

## **Considering the Knowledge Factor in Agile Software Development**

**Claudia Müller**

(University of Potsdam, Germany  
claudia.mueller@uni-potsdam.de)

**Julian Bahrs**

(University of Potsdam, Germany  
julian.bahrs@uni-potsdam.de)

**Norbert Gronau**

(University of Potsdam, Germany  
norbert.gronau@uni-potsdam.de)

**Abstract:** Agile software engineering methods provide a mean for flexible software development. However, these agile methods oppose direct steering and control as well as streamlining for process efficiency. In this contribution such agile processes of software engineering are analysed by applying a method known for its ability to encompass knowledge intensive business processes: the KMDL® (Knowledge Modeling and Description Language). Based on models created, potentials and actions leading towards improvements are derived. Further the KMDL® method as well as a case study of its application is introduced.

**Keywords:** Software Engineering, Knowledge Management, Modelling Method, Model Validation and Analysis

**Categories:** D.2, D.2.9, H.3.0, I.2.4, I.6.4, J.4

### **1 Traditional and Agile Methods of Software Development**

With increasing levels of globalisation competition becomes harder, cycles of innovation are shortened and at the same time quality requirements increase. This turbulent environment causes change in companies to be a permanent condition [Andresen, 04]. All domains within a company are affected, especially business processes, structures, and information technology [Krüger, 98]. Regardless of the industry, companies react to this situation with continual improvements of their processes, amongst other things. Therefore, business process management is of increased importance for a companies' competitive advantage.

In the last decades processes of software development companies were affected by engineering methods. Software Engineering as an engineering discipline affects all aspects of software production [Sommerville, 01]. Basic core processes of software development process are analysis, design, implementation, and test, whereas project management, quality management, and configuration management are support processes. These single processes are united together to a complete, ideal software development process definition. At present various such approaches exist. These

approaches essentially differ in degree of detail. Basic principles in procedure are models like the Water Fall Model and the Spiral Model. But there are strategies, which supply a detailed procedure and concrete guidelines for the project execution, like the Rational Unified Process [IBM, 05] and the V-Model XT [BMI, 05]. Altogether it is assessable that practical application shows: the more static and one-dimensional a method is, the less applicable it is within companies. This fact opposes the turbulent environment. In the last few years software development community was challenged by a new movement that addresses changes from a radically different perspective [Boehm, 05]. This new approach is best exemplified by its Agile Manifesto:

“[...] better ways of developing software [...]
   
Individuals and interactions over processes and tools
   
Working software over comprehensive documentation
   
Customer collaboration over contract negotiation
   
Responding to change over following a plan [...]” [AA, 01]

By introducing an agile method a company is able to produce software with less documentation under conditions of rapid change to achieve higher customer satisfaction. Agile methods encourage programmers to shed their heavyweight process changes [Boehm, 05]. It is possible to completely apply the Agile Manifesto in a company but this raises the following difficulties: How to control the process without a process definition? How to find out a contact person without knowing who is responsible for what? How to measure a project by continuously changing requirements? The truth is, as general, somewhere between Taylorism of traditionalists and the “hacking mentality” of agilists [Boehm, 05].

The distinctive specialty of software development compared to traditional industries, such as mechanical engineering, is the product. Software as a product incorporates a high degree of knowledge of software developer. This knowledge is particulate very abstract that means it does not follow a guideline, or clear formula. Within mechanical engineering in contrast a developer uses for example physical formulas, process instruction, or values of mechanical strength. Therefore, the process of creating software is knowledge intensive. This contribution displays how improvement of software development processes is possible based on a knowledge management method - KMDL® (Knowledge Modeling and Description Language) [KMDL, 05].

### **1.1 Conditions for Applying the Method**

KMDL® is a method of process oriented knowledge management. Process oriented knowledge management contributes to operational added value by supporting, improving, and enhancing operational business processes [Remus, 02a]. The centres of interest are knowledge intensive business processes. These processes can be classified by process oriented, employee-oriented, and resource-oriented criteria. For identifying knowledge intensive processes properties such as high complexity, weak degree of structure, communication-orientation, and high leeway for decision-making are indicators. In the area of software development processes possess these criteria very often. Especially in small and medium-sized businesses (SMB) there is generally

no consistent software development process implemented. This is a further reason for appearance of these criteria in these processes.

