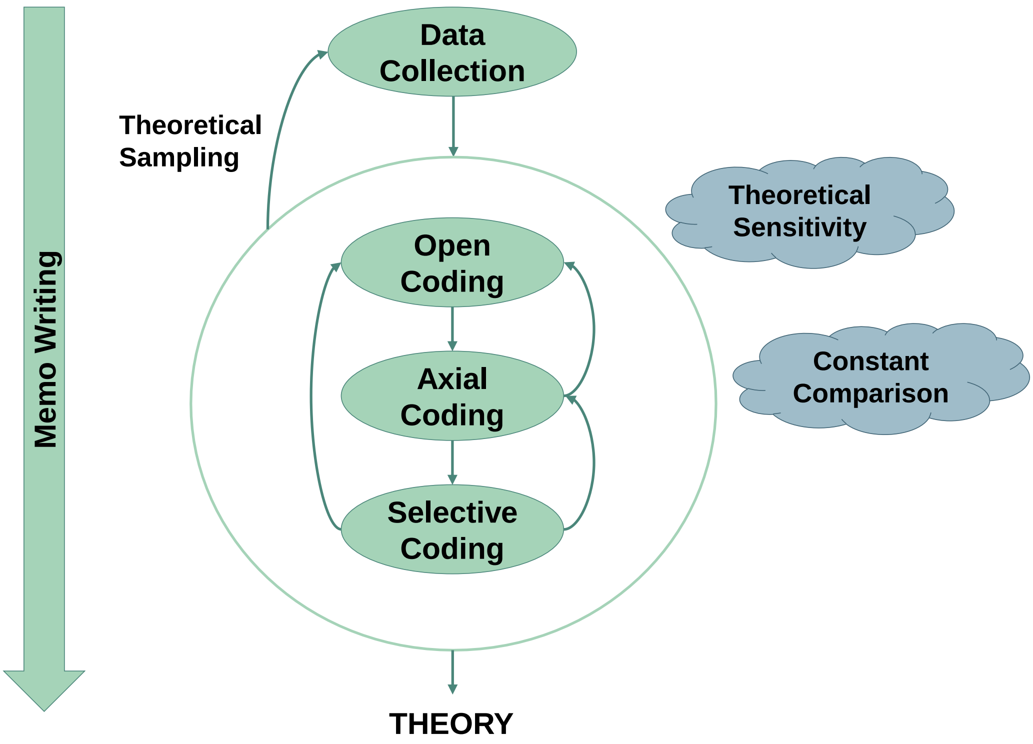


Figure 8: GTM Process

3.2.2.1 Theoretical Sensitivity

Theoretical sensitivity describes the researcher's ability to interpret and reflect upon data with the help of theoretical terms in order to gain insights. This includes identifying significant data and assigning it a meaning, while being aware of nuances. The researcher needs to be able to keep analytical distance while at the same time making use of his experience and theoretical knowledge. Theoretical sensitivity depends upon the researcher's level of experience in qualitative research and in the phenomena under study. However, it develops during the research process and can be enhanced using techniques for questioning the data or systematically analyzing a word or phrase and comparing different incidents. (Corbin & Strauss, 1996; Halaweh, 2012b; Kelle, 2010)

3.2.2.2 Theoretical Sampling

Data analyzed in GTM can be qualitative or quantitative. More common however is qualitative data, in form of interview transcripts, observations, video tapes, articles or books. In contrast to qualitative research, sampling in GTM does not aim at defining a sample which is representative for a population, but one which represents the concepts of the theory in their variety. This includes decisions about which people, places, and situations to investigate and what kind of data to use and can also define the questions asked in an interview. Except for the initial sampling at the beginning of the research, which is based on the general research question and kept very open, the sampling of new data is based on the analysis of already collected data. Thus, categories are enriched by investigating developed concepts further and discovering variations. Theoretical sampling is closely connected to theoretical sensitivity. The latter guides sampling decisions by providing directions for further investigations through questioning the data, identifying significant concepts and hypothesizing. (Corbin & Strauss, 1996; 2014)

3.2.2.3 Coding

Coding involves breaking up the data, conceptualizing it and reassembling it in a new and more abstract way. This is used to capture the substantive content under study and to articulate relationships observed in the data. A code is a conceptual label assigned to a unit of data (Corbin, 2008). Corbin and Strauss (1996) divide the process of coding into three steps, which are conducted in an iterative process. Open coding is the process of identifying and comparing concepts shown in the data and grouping those concepts into categories. The developed concepts and categories are then set into relation during axial coding. In the final stage, selective coding, the categories are integrated around a core category to form a theory. Two aspects are very important throughout the whole coding process: constant comparison and questioning. Only through these techniques the analyst can abstract incidents discovered in the data and thus capture the structure of the area under study in its complexity (Corbin & Strauss, 1996). Coding is an iterative process, in which the analyst goes back and forth between the data and the developed conceptualizations. This also means that developed concepts and categories are not fixed, but should be adapted according to new findings during analysis (Corbin & Strauss, 1996; Gibson & Hartman, 2014).

Coding and theoretical sampling is conducted until theoretical saturation has been reached. This means that all categories and their relationships are well developed and new data does not bring any important new information regarding the categories (Corbin & Strauss, 1996; Hughes & Wood-Harper, 1999).

3.2.2.4 Memo Writing

Writing memos is an integral part of GTM. Memos are written protocols about ideas the researcher has about concepts and their relationships which help him to find gaps, to abstract, and in general to construct the theory. They include thoughts about emergent categories, finding the right terminology, relationships between categories, how theoretical sampling has been conducted, and possible new directions of research. Thus they document the researcher's trail of thought and his decisions regarding the research approach during the whole research process. (Corbin & Strauss, 1996; Gibson & Hartman, 2014)

Code memos are attached to codes and contain the result of the coding process, which are the conceptual terms, properties or indicators of a process (Corbin & Strauss, 1996).

3.2.2.5 Use of Literature

While Glaser advises against conducting literature research prior to and during the analysis process, Corbin and Strauss believe that literature as well as the researcher's knowledge and experience should be used to support the analysis (Gibson & Hartman, 2014). However, the researcher needs to make sure not to force ideas on the data or prevent categories from evolving from the data. If this is kept in mind, specialist literature can be used to animate theoretical sensitivity, to evoke questions, to guide theoretical sampling or as a secondary data source. For example, literature can be compared in order to find aspects which could extend categories or find ideas for new data sources (Corbin & Strauss, 1996). As a precaution, literature research should be avoided until the main categories of the theory have emerged (Gibson & Hartman, 2014). After the theory has been constructed, literature can also be used to verify the theory (Corbin & Strauss, 1996).

3.3 Comparing Grounded Theory Method and Domain Analysis

3.3.1 Purpose of Analysis

GTM is used to explain social phenomena relating to for example interpersonal relations, life, or the workings of an organization (Corbin & Strauss, 1996). Within the area of study, researchers are interested in interactions between and among various types of social units and how they act to deal with a phenomenon (Hughes & Wood-Harper, 1999). The outcome of GTM is a theory which consists of well-developed concepts and sets of concepts which are systematically interrelated to explain phenomena (Corbin, 2008; Hughes & Wood-Harper, 1999). These phenomena are not seen as being static, but as part of an interactive social process, thus causes and conditions leading to as well as consequences of a phenomenon are analyzed (Corbin & Strauss, 1996).

The purpose of identifying the relevant phenomena within a research area and showing how they are related can be transferred to RE. The context in which RE takes place is usually a socio-technical work system, in which humans interact with one another and with systems (Chakraborty et al., 2015). The aim is to uncover the business problem, which is supposed to be solved with a software system and to provide a solution for it (Robertson & Robertson, 2006). To achieve this, elements of the environment in which the software system is supposed to operate and relationships and interactions within this environment have to be analyzed (Browne & Rogich, 2001; Broy, 2013). These elements of a domain therefore represent the phenomena under study, which can be visualized with a domain model. However, although both structural and dynamic aspects and relationships of the domain are investigated in RE, a domain model in form of a UML class diagram only represents the structural aspects and does not incorporate the process aspect of GTM. Another very important difference is that phenomena in GTM are not only described but explained. The researcher should analyze possible theoretical meanings behind data and codes (Charmaz, 2014). For example, instead of coding with in vivo codes such as “boosting self-confidence” or “growing as a person”, which are merely descriptive, the appropriate concept would be “empowerment” (Holton, 2010). For uncovering a business problem and developing a solution, the analyst also has to understand the user’s goals, assumptions, opinions and desires (Browne & Rogich, 2001). However, as a domain model needs to visualize the important concepts and terminology of a domain in order to communicate with stakeholders, the goal of the analysis is not to find underlying theoretical meaning, but to give an abstract description of the domain. While this also involves conceptualizing the data and investigating relationships between concepts, the outcome does not represent a theory as defined in GTM.

In contrast to GTM, where informal integrative diagrams are used to visualize the relationships between concepts (Corbin, 2008), results of RE are represented in more formal documents and models. Although informal ways of documenting requirements can also be used, applying to certain quality criteria and standards and using formal modelling languages is recommended to secure the correctness and completeness of requirements (Pohl & Rupp, 2011).

3.3.2 Data Sources

In GTM, mostly qualitative data is used, such as interview transcripts or observations (Corbin & Strauss, 1990). Requirements also primarily emerge from empirical qualitative data, as interviewing stakeholders is the most commonly used requirements elicitation technique (Chakraborty et al., 2015; Pitts & Browne, 2007). For domain analysis, knowledge elicited from domain experts is an important data source, in addition to use cases, glossaries, and high level problem statements (Rosenberg & Scott, 2001; Wazlawick, 2014). The areas under study both in GTM and domain analysis are highly complex and the gathered data might therefore be unstructured, inconsistent and incomplete (Balzert, 2009; Corbin & Strauss, 1996; Rupp & SOPHISTen, 2014). Therefore, both disciplines require a systematic method for extracting the important information from qualitative data.

3.3.3 Analysis Process

Elicitation, analysis, specification, validation and management of requirements are interwoven activities, which form an iterative process (Wiegers, 2003). In the early phases of RE, data elicited from different stakeholders needs to be conceptualized to derive an abstract description of the world in which an envisioned system will operate (Easterbrook & Nuseibeh, 2000). In an iterative process, nouns are extracted from the data, abstracted to develop classes, and described with attributes and associations (Wazlawick, 2014). GTM provides a more systematic procedure for breaking up data, conceptualizing it and reassembling it in a new and more abstract way (Charmaz, 2014; Corbin & Strauss, 1996). In GTM, similar incidents in the data are being coded with concepts. These concepts are constantly compared to each other and are grouped to categories. Through questioning, relationships between concepts and categories are identified. The sampling of further data is guided by analysis results, thus developed categories are not fixed, but are adapted according to new findings until they are well developed. Therefore, similar to domain analysis and RE, the process of GTM is iterative and the different coding phases are highly interwoven (Corbin & Strauss, 1996).

Questioning techniques are used to gain insights, to improve theoretical sensitivity and to discover aspects, which should be further investigated (Corbin & Strauss, 1996). A requirements engineer also uses questioning techniques to interpret the data and to discover information gaps (Rupp & SOPHISTen, 2014). Some of these questioning techniques overlap, for example the use of W-questions to investigate phenomena further or the determination of variations.

3.3.4 Accountability of Analysis Result

GTM is an inductive research approach, in which the explanations about social phenomena are grounded in and emerge from empirical data. A theory must fit the substantive area and correspond to the data (Corbin & Strauss, 1990, 1996). In addition, it must be understood by and make sense to practitioners in the study area (Corbin & Strauss, 1996). These quality criteria are also important for requirements, which need to correctly represent the ideas of the stakeholders and be clearly understandable (Pohl & Rupp, 2011). A derived domain model must correctly represent the reality of the domain and be easy to understand from the stakeholders perspective (Bolloju & Leung, 2006; Cruzes, Vennesland, & Natvig, 2013). Therefore, the results of RE also need to be grounded in empirical data and be traceable, especially because goals and requirements are constantly changing (Partsch, 2010; Rupp & SOPHISTen, 2014). Traceability means that the origin, the realization and the relation to other RE artefacts of a requirement can be retraced (Pohl & Rupp, 2011). Pre-requirements traceability thereby focuses on the origin of requirements, documenting how stakeholders' needs were discovered and how these needs were integrated in the requirements specification (Gotel & Finkelstein, 1996). In GTM, special attention is paid to an rigorous and comprehensible reasoning and research process in order to ensure the credibility of a theory (Corbin, 2008; Corbin & Strauss, 2014). Concepts are directly connected to the data in which they are grounded and memos are used to document the analyst's thoughts, interpretations and decisions taken during analysis (Gibson & Hartman, 2014). The application of GTM to RE can therefore assist in ensuring traceability and improving the accountability of RE artefacts such as the domain model.

3.3.5 Analyst's Skills

From analysts of both disciplines, good social and communication skills are expected. In addition, a researcher using GTM must be theoretically sensitive, which means that he must be able to interpret and reflect upon data with the help of theoretical terms in order to gain insights (Corbin & Strauss, 1996). This includes identifying significant data and assigning it a meaning, while being aware of nuances. The researcher needs to be able to keep analytical distance while at the same time making use of his experience and theoretical knowledge. This ability is useful for requirements engineers as well, as they also need to be able to think analytically in order to correctly conceptualize the problem which is supposed to be solved with software (Balzert, 2009; Browne & Rogich, 2001; Hruschka, 2014; Rupp & SOPHISTen, 2014). Their task requires expertise and experience in the domain and the respective methods of RE (Hruschka, 2014; Rupp & SOPHISTen, 2014). Although theoretical sensitivity also depends upon the researcher's level of experience in qualitative research and in the phenomena under study, it develops further during the research process and can be enhanced using techniques for questioning the data or systematically analyzing a word or phrase and comparing different incidents (Corbin & Strauss, 1996; Halaweh, 2012b; Kelle, 2010). This suggests that, while a requirements engineer still benefits from his experience in the domain under study, a systematic analysis procedure could support him to develop theoretical sensitivity in regard to domain analysis.

3.4 Preliminary Research Project

In our preliminary research, we aimed at producing a requirements specification and glossary using GTM parallel to requirements analysts in a real world industry project to compare the results and draw conclusions on how to improve RE through the use of GTM. We accompanied seven workshops which were conducted at the beginning of a software development project for replacing an existing software. The goal of these workshops was to write a complete high level requirements description of the software to be developed. The participants worked in product management and product development. Because audio recording was not permissible, two observers took as verbatim notes as possible during the workshops. Due to the high speed of the discussion, some parts of the conversation could not be transcribed. To limit this negative effect, the transcripts of both observers were combined in order to receive a complete transcription of the discussions. However, not all inconsistencies in the data could be clearly resolved afterwards. The observations resulted in more than 200 pages of transcript, which were analyzed using GTM coding.

During analysis, several problems arose. First, the workshops followed a top-down approach in which the participants identified the important domains and subdomains which were supposed to be managed with the software. The outcome of the workshops was a hierarchical list of these domains. GTM, by contrast, follows a bottom-up or middle-out approach in which concepts and categories are generated through constant comparison and related to each other. Due to the predefined hierarchy of functionalities discussed in the workshops however, the analysis resulted in a mere description of the topics covered in the discussions. Second, the contributions to these discussions consisted mostly of short sentences or single phrases. Terms used were not further described but seemed to be clear to everyone or were further described in documents to which we did not have access. Theoretical sampling to answer open questions could not be applied because we were only passive observers. Therefore, the information which could be elicited from analyzing the transcriptions was limited, the iterative elicitation and analysis process could not be applied fully and new concepts and categories did not emerge from the data.

Nevertheless, by using GTM, the process of RE could be analyzed. Difficulties faced during the workshops, such as finding the right granularity in requirements, could be identified through coding the respective incidents and comparing them. This gave valuable insights about challenges that need to be addressed to improve the process. However, our research goal was not to investigate the organization and their processes, but to apply GTM as an actual RE procedure. Therefore, these analysis results were not further investigated.

3.5 Validation of Domain Analysis from a GTM Perspective

In addition to the validation of our domain analysis results from a RE perspective as described in chapter 2.6.2, both the results and the procedure of analysis are discussed in the following from a GTM perspective.

3.5.1 Validation of Analysis Results

According to Corbin and Strauss (1996), a grounded theory needs to fulfil the following factors:

- Fit: The theory must fit the substantive area and correspond to the data.
- Understanding: The theory makes sense to practitioners in the study area.
- Generality: The theory must be sufficiently abstract to be a general guide without losing its relevance.
- Control: The theory acts as a general guide and enables the person to fully understand the situation.

These quality criteria partly correspond to the criteria we used to evaluate the domain model from a RE perspective. The factors "fit" and "understanding" represent the semantic and pragmatic quality of a domain model. The generality is given because we integrated several company specific domain descriptions into one abstract model, thus the result is not company specific but generally applicable. As our analysis results do not present a theory, which aims at explaining actions directed at a phenomenon, the factor "control" is not relevant. If the developed domain model can be used for communicating with stakeholders and as a basis for further software development steps, is evaluated with the above mentioned quality criteria for domain models.

3.5.2 Validation of Analysis Procedure

To validate the derivation of a theory, Corbin and Strauss (1990; 2014) propose to evaluate the research process and the theoretical grounding of the developed theory using a number of questions. Although not all of these questions are relevant when evaluating a domain analysis, they can be used to reflect upon the analysis process.

3.5.2.1 Adequacy of the Research Process

In order to judge the adequacy of the research process, the researcher needs to comprehensibly describe how research was conducted by providing the following information.

- How was the original sample selected? On what grounds?
- Which major categories emerged?
- What were some of the events, incidents or actions (indicators) that pointed to some of these major categories?
- On the basis of what categories did theoretical sampling proceed? That is, how did theoretical formulations guide some of the data collection? After the theoretical sampling was done, how representative of the data did the categories prove to be?
- What were some of the hypotheses pertaining to conceptual relations (i.e. among categories), and on what grounds were they formulated and validated?
- Were there instances in which hypothesis did not explain what was happening in the data? How were these discrepancies accounted for? Were hypothesis modified?
- How and why was the core category selected? Was this collection sudden or gradual, and was it difficult or easy? On what grounds were the final analytic decisions made?

As we wanted to analyze the domain of HR development, the original sample needed to consist of HR representatives from different companies with expertise in HR development. We therefore contacted 14 domain experts who fit this criteria. However, the sample depended on the received responses and availability of domain experts. During the analysis, several main categories emerged, including “competency”, “performance”, “development measure”, “employee assessment” and “staff appraisal”. These aspects were developed into categories because the data showed that they are central to the domain. To give an example, several aspects discovered in the data pointed towards the category “development measure”. These included the mentioning of several types of development measures, the description of employees attending a development measure and other activities such as organizing or evaluating development measures. Inconsistencies or information gaps identified during analysis guided the questions for the following interviews. For example, because the relationship between competency, performance and employee assessment was not clear, we asked domain experts how competency and performance were related and according to which aspects employees were assessed. To increase the depth of the analysis and clarify some uncertainties, we also conducted follow up interviews with two of the domain experts in which we asked more detailed questions about aspects they mentioned in their first interview. Theoretical sampling in regards to the selection of interview partners could not be performed due to limited availability of domain experts. Ideas for possible relationships between concepts were documented in code memos and investigated through conducting and analyzing further data. For example, it had to be investigated how a development need was identified. Several possibilities were mentioned in the data, such as a change of job responsibilities. This led to the hypothesis that the job and therefore the requirements profile of a job results in development needs for an employee. This hypothesis was confirmed during further analysis. Other hypotheses, for example that a promotion is a consequence of an employee assessment, could not be clearly confirmed or disproved and would need to be investigated further in additional interviews. The selection of a single core category was not appropriate for domain analysis. However, concepts were developed into categories on the grounds that they showed to be central to the domain according to the data.

3.5.2.2 Empirical Grounding of the Theory

Because GTM aims at providing theoretical explanations which are grounded in empirical data, this aspect of the research process is evaluated in detail using the following questions.

- Are concepts generated?
- Are the concepts systematically related?
- Are there many conceptual linkages, and are the categories well developed? Do categories have conceptual density?
- Is variation built into the theory?
- Are the conditions under which variation can be found built into the study and explained?
- Has process been taken into account?
- Do the theoretical findings seem significant, and to what extend?