These process characteristics are the reason for assumptions required to model knowledge work in this context. In this contribution KMDL® is applied to record real instances of a process. The aim is not to document processes for example for an organisation handbook or to gather instances for example to deduce a reference process. In this case KMDL® is used for modelling past based instances. It is a detailed modelling of not repeatable, unique process instances. The modelling of processes is a workaround to achieve a better understanding of basic problems within the company. The generated models were used to identify repetitive problems in the development processes. Furthermore, KMDL® was used for person related modelling. The idea was to identify knowledge brokers or knowledge keepers within the company. The target of all these efforts was the conception of knowledge activities within software development processes. Of course, like other process modelling methods, the KMDL® was applied as a communication instrument and a description concept.

## **1.2 Proceeding in this Contribution**

In the first part of this contribution the concept of KMDL® (Knowledge Modeling and Description Language) is explained. Thereafter, the syntax of modelling technique is introduced. In order to apply the modelling method in real consulting projects the KMDL® procedural model is presented. This procedural model has been validated in practice. To support modelling and analysis work of consultants the K-Modeler and its most important features are illustrated.

The second part of this contribution consists of a case study to demonstrate the application of modelling technique and analysis of recorded models. Three analysis concepts are introduced – reports, process patterns, and analysis views (see section 3.4). Based on the KMDL® procedural model main topics within the conducted project are clarified.

In the third part results of conducted project, which are currently examined in a company, are discussed. This contribution concludes with further considerations for application the modelling method.

## **2 Brief Introduction to KMDL®**

The KMDL® (Knowledge Modeling and Description Language) method consists of the concept, the description language and the procedural model to apply the method in a consulting project. Within these sections underlying theory and basic semantic of the modelling method is explained. The description language depicts the syntax of language for example objects, their relations, and possible values. Further explanations of the modelling method are available in [Gronau, 04a] and [Gronau, 04b].

## **2.1 KMDL® Concept**

The modelling technique KMDL® provides a framework for modelling knowledge intensive business processes or knowledge intensive parts of conventional business processes. KMDL® enables modelling of these processes as a sequence of tasks and, additionally, knowledge conversion and flows in and between them.

Supplementary to process-oriented view, knowledge generated and used within processes is visualised. Consequently, a difference between explicit and tacit knowledge is made [Polanyi, 58]: Tacit knowledge is personal, context specific and difficult to communicate. In contrast it is possible to hand over explicit knowledge in a formal and systematic language [Nonaka, 95]. Along with this, information used and generated within a knowledge intensive business process is educible. Information is understood as a flow of messages, while knowledge is generally created by this very flow of information, through anchoring it in beliefs and commitments of its holder. This understanding emphasizes that knowledge is essentially related to human action [Nonaka, 95].

In KMDL® process models, information and explicit knowledge are represented as information objects. Tacit knowledge used within the process is represented by knowledge objects. Both types should be examined together because their interactions create new or extend existing knowledge. This interdependency is referred to by Nonaka und Takeuchi as knowledge conversion [Nonaka, 95]. Their model, which differentiates in internalisation, externalisation, socialisation, and combination, proposes a logical framework. The framework can be used to analyse tacit and explicit knowledge as well as conversions between those types of knowledge and therefore the creation of knowledge and the conditions and requirements for conversion to happen. It serves as the basic framework for modelling a dynamic process of knowledge creation within the authors' approach.

Internalization is the conversion of explicit knowledge into tacit knowledge. It is closely related to learning-by-doing. Experiences gained through socialization, externalization or combination are internalized and integrated into one's own knowledge framework. By this, experiences can create know-how or mental models and, according to this very important knowledge assets.

Externalization is the conversion from tacit to explicit knowledge. By using metaphors, analogies or models one can express parts of one's tacit knowledge in a manner which can be understood by others. It is the essence of tacit knowledge which can then be handed over in a written form. Yet it can be very difficult to externalize tacit knowledge. Often it is simply impossible.