- Does the theory stand the test of time and become part of the discussions and ideas exchanged among relevant social and professional groups?

Concepts were generated during our analysis by coding structural and dynamic aspects of the domain and constantly comparing them. All concepts were related with structural and dynamic relationship types and grouped into categories. The categories represented aspects which showed to be central to the domain according to the data. They were further described by identifying their properties and activities directed at them and were structurally related to each other. Some categories could not be fully developed, thus theoretical saturation could not be reached due to limited availability of interview partners. Variation was examined during analysis while comparing the company specific descriptions of HR development. However, for deriving a domain model, these variations needed to be integrated into a consistent model. Because of the limited data available, we concentrated on investigating structural relationships which are represented in a domain model. Some dynamic relationships were identified as well, which for example described the process of performance management. Through investigating these more closely in further analysis, process descriptions can be derived as well. The evaluation of significance and presence in research discussions are specific to theory development and not relevant for evaluating a domain analysis.

4 Appendices

A: Interview Script

The following interview script (based on King & Horrocks, 2010; Myers & Newman, 2007) was used in the first of our interviews. However, as we conducted semi-structured interviews, the key questions were only used as a guideline and we reacted to participants answers (Corbin & Strauss, 1996; Myers & Newman, 2007). In addition, preliminary analysis results led to additional key questions for the following interviews (Corbin & Strauss, 1996). The interviews were conducted in German and all but two interviews were conducted by telephone.

1. Introduction
    - Thanking for their willingness to participate
    - Giving background information on the research project and explaining the purpose of the interview
    - Asking permission to audio record the interview and explaining how the data will be processed
2. Key questions
    - What is your position in the company?
    - What does HR development contain?
    - What is the purpose of HR development?
    - Can you please explain the processes of HR development?
        - When is this process performed?
        - Which steps does the process include?
        - Who performs these steps?
        - What information is needed?
        - How would you like to change the process?
    - What kind of measures does HR development use?
    - How are employees assessed?
    - What kind of evaluations are important for HR development?
3. Closing
    - Thanking for their time
    - Asking permission to follow-up

B: Code System

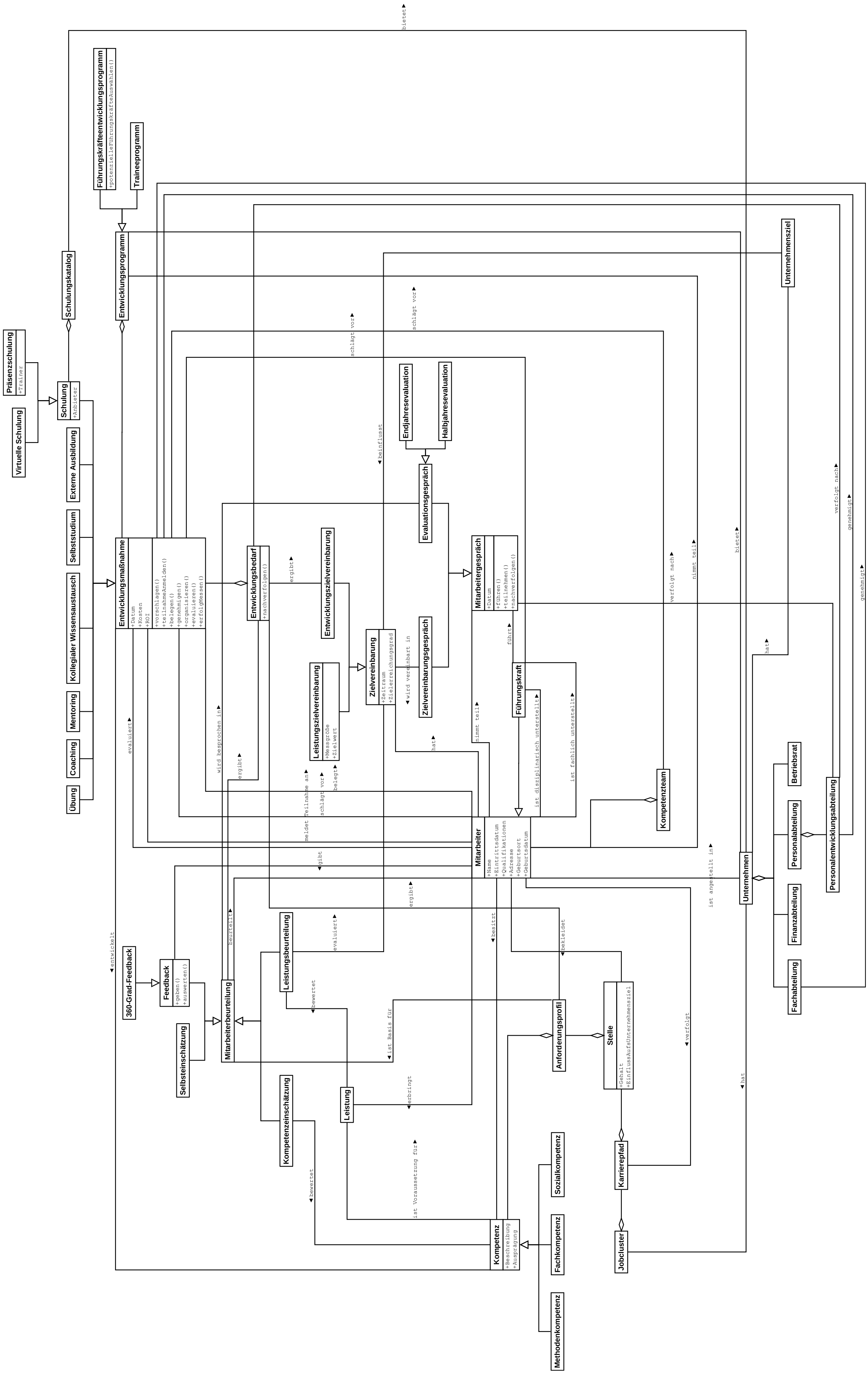
Note: codes with quotation marks represent in vivo codes, codes in italics represent background information

- Mitarbeiterbeurteilung
  - "Selbsteinschätzung"
  - "Feedback"
    - Feedbackfragebogen
    - Feedback geben
    - Feedback auswerten
    - "360-Grad-Feedback"
  - "Kompetenzeinschätzung"
  - "Leistungsbeurteilung"
- Leistung
  - "Performance Management"
    - "Mitarbeitergespräch"
      - Datum
      - Mitarbeitergespräch führen
      - An Mitarbeitergespräch teilnehmen
      - Mitarbeitergespräche nachverfolgen
      - "Gesprächsleitfaden"
      - "Zielvereinbarungsgespräch"
        - Unternehmensziel
        - "Zielvereinbarung"
          - Zielerreichungsgrad
            - "Variable Vergütung"
          - Zeitraum
          - Entwicklungszielvereinbarung
          - Leistungszielvereinbarung
            - Messgröße
            - Zielwert
      - Evaluationsgespräch
        - Endjahresevaluation
        - "Halbjahresevaluation"
- Kompetenz
  - Ausprägung
  - Beschreibung
  - Methodenkompetenz
  - Fachkompetenz
  - "Sozialkompetenz"
- "Jobcluster"
  - Stelle
    - Einfluss aufs Unternehmensziel
    - "Gehalt"
    - "Anforderungsprofil"
  - "Karrierepfad"
- "Entwicklungsbedarf"
  - Geplante Maßnahmen nachverfolgen
  - "Rollenwechsel"