Socialization is a conversion from tacit knowledge of one person to tacit knowledge of a different person. Often it is done by sharing experience: Just like apprentices of a craftsman learn their skills by observation, a knowledge-worker can learn needed or required abilities through on-the-job training. Socialization does not necessarily require speaking or writing a single word.

Combination is the conversion from explicit to explicit knowledge. Different kinds of explicit knowledge can be combined through media such as telephone, mail, word processing, or by reconfiguring, categorizing and adding new information and context to the knowledge.

According to the authors, the creation of organizational knowledge needs all types of knowledge conversion. To stimulate the process of knowledge creation in a company knowledge management plays an important role.

Accordingly, the modelling of the used and generated information and knowledge objects and the knowledge conversions enriches the sequential description of knowledge intensive business processes. The modelling aims at identifying process improvements but also at placing knowledge management activities where they directly generate value, for which the captured knowledge conversion ... can serve as important indicators.

### 2.2 KMDL® Description Language

Tasks are the basic framework of business process models. Sequences of tasks determine the logical structure of the process. A task is defined as an atomic transfer from input to output, represented by information objects. These objects are connected through the flow of information. Tasks are executed by roles. A role integrates textual or organisational related activities of a function. A function is realized by a person (process actor) or a group of persons.

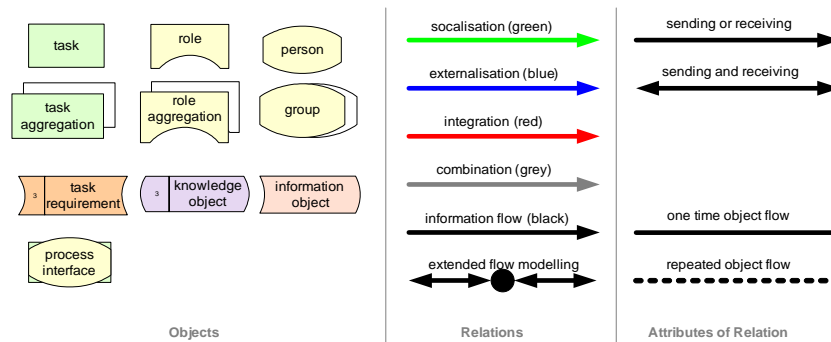


Figure 1: Objects and Relations

For each role, specific requirements (knowledge) are necessary to execute a specific task in the desired manner and to generate required output using specified input. A person who performs a task is assigned to a role. In addition to information, knowledge is required to execute a task. This knowledge is modelled as knowledge object. Each knowledge object is linked to a person and is personal. Each knowledge object must have a reference to a knowledge descriptor for describing which part of a knowledge domain is covered in which quality. Every used and needed tacit capability is represented by a knowledge object. Attributes are used for further description of the respective objects. For instance each single knowledge and information object is assigned to a specific topic through the attribute “knowledge domain”. This attribute enables hierarchical assignment of knowledge and information objects and hence describing used explicit and tacit knowledge within a considered process.

In addition, the description language allows visualising the flow of knowledge. A socialisation is represented as a directed relation from a knowledge object of one person to a knowledge object of another person. A conversion is either a one time event (solid line) or repeated (dashed line). The externalisation is modelled through the connection of at least one knowledge object with an information object. The association of at least two information objects and the generation of a new information object are called combination. The internalisation starts with an information object and ends with a knowledge object. Figure 1 shows the graphical representation of KMDL® objects and relations.

### 2.3 KMDL® Procedural Model

A procedural model (see Figure 2) has been developed to support applying the modelling technique during knowledge management projects [Gronau, 04c]. Each phase supplies various tools and methods for consultants. Basically the procedural model consists of seven phases. During project acquisition it has to be validated whether the KMDL® is a sufficient concept to fulfil the expectation of the potential project partner. Applying KMDL® is appropriate if defined conditions are met.