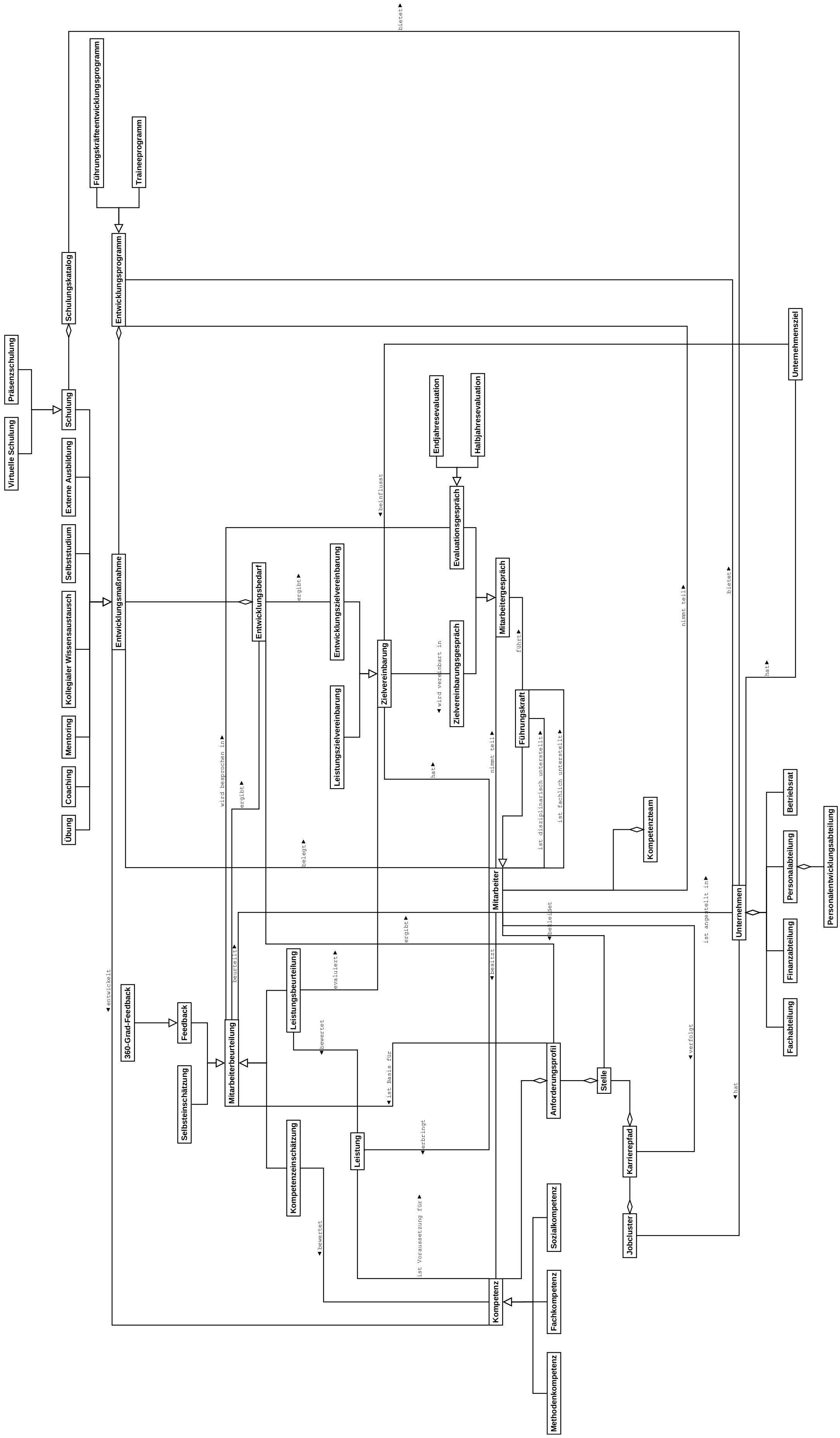
        - "Beförderung"
    - Marktsituation
- Entwicklungsmaßnahme
    - Datum
    - Kosten
    - Entwicklungsmaßnahme vorschlagen
    - Mitarbeiter informieren
    - Teilnahme an Entwicklungsmaßnahme anmelden
    - Entwicklungsmaßnahme genehmigen
    - Entwicklungsmaßnahme belegen
    - Entwicklungsmaßnahmen organisieren
    - Entwicklungsmaßnahme evaluieren
    - "Erfolg von Entwicklungsmaßnahmen messen"
        - "Return on Investment"
    - Übung
    - "Coaching"
    - Mentoring
    - Kollegialer Wissensaustausch
    - Selbststudium
    - "Externe Ausbildung"
    - Schulung
        - Anbieter
        - "Schulungskatalog"
        - "Virtuelle Schulung"
        - "Präsenzschulung"
            - Trainer
    - Entwicklungsprogramm
        - "Traineeprogramm"
        - "Führungskräfteentwicklungsprogramm"
            - Potentiellen Führungskräfte auswählen
- Akteure
    - "Benutzerrechte"
    - "Mitarbeiter"
        - "Name"
        - "Geburtsdatum"
        - "Geburtsort"
        - "Qualifikationen"
        - Eintrittsdatum
        - "Adresse"
        - Neueinstellung
        - Führungskraft
    - "Kompetenzteam"
    - Unternehmen
        - "Personalabteilung"
            - "Personalentwicklungsabteilung"
        - "Betriebsrat"
        - Fachabteilung
        - "Finanzabteilung"

- *Aufgaben der Personalentwicklung*
    - *Kompetenz halten und entwickeln*
    - *Unternehmenswerte vermitteln*
    - *"Mitarbeiter motivieren"*
    - *"Mitarbeiter binden"*
- *Aktuelle Situation*
    - *Unternehmensgröße*
    - *Branche*
    - *Macht der Personalabteilung*
    - *Position des Interviewten*
    - *Bestehende Tools*
    - *Konzept erstellen*

C: Complete Domain Model



D: Reduced Domain Model



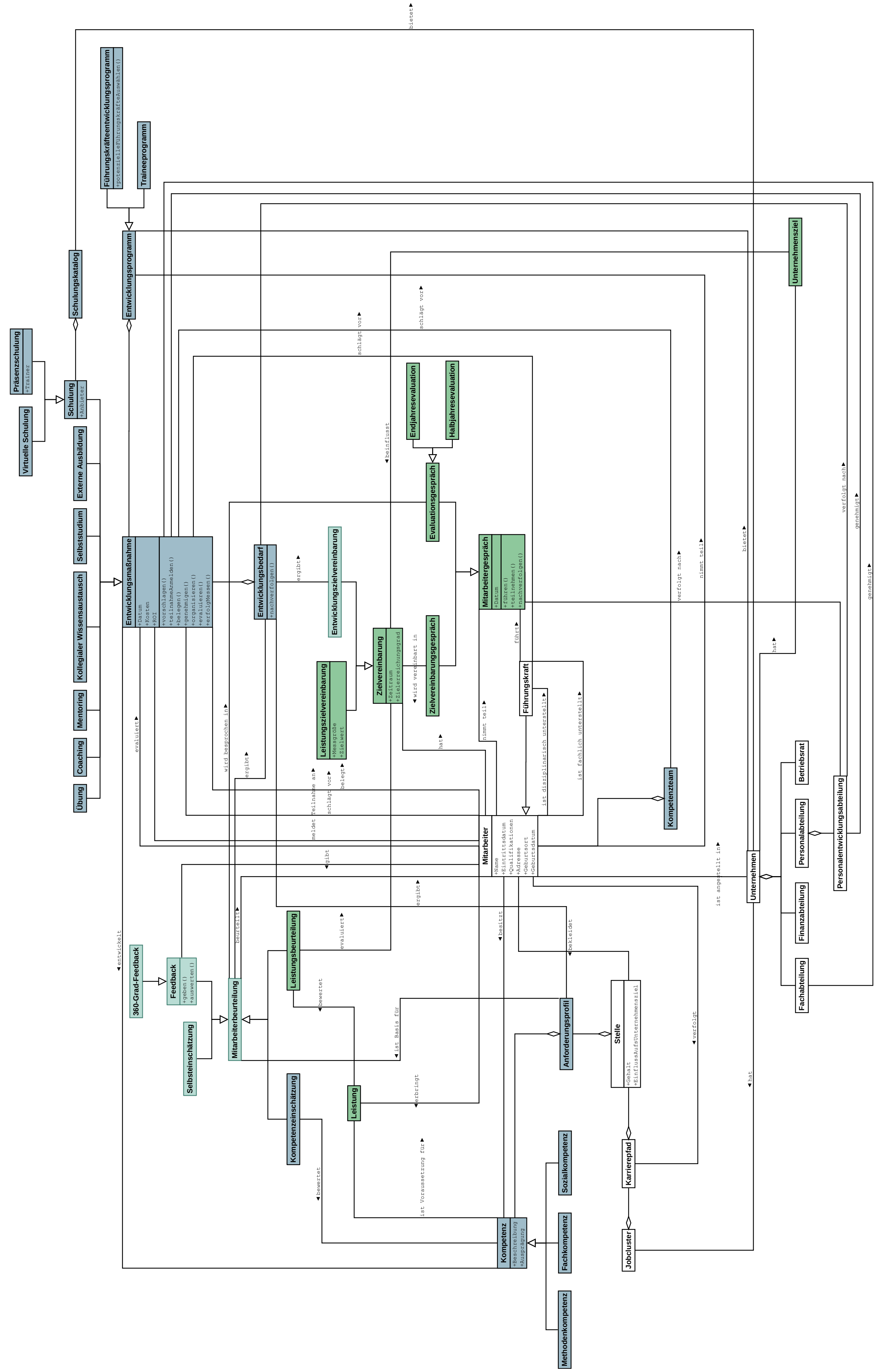
E: Colored Domain Model

Notation:

competency-related class

performance-related class

competency and performance-related class


Präsenzschulung
Virtuelle Schulung
Schulung
Schulungskatalog
Führungskräfteentwicklungsprogramm
Traineeprogramm
Entwicklungsprogramm
Externe Ausbildung
Selbststudium
Kollegialer Wissensaustausch
Mentoring
Coaching
Übung
Entwicklungsmaßnahme
Entwicklungsbedarf
Entwicklungszielvereinbarung
Leistungszielvereinbarung
Zielvereinbarung
Zielvereinbarungsgespräch
Evaluationsgespräch
Endjahresevaluation
Halbjahresevaluation
Mitarbeitergespräch
Führungskraft
Mitarbeiter
Kompetenzteam
Unternehmensziel
Unternehmen
Betriebsrat
Personalabteilung
Personalentwicklungsabteilung
Finanzabteilung
Fachabteilung
360-Grad-Feedback
Feedback
Selbsteinschätzung
Mitarbeiterbeurteilung
Leistungsbeurteilung
Kompetenzeinschätzung
Leistung
Anforderungsprofil
Stelle
Karrierepfad
Jobcluster
Kompetenz
Sozialkompetenz
Fachkompetenz
Methodenkompetenz

F: Glossary

Title	Memo Text	Author	Creation Date
360-Grad-Feedback	Definition: Feedback wird nicht nur von der Führungskraft, sondern auch von Kollegen, unterstellten Mitarbeitern, Kunden etc. eingeholt, um ein ganzheitliches Bild eines Mitarbeiters zu erhalten Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Feedback Notes: Wem wird das Feedback angezeigt?	Katharina	15.01.2015 10:58:00
Adresse	Definition: Wohnort eines Mitarbeiters (Straße, Hausnummer, Postleitzahl, Stadt) Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Mitarbeiter Notes:	Katharina	29.04.2015 17:30:00
Anbieter	Definition: Organisation, die die Schulung erstellt, organisiert und ausführt Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Schulung Notes:	Katharina	08.03.2015 14:45:00
Anforderungs-profil	Definition: Liste an fachlichen und persönlichen Anforderungen, die für die Bewältigung der Aufgaben einer Stelle nötig sind Synonyms: Performance-Cluster, Kernqualifikationen, Soll-Profil Abbreviations: Concept Type: category (object) Relationship Type: is part of Stelle; is related to Entwicklungsbedarf (ergibt); is related to Mitarbeiterbeurteilung (ist Basis für) Notes: für jedes Rolle festgelegte Erwartungen, nötige Kompetenzen. Liste und Beschreibung Kompetenzen (fachlich, sozial), die gewünscht sind. Soll-Zustand	Katharina	15.01.2015 10:55:00
An Mitarbeiter-gespräch teil-nehmen	Definition: Mitarbeiter nimmt an einem Mitarbeitergespräch mit seiner Führungskraft teil Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Mitarbeitergespräch Notes:	Katharina	22.05.2015 17:26:58

Ausprägung	Definition: Einstufung des Levels einer Kompetenz auf einer Skala Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Kompetenz Notes: Skala abhängig von Unternehmen: entwicklungsbedürftig bis sehr gut, gar nicht bis überragend, untererfüllt-erfüllt-übererfüllt	Katharina	05.03.2015 16:36:00
Beschreibung	Definition: Textuelle Definition einer Kompetenz Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Kompetenz Notes: mit Beschreibung, wie die Kompetenz geprüft werden kann	Katharina	26.02.2015 15:10:00
Betriebsrat	Definition: Institutionalisierte Arbeitnehmervertretung im Unternehmen Synonyms: Abbreviations: Concept Type: category (actor) Relationship Type: is part of Unternehmen Notes:	Katharina	08.04.2015 18:34:00
Coaching	Definition: Einzelbetreuung durch einen Coach zur Entwicklung eines Mitarbeiters, in der individuell auf Entwicklungsbedürfnisse und Fragen eingegangen werden kann Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Entwicklungsmaßnahme Notes:	Katharina	07.02.2015 08:43:00
Datum	Definition: Tag und Uhrzeit, zu welcher das Mitarbeitergespräch stattfindet Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Mitarbeitergespräch Notes:	Katharina	09.03.2015 12:43:00
Datum	Definition: Tag/Tage, an dem/denen eine Entwicklungsmaßnahme stattfindet Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Entwicklungsmaßnahme Notes:	Katharina	26.03.2015 16:40:00

Einfluss aufs Unter-nehmensziel	Definition: Ausmaß in dem sich die Leistung einer Stelle auf das Unternehmensziel auswirkt Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Stelle Notes: Hat Einfluss auf Zielvereinbarungen und in wie weit die variable Vergütung von der Zielerreichung ab-hängt	Katharina	24.04.2015 09:20:00
Eintritts-datum	Definition: Tag, Monat und Jahr, an dem ein Mitarbei-ter vom Unternehmen eingestellt wurde Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Mitarbeiter Notes:	Katharina	29.04.2015 17:30:00
Endjahres-evaluation	Definition: Gespräch am Ende eines (Fiskal-)jahres um die Zielerreichung zu evaluieren und die Mitarbeiterbe-urteilung zu besprechen Synonyms: Endjahresgespräch, Review Abbreviations: Concept Type: category (event) Relationship Type: is a Evaluationsgespräch Notes: Im Performance Management Cycle am Ende des Jahre Es wird besprochen: Hat der Mitarbeiter seine Ziele er-reicht? Wo steht der Mitarbeiter, auch im Vergleich zu anderen Mitarbeitern? Wurden die geplanten Entwicklungsmaßnahmen durchgeführt? Wie haben sich seine Kompetenzen dadurch verändert --> Erfolgsmessung von Entwick-lungsmaßnahmen Input: Ziele, Leistungsbeurteilung, Kompetenzeinschät-zung und evtl. Dokumentation der Halbjahresevalua-tion Fällt zeitlich oft mit Zielvereinbarungsgespräch zusam-men (ein Gespräch)	Katharina	17.01.2015 09:07:00
Entwicklungs-bedarf	Definition: Bedarf an Entwicklungsmaßnahmen für ei-nen Mitarbeiter im kommenden Jahr Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: Notes: Personal bekommt diese Daten und leitet Orga-nisation der Maßnahmen ein Vorgesetzter ist für Durchführung der Maßnahmen ver-antwortlich, muss vom Personal gemonitort werden	Katharina	17.01.2015 09:27:00

Entwicklungs-maßnahme	Definition: Maßnahme zur Entwicklung eines Mitarbeiters Synonyms: Weiterbildungsmaßnahme Abbreviations: Concept Type: category (event) Relationship Type: is part of Entwicklungsbedarf; is related to Kompetenz (entwickelt); is part of Entwicklungsprogramm Notes:	Katharina	12.02.2015 10:14:00
Entwicklungs-maßnahme belegen	Definition: Mitarbeiter nimmt an einer Entwicklungs-maßnahme teil Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Entwicklungsmaßnahme Notes:	Katharina	09.04.2015 09:17:00
Entwicklungs-maßnahme evaluieren	Definition: Teilnehmer bewerten Entwicklungsmaßnahme Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Entwicklungsmaßnahme Notes: mit Fragebögen	Katharina	12.02.2015 11:24:00
Entwicklungs-maßnahme vorschlagen	Definition: Mitarbeiter schlägt eine Entwicklungsmaßnahme vor Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Entwicklungsmaßnahme Notes:	Katharina	16.01.2015 10:58:00
Entwicklungs-maßnahmen genehmigen	Definition: Fachabteilung und Personalentwicklungsabteilung genehmigen Entwicklungsmaßnahme Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Entwicklungsmaßnahme Notes:	Katharina	17.01.2015 09:25:00
Entwicklungs-maßnahmen organisieren	Definition: Entwicklungsmaßnahme wird vorbereitet (Trainer, Raum, Anmeldungen/Teilnehmer, Verpflegung/Unterkunft, Material bzw. Anbindung ans Tool bei eLearning) Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: is consequence of Entwicklungsbedarf; affects Entwicklungsmaßnahme Notes: durch Personalentwicklung evtl. Hilfe von Kompetenzteams? Unterscheidung zwischen Organisation von offenen Angeboten bzw. individualisiertem Bedarf?	Katharina	18.01.2015 10:20:00

Entwicklungs-programm	Definition: Im Unternehmen festgelegtes Programm, das auf eine bestimmte Zielgruppe ausgerichtet ist und verschiedene Entwicklungsmaßnahmen beinhaltet Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: Notes:	Katharina	15.03.2015 13:34:00
Entwicklungs-ziel-vereinbarung	Definition: Ziele bezüglich der Entwicklung eines Mitarbeiters, meist qualitativ Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Zielvereinbarung; is related to Entwicklungsbedarf (ergibt) Notes:	Katharina	09.02.2015 17:42:00
Erfolg von Entwicklungs-maßnahmen messen	Definition: Nutzen einer Entwicklungsmaßnahme für den Mitarbeiter und das Unternehmen wird bestimmt Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Entwicklungsmaßnahme Notes: anhand von KPIs und ob sich Kernqualifikationen geändert haben	Katharina	16.01.2015 11:19:00
Evaluationsge-spräch	Definition: Mitarbeitergespräch, das zur Evaluation der Zielerreichung und Besprechung der Mitarbeiterbeurteilung dient Synonyms: Review-Gespräche, Review Abbreviations Concept Type: category (event) Relationship Type: is a Mitarbeitergespräch Notes:	Katharina	15.03.2015 11:25:00
Externe Ausbildung	Definition: Weiterbildungsprogramm eines externen Anbieters mit Abschluss Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Entwicklungsmaßnahme Notes:	Katharina	08.03.2015 08:46:00
Fach-abteilung	Definition: Fachlicher Bereich des Unternehmens Synonyms: Abbreviations: Concept Type: category (actor) Relationship Type: is part of Unternehmen; affects Entwicklungsmaßnahme genehmigen Notes:	Katharina	08.04.2015 18:34:00

Fach- kompetenz	Definition: Kompetenz im Bezug auf das Fachgebiet einer Rolle Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Kompetenz Notes:	Katharina	18.01.2015 13:29:00
Feedback	Definition: Dokumentierte, formelle Rückmeldung über das wahrgenommene Verhalten eines Mitarbeiters Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Mitarbeiterbeurteilung Notes: Informelles Feedback sollte auch außerhalb der Mitarbeiterbeurteilung und -gespräche stattfinden! Feedback muss abgefragt (über Tool, Erinnerung) und dokumentiert werden.	Katharina	18.01.2015 13:31:00
Feedback auswerten	Definition: Eingeholtes Feedback wird verglichen und ausgewertet Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Feedback Notes: wird automatisch bzw. händisch ausgewertet und von Führungskraft und Personalentwicklungsabteilung angeschaut	Katharina	18.02.2015 11:52:00
Feedback geben	Definition: Personen geben Feedback Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Feedback Notes: mit Hilfe von Fragebogen, je nach Feedbacktyp wird Feedback durch Kollegen, Vorgesetzte, Kunden,... gegeben	Katharina	18.02.2015 11:53:00
Finanz- abteilung	Definition: Abteilung, die für das Finanzwesen im Unternehmen zuständig ist Synonyms: Abbreviations: Concept Type: category (actor) Relationship Type: is part of Unternehmen Notes:	Katharina	08.04.2015 18:34:00
Führungskraft	Definition: Mitarbeiter, der Führungsverantwortung für andere Mitarbeiter trägt Synonyms: Vorgesetzter Abbreviations: Concept Type: category (actor) Relationship Type: is a Mitarbeiter; affects Mitarbeitergespräch führen; affects Entwicklungsmaßnahme vorschlagen Notes:	Katharina	08.03.2015 15:13:00

Führungs-kräfte-entwicklungs-programm	Definition: Im Unternehmen definiertes Programm zur Auswahl von potenziellen Führungskräften und zur Entwicklung der notwendigen Kompetenzen Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Entwicklungsprogramm Notes:	Katharina	18.01.2015 13:32:00
Geburts-datum	Definition: Tag, Monat und Jahr an dem ein Mitarbeiter geboren wurde Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Mitarbeiter Notes:	Katharina	29.04.2015 17:30:00
Geburtsort	Definition: Stadt in der ein Mitarbeiter geboren wurde Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Mitarbeiter Notes:	Katharina	29.04.2015 17:30:00
Gehalt	Definition: Jährliche Vergütung Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Stelle Notes: ist meist Rollen im Jobcluster zugeordnet auch andere Variablen (quantitative Ziele etc., Verhandlung etc.), Abhängigkeiten klären!	Katharina	06.02.2015 14:37:00
Geplante Maßnahmen nach-verfolgen	Definition: Personalentwicklungsabteilung prüft, ob die geplanten Maßnahmen im vergangenen Jahr erfolgt sind und wenn nicht, warum nicht Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Entwicklungsbedarf Notes: Wird im Endjahresevaluationsgepräch besprochen und festgehalten. Evtl. auch in Halbjahresevaluation	Katharina	07.04.2015 19:16:00