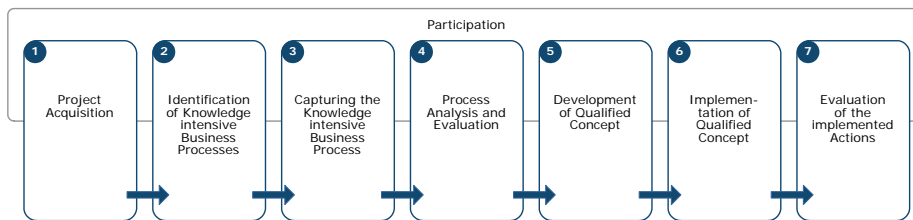


Figure 2: Phases of the Procedural Model

After the first phase of project acquisition in the second phase, project definition, a workshop is recommended. Here, the object of investigation has to be identified and defined. The consultant should have a sufficient understanding of general conditions in the company. Within the next phase capturing of specified processes is performed. For this phase a sub-model which consists of three steps is supplied (see section 3.2). Process analysis and evaluation serves the identification of weak spots within the process models. Based on these weak spots potentials and optional steps to assess these potentials are deduced. It is possible to define organisational, technical and personal (regarding the motivation) actions. These are combined in phase five to a sustainable qualified concept. The implementation of the qualified concept challenges the project partner. For example, if there is a need to implement technical actions, this phase may contain an entire software development project or a software selection process. Such projects are complex by themselves. This is why phase seven is essential: An evaluation of actions taken in which the project goals are compared to project results.

## **2.4 Modelling and Analysis Tool: K-Modeler**

The software tool K-Modeler [K-Modeler 05] supports consultants during modelling, visualisation, analysis, and evaluation of captured knowledge intensive business processes. The current version is released in the open-source development environment Eclipse [Eclipse 05]. Eclipse is only the core, therefore further required plug-ins can be integrated with the respective functionality. Both, eclipse and plug-ins, are based on the development language Java [Java, 05]. The K-Modeler user interface is clearly arranged, easy to use, and individual customisations are possible. KMDL® based process models are captured and changed in the model editor. KMDL® objects and relations are available as template. By utilizing drag-and-drop functions they can be moved from the object palette in the model area. Editing of object attributes is carried out by either selecting the respective object during modelling through a pop-up menu or by using an attribute window. Navigation within complex process models is supported through the process overview window. The navigator enables switching between different process models, which are related via the object “process interface”. Wizards support users in modelling as well as analysis and evaluation of process models. The evaluation component generates individual definable reports. These can be saved as HTML or XML documents.

## **3 Settings of the Project Preparation**

The case introduced in this contribution, describes the analysis of knowledge intensive processes followed by the definition of work improvements in a software engineering company. The following report of this project is structured in analogy to the procedural model.

### **3.1 Project Acquisition**

The company looked at in this case has about 25 employees in software development department. The project is initiated as part of the scientific project M-WISE [M-WISE 05] targeting at validating the application of KMDL® in software engineering industry. In addition, the project sets up to achieve sustainable results by applying knowledge management to the domain of software engineering. It is the goal of the project to identify strength and weaknesses in knowledge processing in this company. Further, knowledge management activities and/or changes in software engineering processes which based on findings of the process analysis should be suggested. These goals were discussed and defined at the kick-off meeting, which is arranged as a workshop on KMDL®. In this workshop fundamental concepts of KMDL® are explained to provide an overall understanding of the methodology and project steps among the participating employees.

### **3.2 Identifying knowledge intensive business processes**

In order to isolate knowledge intensive processes that are analysed in this project a predefined list of criteria of knowledge intensive processes was used [Remus, 02b]. Similar to product development in other industries, a high number of processes involving knowledge work were identified. This involved requirement elicitation and

management, design, and implementation, both for software products and project based customer specific software solutions. In order to narrow down the number of processes relevant for further analysis, typical work procedures were discussed. It turned out, that in this software company documentation was not highest priority. Rather, following almost extreme programming procedures, most tasks were tackled in teams involving informal communication. This lead to problems with requirement changes at a later project stage, as no means of formal orientation is provided.

In this contribution changes of requirements after the functional specification are labelled as Emerging Functions (EF). EF appear for a number of reasons during software projects at various points in time during project execution, especially during the implementation phase. Solving challenges EF impose is often critical for project success. Reasons for EF can be divided into two main categories: shortcomings in functional specification and changes necessary caused by the dynamic environment. Reasons for shortcomings of functional specification are enormous complexity of software projects and involve, but are not limited to, incomplete knowledge of customer requirements and processes. Turbulences in environment may occur with all partners involved in software development projects. For software supplier these may involve rapidly developing new technologies and software engineering concepts as well as organisational or procedural changes. In case of a client, reasons involve organisational and work procedure changes as well as incomplete knowledge or possible misunderstanding of functions of the software solution.