Halbjahres- evaluation	Definition: Gespräch zur Mitte des (Fiskal-)jahres um zu evaluieren, ob die gesetzten Ziele erreicht werden und welche Maßnahmen noch ergriffen werden müssen Synonyms: Mid-Review, Midyear-Gespräch, Review Abbreviations: Concept Type: category (event) Relationship Type: is a Evaluationsgespräch Notes: Es wird besprochen: Wo steht der Mitarbeiter? Kann er das Ziel erreichen? Ist das Ziel realistisch? Wurden die vereinbarten Entwicklungsmaßnahmen durchgeführt bzw. werden noch durchgeführt? Welche zusätzlichen Maßnahmen müssen durchgeführt werden? Haben externe Einflussfaktoren die Situation verändert? Dokumentation des Gesprächs muss mit Dokumentation des Zielvereinbarungsgesprächs zusammenhängen!	Katharina	16.01.2015 15:03:00
Jobcluster	Definition: Sammlung aller Stellen im Unternehmen, geclustert nach Fachbereichen und Jobhierarchien Synonyms: Laufbahnmodell, Karrieremodell Abbreviations: Concept Type: category (object) Relationship Type: Notes:	Katharina	15.01.2015 10:53:00
Karrierepfad	Definition: Reihe an zeitlich aufeinanderfolgenden (meist hierarchisch aufsteigenden) Stellen, die ein Mitarbeiter in einem Unternehmen durchläuft Synonyms: Laufbahn Abbreviations: Concept Type: category (object) Relationship Type: is part of Jobcluster Notes: z.B.: Business Analyst: Junior Business Analyst, Business Analyst Lead, Senior Analyst, Managing Business Analyst, Principal Business Analyst, Vice President z.B.: Assistenten, Junior Berater, Berater, Senior Berater	Katharina	09.02.2015 11:58:00
Kollegialer Wissens- austausch	Definition: Mitarbeiter geben ihr Wissen an Kollegen weiter, durch Präsentationen oder Beratung Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Entwicklungsmaßnahme Notes:	Katharina	16.01.2015 14:11:00

Kompetenz	Definition: Fähigkeit, die relevant für die berufliche Leistung ist und hinreichend messbar/beobachtbar ist Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is related to Leistung (ist Voraussetzung für); is part of Anforderungsprofil Notes: Leistungsvoraussetzung für die Bewältigung beruflicher Aufgaben oder Situationen. Zeigt sich in der konkreten Anwendung in Form kontextgebundener, beobachtbarer Verhaltensweisen. Ist kontextspezifisch, Anforderungen abhängig von Tätigkeit (siehe Anforderungsprofil)(Kurzhals 2011), müssen im Unternehmen definiert sein	Katharina	26.02.2015 11:57:00
Kompetenz-einschätzung	Definition: Subjektive Beurteilung der Ausprägung der Kompetenzen eines Mitarbeiters Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Mitarbeiterbeurteilung; is related to Kompetenz (bewertet) Notes: Abgrenzung zu Leistungsbeurteilung klären! Basis für Einschätzung: Anforderungsprofil Muss im Rahmen von Mitarbeitergesprächen besprochen werden	Katharina	16.01.2015 14:17:00
Kompetenz-team	Definition: Gruppen von Mitarbeiter mit fachlichen Gemeinsamkeiten, welche helfen, passende Schulungen für ein fachliches Gebiet auszuwählen Synonyms: Abbreviations: Concept Type: category (actor) Relationship Type: affects Entwicklungsmaßnahme vorschlagen Notes: organisieren evtl. auch	Katharina	12.02.2015 10:03:00
Kosten	Definition: Kosten einer Entwicklungsmaßnahme inklusive Seminarkosten, Reisekosten und Arbeitsausfall Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Entwicklungsmaßnahme Notes:	Katharina	16.01.2015 14:06:00
Leistung	Definition: Die Leistung eines Mitarbeiters im Sinne konkreter Handlungen zur Berufsbewältigung Synonyms: Performance, Performanz Abbreviations: Concept Type: category (object) Relationship Type: Notes: Abgrenzung zur Leistung im Sinne quantitativer Ziele, z.B. Verkaufszahlen etc.?	Katharina	11.02.2015 13:16:00

Leistungs- beurteilung	Definition: Bewertung der erbrachten Leistung eines Mitarbeiters im vergangenen Jahr anhand der für ihn festgelegten Zielvereinbarungen Synonyms: Performance Rating Abbreviations: Concept Type: category (object) Relationship Type: is related to Leistung (bewertet); is related to Zielvereinbarung (evaluiert); is a Mitarbeiter-beurteilung; is part of Performance Management Notes: Ist dies nur in der Halbjahres-/Endjahresevaluation im Rahmen des Performance Management Cycles anhand der gesetzten Ziele? Anhand der Ziele und Erwartungen an den Mitarbeiter (aus Zielvereinbarungsgespräch). Bewertet durch Feedback, Projektbewertungen (Kompetenzbeurteilung?!), Vergleich mit Kollegen Performancerating: Stufen von Low Performer bis High Performer bzw. sehr gut bis entwicklungsbedürftig Evtl. zusammenführen mit Kompetenzeinschätzung	Katharina	18.01.2015 17:03:00
Leistungsziel- verein-barung	Definition: Messbare (quantitative) Ziele hinsichtlich der Leistung eines Mitarbeiters Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Zielvereinbarung Notes: Hauptsächlich von Zielen des Unternehmens abgeleitet/heruntergebrochen von Zielerreichung kann auch die variable Vergütung abhängen werden von qualitativen Zielen unterstützt	Katharina	09.02.2015 17:44:00
Mentoring	Definition: Persönliche Betreuung durch erfahreneren Mitarbeiter. z.B. für Berufseinsteiger Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Entwicklungsmaßnahme Notes:	Katharina	17.02.2015 17:52:00
Messgröße	Definition: Einheit, in welcher das gesetzte Ziel gemessen wird Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Leistungszielvereinbarung Notes: Prozent, absolute Zahlen	Katharina	09.04.2015 08:49:00

Methoden- kompetenz	Definition: Kompetenz im Bezug auf analytisches, strukturiertes Denken, das Erkennen von Zusammen-hängen sowie Kreativität und Innovationsfähigkeit Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Kompetenz Notes:	Katharina	26.02.2015 12:00:00
Mitarbeiter	Definition: Eine im Unternehmen angestellte Person Synonyms: Abbreviations: Concept Type: category (actor) Relationship Type: is related to Leistung (erbringt); is related to Stelle (bekleidet); is related to Karrierepfad (verfolgt); affects An Mitarbeitergespräch teilnehmen; is related to Zielvereinbarung (hat); is related to Vorge-setzter (ist fachlich unterstellt); is related to Vorgesetz-ter (ist disziplinarisch unterstellt); is related to Kompe-tenz (besitzt); affects Entwicklungsmaßnahme vor-schlagen; is related to Entwicklungsprogramm (nimmt teil); is part of Kompetenzteam; is related to Unterneh-men (ist angestellt in); is part of Kompetenzteam; af-fects Teilnahme an Entwicklungsmaßnahme anmel-den; affects Entwicklungsmaßnahme belegen; affects Entwicklungsmaßnahme evaluieren; affects Feedback geben; is related to Entwicklungsprogramm (nimmt teil) Notes:	Katharina	06.02.2015 14:06:00
Mitarbeiter- beurteilung	Definition: Beurteilung eines Mitarbeiters anhand fest-gelegter Kriterien und dem von ihm gezeigten Verhal-ten Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is related to Evaluationsgespräch (wird besprochen in); is related to Mitarbeiter (beur-teilt); is related to Entwicklungsbedarf (ergibt) Notes:	Katharina	29.04.2015 10:16:00
Mitarbeiter- gespräch	Definition: Strukturiertes Gespräch zwischen Mitarbei-ter und Führungskraft Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is part of Performance Manage-ment Notes: müssen vereinbart (Termin, Ort) und dokumen-tiert werden sitzt Personaler mit drin? --> Meistens nur bei Eskalati-onsgefahr. Abhängig von Unternehmensgröße gestützt durch Leitfäden	Katharina	16.01.2015 10:56:00

Mitarbeiter-gespräch führen	Definition: Führungskraft führt ein Mitarbeitergespräch Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Mitarbeitergespräch Notes:	Katharina	22.05.2015 17:21:26
Mitarbeiter-gespräche nach-verfolgen	Definition: Personal verfolgt, wer ein Mitarbeitergespräch führen muss und wer schon ein Mitarbeitergespräch geführt hat, um evtl. die Führungskraft zu erinnern Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Mitarbeitergespräch Notes: Muss hinterlegt sein, wer wann Mitarbeitergespräche führen muss (Deadline oft gesamt für Zielvereinbarungsgespräche etc.) und Dokumentation des Gesprächs muss dem Personal verfügbar sein	Katharina	16.01.2015 14:56:00
Name	Definition: Vollständiger Vor- und Nachname eines Mitarbeiters Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Mitarbeiter Notes:	Katharina	29.04.2015 17:30:00
Personal-abteilung	Definition: Abteilung, die für das Personalwesen zuständig ist Synonyms: Abbreviations: Concept Type: category (actor) Relationship Type: is part of Unternehmen Notes:	Katharina	08.04.2015 18:32:00
Personalent-wicklungs-abteilung	Definition: Abteilung, die für die Personalentwicklung zuständig ist Synonyms: Abbreviations: Concept Type: category (actor) Relationship Type: is part of Personalabteilung; affects Entwicklungsmaßnahme genehmigen; affects Geplante Maßnahmen nachverfolgen; affects Mitarbeitergespräche nachverfolgen Notes:	Katharina	06.02.2015 14:42:00

Potentiellen Führungs-kräfte auswählen	Definition: Kandidaten für das Führungskräfteentwicklungsprogramm werden ausgewählt Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Führungskräfteentwicklungsprogramm Notes: äbhängig von Performace, vorgeschlagen durch Vorgesetzten, dann besprochen in Gremium	Katharina	07.02.2015 09:12:00
Präsenz-schulung	Definition: Schulungen mit physischer Anwesenheit Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Schulung Notes:	Katharina	21.01.2015 08:40:00
Qualifikationen	Definition: Höchster Abschluss den ein Mitarbeiter erworben hat und innegehabte Stellen in anderen Unternehmen Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Mitarbeiter Notes:	Katharina	29.04.2015 17:30:00
Return on Investment	Definition: Gewinn im Vergleich zum eingesetzten Kapital Synonyms: Abbreviations: ROI Concept Type: property Relationship Type: is property of Entwicklungsmaßnahme Notes:	Katharina	08.04.2015 18:21:00
Schulung	Definition: Veranstaltung zur Vermittlung von Inhalten Synonyms: Seminar, Kurs Abbreviations: Concept Type: category (event) Relationship Type: is part of Schulungskatalog; is a Entwicklungsmaßnahme Notes:	Katharina	12.03.2015 11:48:00
Schulungs-katalog	Definition: Auflistung aller Schulungen, die ein Unternehmen seinen Mitarbeitern standardmäßig anbietet Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: Notes: Evtl. geben Kompetenzteams Input	Katharina	07.02.2015 08:25:00

Selbstein-schätzung	Definition: Einschätzung des Verhaltens eines Mitarbeiters durch sich selbst Synonyms: Abbreviations: Concept Type: cateory (object) Relationship Type: is a Mitarbeiterbeurteilung Notes:	Katharina	06.03.2015 15:12:00
Selbst-studium	Definition: Mitarbeiter eignet sich selbst Wissen zu einem Thema an, z.B. durch Literaturrecherche Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Entwicklungsmaßnahme Notes:	Katharina	07.02.2015 09:34:00
Sozial-kompetenz	Definition: Kompetenz im Bezug auf Interaktionsfähigkeiten, wie Team-, Kooperations- und Kommunikationsfähigkeit, Konfliktlösung- und Verständigungsbereitschaft Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Kompetenz Notes: Fachlich übergreifende Kompetenzen, z.B. Visualisierung von Flip-Charts, Päsentation, Rhetorik, Stimmenschulung etc.	Katharina	16.01.2015 14:03:00
Stelle	Definition: Kleinste organisatorische Einheit einer Organisation, für die abgegrenzte Aufgaben und Anforderungen definiert sind Synonyms: Rolle Abbreviations: Concept Type: category (object) Relationship Type: is part of Karrierepfad Notes:	Katharina	05.03.2015 17:58:00
Teilnahme an Entwicklungs-maßnahme anmelden	Definition: Mitarbeiter meldet sich für eine Entwicklungsmaßnahme an Synonyms: Abbreviations: Concept Type: concept (activity) Relationship Type: affects Entwicklungsmaßnahme Notes:	Katharina	08.04.2015 12:06:00
Trainee-programm	Definition: Programm zur systematischen Einarbeitung und Integration von Hochschulabgängern ins Unternehmen Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is a Entwicklungsprogramm Notes:	Katharina	21.01.2015 08:38:00

Trainer	Definition: Person, die die Präsenzschulung leitet Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Präsenzschulung Notes:	Katharina	26.03.2015 16:42:00
Uebung	Definition: Möglichkeit für den Mitarbeiter, Kompeten-zen anzuwenden und zu verbessern Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Entwicklungsmaßnahme Notes:	Katharina	27.02.2015 14:36:00
Unternehmen	Definition: Wirtschaftliche selbstständige Organisati-onseinheit Synonyms: Firma Abbreviations: Concept Type: category (actor) Relationship Type: is associated with Jobcluster (hat); is related to Schulungskatalog (bietet); is related to Entwicklungsprogramm (bietet); is related to Unterneh-mensziel (hat) Notes:	Katharina	24.04.2015 10:23:00
Unterneh-mensziel	Definition: Qualitative und quantitative Ziele des Unter-nehmens Synonyms: Abbreviations: Concept Type: concept (object) Relationship Type: is related to Zielvereinbarung (be-einflusst) Notes: sollten dokumentiert sein!	Katharina	10.02.2015 09:40:00
Virtuelle Schu-lung	Definition: Computergestützte Schulung Synonyms: E-Learning Abbreviations: Concept Type: category (event) Relationship Type: is a Schulung Notes: keine Teilnahmebeschränkung	Katharina	18.01.2015 12:54:00
Zeitraum	Definition: Zeitraum in dem das gesetzte Ziel erreicht werden soll Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Zielvereinbarung Notes: meist ein Jahr	Katharina	09.04.2015 08:48:00

Zielerreichungsgrad	Definition: Grad, zu dem ein Ziel erreicht wurde Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Zielvereinbarung Notes:	Katharina	09.04.2015 08:29:00
Zielvereinbarung	Definition: Zwischen Mitarbeiter und Führungskraft vereinbarter Zustand, der in Zukunft von einem Mitarbeiter erreicht werden soll Synonyms: Abbreviations: Concept Type: category (object) Relationship Type: is related to Zielvereinbarungsgespräch (wird vereinbart in) Notes: Meist für ein Jahr. Ziele müssen zeitlich befristet sein	Katharina	19.01.2015 10:17:00
Zielvereinbarungsgespräch	Definition: Mitarbeitergespräch zu Beginn des (Fiskal)jahres, in dem Ziele des Unternehmens und des Mitarbeiters besprochen, abgeglichen und die Leistungs- und Entwicklungsziele für das kommende Jahr für einen Mitarbeiter vereinbart werden Synonyms: Abbreviations: Concept Type: category (event) Relationship Type: is a Mitarbeitergespräch Notes: Mitarbeiter und Vorgesetzter müssen Gespräch und Ziele abzeichnen. fällt meist mit Endjahresevaluation zusammen (ein Gespräch)	Katharina	16.01.2015 14:40:00
Zielwert	Definition: Angestrebter Wert des gesetzten Ziels in der genutzen Messgröße Synonyms: Abbreviations: Concept Type: property Relationship Type: is property of Leistungszielvereinbarung Notes: Endwert oder Differenz	Katharina	09.04.2015 08:50:00

5 References

Achouri, C. (2015). *Human Resources Management: Eine praxisbasierte Einführung* (2. Aufl.). Wiesbaden: Gabler Verlag.

Balzert, H. (2009). *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering* (3. Auflage). *SpringerLink : Bücher*. Heidelberg: Spektrum Akademischer Verlag.

Bazeley, P. (2013). *Qualitative data analysis: Practical strategies*. Los Angeles [i.e. Thousand Oaks, Calif.], London: SAGE Publications.

Becker, M. (2013). *Personalentwicklung: Bildung, Förderung und Organisationsentwicklung in Theorie und Praxis* (6th ed.). Stuttgart: Schäffer-Poeschel Verlag für Wirtschaft Steuern Recht GmbH.

Berntsson-Svensson, R., & Aurum, A. (2006). Successful software project and products. In G. H. Travassos, J. C. Maldonado, C. Wohlin, & E. Mendes (Eds.), *the 2006 ACM/IEEE international symposium* (pp. 144–153).

Boehm, B. W. (1981). *Software engineering economics*. *Prentice-Hall advances in computing science and technology series*. Englewood Cliffs, N.J.: Prentice-Hall.

Bolloju, N., & Leung, F. S. (2006). Assisting novice analysts in developing quality conceptual models with UML. *Communications of the ACM, 49*(7), 108–112. doi:10.1145/1139922.1139926

Browne, G. J., & Rogich, M. B. (2001). An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. *Journal of Management Information Systems, 17*(4), 223–249. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=4326066&site=ehost-live

Broy, M. (2013). Domain Modeling and Domain Engineering: Key Tasks in Requirements Engineering. In J. Münch & K. Schmid (Eds.), *Perspectives on the future of software engineering. Essays in honor of Dieter Rombach* (pp. 15–30). Berlin, New York: Springer.

Bryant, A., & Charmaz, K. (2010a). Grounded Theory in Historical Perspective: An Epistemological Account. In A. Bryant & K. Charmaz (Eds.), *The Sage handbook of grounded theory* (pp. 31–57). Los Angeles, Calif.: Sage.

Bryant, A., & Charmaz, K. (2010b). Introduction: Grounded Theory Research: Methods and Practices. In A. Bryant & K. Charmaz (Eds.), *The Sage handbook of grounded theory* (pp. 1–28). Los Angeles, Calif.: Sage.

Carvalho, L., Scott, L., & Jeffery, R. (2005). An exploratory study into the use of qualitative research methods in descriptive process modelling. *Information and Software Technology, 47*(2), 113–127. doi:10.1016/j.infsof.2004.06.005

Chakraborty, S., & Dehlinger, J. (2009). Applying the Grounded Theory Method to Derive Enterprise System Requirements. In *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing* (pp. 333–338).

Chakraborty, S., Rosenkranz, C., & Dehlinger, J. (2015). Getting to the Shalls. *ACM Transactions on Management Information Systems, 5*(3), 1–30. doi:10.1145/2629351

Charmaz, K. (2014). *Constructing grounded theory* (2nd ed.). *Introducing qualitative methods*. London, Thousand Oaks, Calif.: Sage.

Corbin, J. (2008). *Basics of qualitative research* (3rd ed.). Los Angeles, Calif. [u.a.]: Sage. Retrieved from http://www.opac.fau.de/InfoGuideClient.uersis/start.do?Login=wouer30&Query=540="978-1-4129-0644-9"

Corbin, J., & Strauss, A. (1990). Grounded Theory Research: Procedures, Canons, and Evaluative Criteria. *Qualitative Sociology, 13*(1), 3. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=10951749&site=ehost-live

Corbin, J., & Strauss, A. (1996). *Grounded theory: Grundlagen qualitativer Sozialforschung*. Weinheim: Beltz, PsychologieVerlagsUnion.

Corbin, J., & Strauss, A. (2014). Criteria for Evaluation. In A. Clarke & K. Charmaz (Eds.), *SAGE benchmarks in social research methods. Grounded theory and situational analysis* (pp. 213–221). London: Sage.

Cossick, K. L., Byrd, T. A., & Zmud, R. W. (1992). A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques. *MIS Quarterly, 16*(1), 117–138. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=9604010625&site=ehost-live

Cruzes, D. S., Vennesland, A., & Natvig, M. K. (2013). Empirical Evaluation of the Quality of Conceptual Models Based on User Perceptions: A Case Study in the Transport Domain. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, . . . J. C. Trujillo (Eds.), *Lecture Notes in Computer Science. Conceptual Modeling* (pp. 414–428). Berlin, Heidelberg: Springer Berlin Heidelberg.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM, 31*(11), 1268–1287. doi:10.1145/50087.50089

Daoust, N. (2012). *UML requirements modeling for business analysts* (1st ed.). Westfield, NJ: Technics Publications.