Dealing with EF is identified as knowledge intensive process by the following characteristics: their weak process structure, significant autonomy of process actors, multiple actors involved, process complexity and variability as well as undefined actions and process results.

Existing concepts for software engineering have addressed EF mostly as misconceptions during analysis or design phase and aim at reducing their numbers [Sommerville, 01]. However, flexibility and the ability to adapt to changed conditions are necessary for software companies. Therefore, dealing efficiently with EF is an important skill for competitive engineering of software and achieving customer satisfaction.

At the software company introduced in this case study workshops are arranged between customer and supplier regularly throughout the project. During these meetings the actual software solution is presented. It is not uncommon that EF result directly from these meetings.

It was agreed with the company to further analyse processes that deal with the design and implementation of EF.

### **3.3 Capturing Knowledge Intensive Processes**

Four separate instances of processes dealing with EF were singled out and further analysed. Though details of these processes are not discussed in this contribution a rough description of each instance is provided to provide a better project understanding:

1. In a customer project adjustments to an existing function for deleting data were required. This customer requirement was identified during a customer presentation of the prototype software solution. Based upon knowledge gained during presentation, an instance of an EF process was initialized and



a development task was assigned to an employee. In this case a workflow system was used to keep track of current project status.

2. This EF dealt with the design and implementation of a billing function, which is needed due to changes in a customer's billing procedure. This EF was initiated at a customer workshop, too. At first it was planned to address this EF in a customer specific solution. This was later changed, as a decision to redesign the standard product functions after a rather coincidental intervention of a product manager was taken.
3. This instance of an EF dealt with changing the platform of a database to a different operating system. This requirement was well known during functional specification but has not been addressed in detail, as it was known that a solution has been created before. Nevertheless, the employees were not involved in this process phase.
4. This EF was initiated due to licensing costs for a third party product and a customer request for improved usability of a document management function. Ultimately, the decision was made to replace the third party application by self-made functions for document management.

None of the instances were documented and most of them only had a name and just a few people involved. Details were captured based on interviews with actors of each instance. The method of interviews has its strengths in quickly capturing realistic working procedures and company structures, directly involving employees and the ability to reduce rejections to the project during the interview sessions. Its downsides are the necessity for qualified interviewers and the associated cost and time efforts. Prerequisites for successful interviews are an atmosphere of trust, precise questions and accurate documentation of the results [Krallmann, 02]. In this project partly structured interviews were used. In the beginning of an interview questions aiming at generating an overview on the instance are raised, e.g. through questions such as "Describe all activities of this instance in the order of their occurrence". In a later phase of the interview more specific questions are raised, such as "Describe knowledge required to fulfil this task". Step by step a precise documentation of the process as well as involved knowledge activities were captured. Further, involved business roles and knowledge objects were captured.

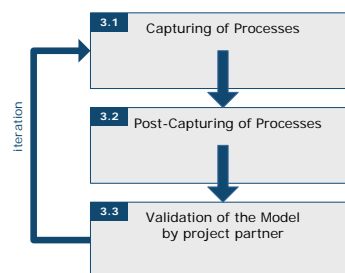


Figure 3: *Phase of Capturing Knowledge intensive Business Processes*

The procedure of capturing a process follows three iterative steps: Capturing the process in interviews, post-capturing of the process in which interview results are documented in text or KMDL® models. And last, the validation of the model by the involved process actors. This iterative procedure assures accurate and realistic process models.

During the first iteration, a textual documentation of the process is worked out. This eliminates methodical misunderstandings by the process actor as well as process shortcomings in the understanding of the instance by the modeller. In later iterations a KMDL® model is created. This can be supported by the K-Modeler (see section 2.4). K-Modeler assures the formal correctness of KMDL® models through real-time syntax checking. Finally, this phase of the project results in a verified and well documented KMDL® model of each of the described instances.