Dieste, O., Juristo, N., & Shull, F. (2008). Understanding the Customer: What Do We Know about Requirements Elicitation? *IEEE Software, 25*(2), 11–13. doi:10.1109/MS.2008.53

Easterbrook, S., & Nuseibeh, B. (2000). Requirements Engineering: A Roadmap. In A. Finkelstein (Ed.), *Proceedings of the Conference on The Future of Software Engineering* (pp. 35–46). New York, NY: ACM.

Ebert, C. (2012). *Systematisches Requirements Engineering: Anforderungen ermitteln, spezifizieren, analysieren und verwalten*. Heidelberg: Dpunkt.verlag.

Gibson, B., & Hartman, J. (2014). *Rediscovering grounded theory*. Los Angeles: Sage.

Glaser, B. G. (1978). *Theoretical sensitivity*. *Advances in the methodology of grounded theory*. Mill Valley, Calif.: Soc. Pr.

Glaser, B. G., & Strauss, A. L. (1999). *The discovery of grounded theory: Strategies for qualitative research*. New York: Aldine de Gruyter.

Gotel, O., & Finkelstein, A. (1996). Revisiting requirements production. *Software Engineering Journal*, (3), 166–182.

Grady, R. B. (1999). An economic release decision model: Insights into software project management. In *Proceedings of the Application of Software Measurement Conference* (pp. 227–239).

Halaweh, M. (2012a). Application Of Grounded Theory Method In Information Systems Research: Methodological And Practical Issues. *The Review of Business Information Systems (Online), 16*(1), 27–34. Retrieved from http://search.proquest.com/docview/1418721905?accountid=10755

Halaweh, M. (2012b). Using Grounded Theory as a Method for System Requirements Analysis. *Journal of Information Systems and Technology Management : JISTEM, 9*(1), 23–38. Retrieved from http://search.proquest.com/docview/1037355330?accountid=10755

Hamill, M., & Goseva-Popstojanova, K. (2009). Common Trends in Software Fault and Failure Data. *IEEE Transactions on Software Engineering, 35*(4), 484–496. doi:10.1109/TSE.2009.3

Hickey, A. M., & Davis, A. M. (2003). Elicitation technique selection: how do experts do it? In *11th IEEE International Requirements Engineering Conference* (pp. 169–178).

Hoda, R., Noble, J., & Marshall, S. (2012). Developing a grounded theory to explain the practices of self-organizing agile teams. *Empirical Software Engineering, 17*(6), 609–639. doi:10.1007/s10664-011-9161-0

Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software, 18*(4), 58–66. doi:10.1109/MS.2001.936219

Holton, J. A. (2010). The Coding Process and Its Challenges. In A. Bryant & K. Charmaz (Eds.), *The Sage handbook of grounded theory* (pp. 265–289). Los Angeles, Calif.: Sage.

Hruschka, P. (2014). *Business Analysis und Requirements Engineering: Prozesse und Produkte nachhaltig verbessern*. München: Hanser.

Hughes, J., & Wood-Harper, T. (1999). Systems development as a research act. *Journal of Information Technology, 14*(1), 83–94. Retrieved from http://search.proquest.com/docview/216197597?accountid=10755

Ibáñez, M., Kugler, H.-J., & Rementeria, S. (1996). Has Europe learnt enough? *Journal of Systems Architecture, 42*(8), 583–590. doi:10.1016/S1383-7621(96)00043-4

Kecher, C. (2011). *UML 2: Das umfassende Handbuch ; [UML lernen und effektiv in Projekten anwenden ; alle Diagramme und Notationselemente im Überblick ; mit zahlreichen Praxisbeispielen in C# und Java ; aktuell zu UML 2.3 und 2.4]* (4., aktualisierte und erw. Aufl.). *Galileo Computing*. Bonn: Galileo Press.

Kelle, U. (2010). The Development of Categories: Different Approaches in Grounded Theory. In A. Bryant & K. Charmaz (Eds.), *The Sage handbook of grounded theory* (pp. 191–213). Los Angeles, Calif.: Sage.

King, N., & Horrocks, C. (2010). *Interviews in qualitative research*. Los Angeles: Sage.

Kleuker, S. (2013). *Grundkurs Software-Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten* (3., korr. und erw. Aufl.). *SpringerLink : Bücher*. Wiesbaden: Springer Vieweg.

Larman, C. (2010). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development* (3. ed., 13. printing). Upper Saddle River, NJ: Prentice Hall PTR.

Leffingwell, D. (1997). Calculating your return on investment from more effective requirements management. *American Programmer, 10*(4), 13–16.

McManus, J., & Wood-Harper, T. (2007). Understanding the Sources of Information Systems Project Failure. *Management Services, 51*(3), 38–43. Retrieved from http://search.proquest.com/docview/234265241?accountid=10755

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization, 17*(1), 2–26. doi:10.1016/j.infoandorg.2006.11.001

Mylopoulos, J., Chung, L., & Nixon, B. (1992). Representing and using nonfunctional requirements: a process-oriented approach. *IEEE Transactions on Software Engineering, 18*(6), 483–497. doi:10.1109/32.142871

Nohl, A.-M. (2013). Narrativ fundierte Interviews. In A.-M. Nohl (Ed.), *Interview und dokumentarische Methode* (pp. 13–26). Wiesbaden: VS Verlag für Sozialwissenschaften.

Object Management Group, Inc. (2013). *OMG Unified Mondeling Language: Version 2.5.*

Partsch, H. A. (2010). *Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme*. *EXamen.press*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Pidgeon, N. F., Turner, B. A., & Blockley, D. I. (1991). The use of Grounded theory for conceptual analysis in knowledge elicitation. *International Journal of Man-Machine Studies, 35*(2), 151–173. doi:10.1016/S0020-7373(05)80146-4

Pitts, M. G., & Browne, G. J. (2007). Improving requirements elicitation: an empirical investigation of procedural prompts. *Information Systems Journal, 17*(1), 89–110. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=23658698&site=ehost-live

Podeswa, H. (2010). *UML for the IT business analyst: A practical guide to object-oriented requirements gathering* (2nd ed.). Australia, United States: Course Technology/Cengage Learning.

Poels, G., Maes, A., Gailly, F., & Paemeleire, R. (2005). Measuring the Perceived Semantic Quality of Information Models. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, . . . H. C. Mayr (Eds.), *Lecture Notes in Computer Science. Perspectives in Conceptual Modeling* (pp. 376–385). Berlin, Heidelberg: Springer Berlin Heidelberg.

Pohl, K., & Rupp, C. (2011). *Requirements engineering fundamentals: A study guide for the Certified Professional for Requirements Engineering exam : foundation level, IREB compliant* (1st ed.). *Rocky Nook computing*. Santa Barbara, CA: Rocky Nook.

Robertson, S., & Robertson, J. (2006). *Mastering the requirements process* (2nd ed.). *ACM Press books*. Upper Saddle River, NJ: Addison-Wesley.

Rosenberg, D., & Stephens, M. (2007). *Use Case Driven Object Modeling with UMLTheory and Practice*. Berkeley, CA: Apress. Retrieved from https://books.google.de/books?id=fPwlCD5JtaMC

Rosenberg, D., & Scott, K. (2001). *Applying use case driven object modeling with UML: An anotated e-commerce example*. *Addison-Wesley Object technology series*. Boston: Addison-Wesley.

Rumpe, B. (2011). *Modellierung mit UML: Sprache, Konzepte und Methodik* (2. Aufl.). *Xpert.press*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Rupp, C., & Queins, S. (2012). *UML 2 glasklar: Praxiswissen für die UML-Modellierung* (4., aktualis. u. erw. Auflage). München: Carl Hanser Verlag.

Rupp, C., Simon, M., & Hocker, F. (2009). Requirements Engineering und Management. *HMD Praxis der Wirtschaftsinformatik, 46*(3), 94–103. doi:10.1007/BF03340367

Rupp, C., & SOPHISTen, d. (2014). *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil* (6., aktualisierte und erweiterte Auflage). München: Hanser, Carl.

Ryschka, J., Solga, M., & Mattenklott, A. (2011). *Praxishandbuch Personalentwicklung: Instrumente, Konzepte, Beispiele* (3., vollständig überarbeitete und erw. Aufl.). Wiesbaden: Gabler Verlag / Springer Fachmedien Wiesbaden.

Schmidt, A., & Kunzmann, C. (2006). Towards a Human Resource Development Ontology for Combining Competence Management and Technology-Enhanced Workplace Learning. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, . . . P. Herrero (Eds.), *Lecture Notes in Computer Science. On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops* (pp. 1078–1087). Berlin, Heidelberg: Springer Berlin Heidelberg.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems, 17*(4), 5–36. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=4326032&site=ehost-live

Seidl, M., Brandsteidl, M., Huemer, C., & Kappel, G. (2012). *UML @ Classroom: Eine Einführung in die objektorientierte Modellierung*. Heidelberg: Dpunkt.verlag.

Sharp, H., Finkelstein, A., & Galal, G. (1999). Stakeholder identification in the requirements engineering process. In *Tenth International Workshop on Database and Expert Systems Applications* (pp. 387–391).

The Standish Group International, Inc. (1995). *The CHAOS Report (1994).*

Thom, N., & Zaugg, R. J. (2008). *Moderne Personalentwicklung: Mitarbeiterpotenziale erkennen, entwickeln und fördern* (3., aktualis. Aufl.). Wiesbaden: Betriebswirtschaftlicher Verlag Gabler.

Thomas, K., Bandara, A. K., Price, B. A., & Nuseibeh, B. (2014). Distilling privacy requirements for mobile applications. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 871–882).

Titscher, S. (1998). *Methoden der Textanalyse: Leitfaden und Überblick*. Opladen [u.a.]: Westdeutscher Verlag.

Wazlawick, R. S. (2014). *Object-oriented analysis and design for information systems: Modeling with UML, OCL, and IFML*. Amsterdam [u.a.]: Elsevier, Morgan Kaufmann.

Wiegers, K. E. (2003). *Software requirements: Practical techniques for gathering and managing requirements throughout the product development cycle* (2nd ed.). Redmond, Wash.: Microsoft Press.