### 3.4 Process Analysis and Evaluation

The process analysis aims at identifying indicators of weaknesses within the captured processes. Currently, three separate methodical families for formal analysis of the KMDL® models have been developed: reports, process patterns and views. They are accomplished by free analysis performed by the process consultant.

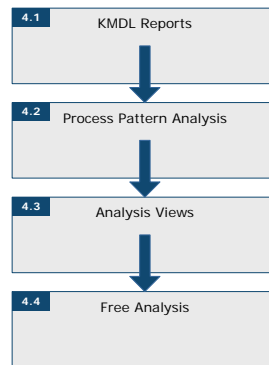


Figure 4: *Analysis of Knowledge intensive Business Processes*

Reports, which can be generated automatically by the KMDL® modelling tool K-Modeler, aggregate information on probabilities and number of occurrences within the process model. Currently three reports are implemented:

- KMDL® Competence Report: This report generates a complete profile of all knowledge objects associated with a process actor.
- KMDL® Task Coverage Report: This report matches knowledge requirements of tasks with knowledge objects associated to process actors. The discrepancies allow deducing conclusions e.g. for future training of employees.
- KMDL® Externalization Report: This report indicates which knowledge objects have been partially documented.

As an example, with the models of this case, the KMDL® Externalisation Report showed only little activity. This indicates that knowledge was often not documented.

The second approach for analysing processes is based on Process Patterns [Gronau, 04d]. These are a formal specification of KMDL® object constellations, which have been associated to a weakness before. Based on these predefined patterns that typically consist of five to 20 objects and edges even complex indications of weaknesses can be detected automatically with the help of K-Modeler. Five process pattern families have been defined so far [Bahrs, 05]. As an example the multi-step socialization pattern, part of the multi-step process pattern family, is introduced. This pattern describes a situation in which knowledge is passed on informally. This may lead to a “Chinese whisper” effect, where information gets lost or changed. The pattern is defined by a chain of socializations of knowledge objects representing the same knowledge domain among at least three process actors. The object constellation is marked red in Figure 5.

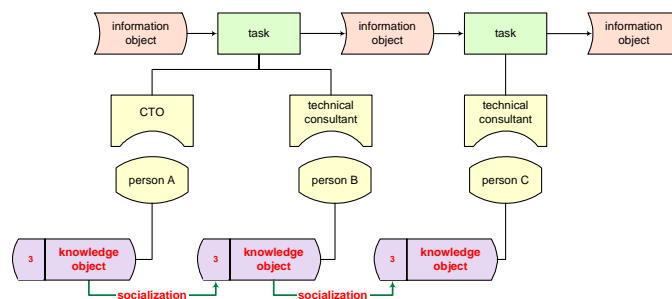


Figure 5: Multi Step Socialisation Pattern

Each pattern is further associated with a possible potential. In case of the described pattern this would be either to improve means of socialisation between the involved employees or to strengthen the documentation (codifying) of parts of knowledge. In the described scenario, this pattern occurs repeatedly among process instances. Later, an overall strategy which requires strengthening codifying parts of knowledge was chosen.

Analysis views allow for investigation of process models in different contexts and with different focus regarding special object constellations and therefore help to identify weaknesses and potentials that can not be recognized from process models directly. Currently three views exist: the conversions view, the communication view and the view of information and knowledge use.

The analysis views are introduced based on the example of a (for reasons of anonymity falsified mock-up) process model similar to instance one, in which adjustments to an existing deleting function were necessary. This customer requirement was identified during a customer presentation of the prototype software solution. The project manager of the delivering software company did not document this requirement. However, based upon the knowledge gained during the presentation, he initialized an instance of the EF process and assigned a development task to person one. A workflow system was used to keep track of the current project status. Among other status descriptions, the ones used here are “development assignment accepted”,

“assignment solved” and “test”. At any point in time a responsible person was assigned to the task. Figure 6 shows a cut-out of the KMDL model.

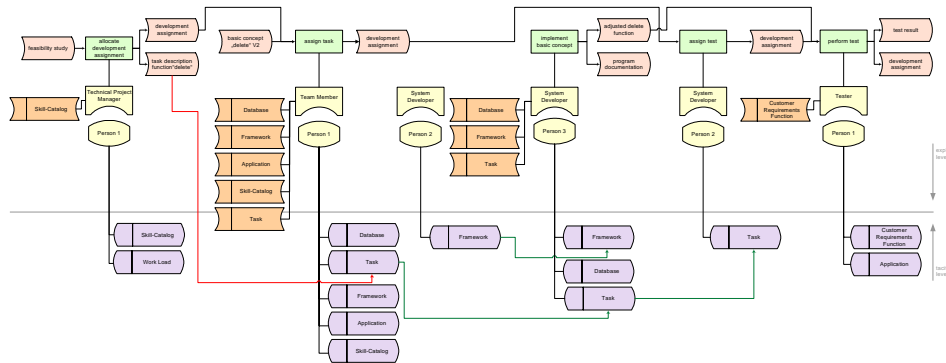


Figure 6: KMDL® Process Model of an EF instance

The view of knowledge conversions is based on the fact that the process model is dividable into a tacit level, which consists of the generated and used tacit knowledge and the explicit level, which contains the generated and used information (see Figure 6). Only one type of information objects participate in conversions. It is therefore necessary to differentiate between pure information and explicit knowledge (definition see section 2.1). Furthermore, each knowledge object consists of an expressible and an inexpressible part, since tacit knowledge includes cognitive as well as technical elements [Nonaka, 95]. Cognitive elements are based on the mental models, in which human beings create working models by creating and manipulating analogies in their minds. Technical elements include concrete know-how, crafts, and skills. The latter is only the expressible part, which participates in knowledge conversions. Figure 7 shows the conversion view of the process instance introduced.

The model displays only objects that are directly involved in a knowledge conversion. Therefore cognitive knowledge is not included. Because of this simplification (methodical knowledge is part of the conversion node) the nodes one and three are conversion nodes of combination. In the KMDL® description language, a task normally is a combination of different knowledge conversions. Due to the conversion view the task “adjust basic concept” can be specified as a conversion node of combinations. Furthermore, the task “assign task” consists of the conversion nodes three, four, and five. In the process model the nature of the task “adjusted delete function” was not visible. The conversion view, however, showed an externalisation. The objective of the analysis is on the one hand to improve the overview of the knowledge and information objects concerned and on the other hand to gain a better understanding of the concurrent processes within the task in order to improve the knowledge processes. Besides the investigation of tasks, the analysis of the communication channels is necessary. The nodes of conversion five and six were based on direct verbal communication whereas for node eight a telephone is used. The objective of the analysis is to improve or support the existing communication channels or to add additional ones.

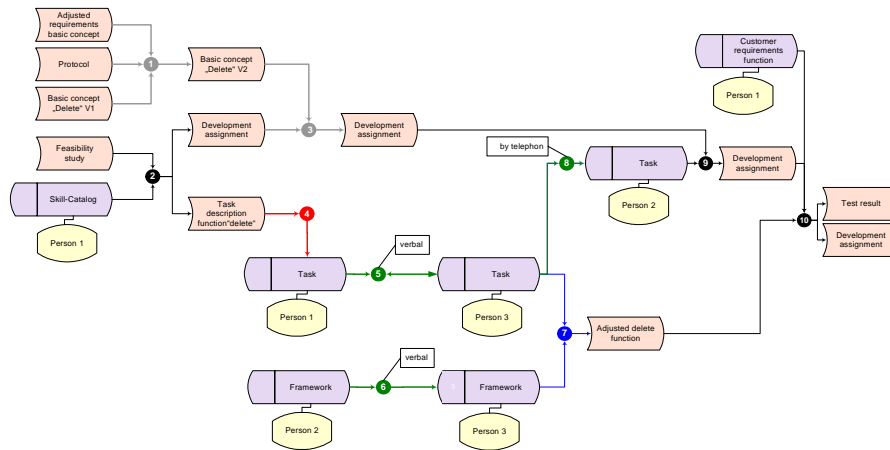


Figure 7: Conversion View of the EF process instance

The Communication View is used to analyze communication structure. Doing so helps to gain information on how to improve communication between participants, e.g. by improving the quality of communication channels. It also identifies isolated individuals and persons who are not part of the explicit level of the process (i.e. not involved in the official process), but who are needed for the success of the process.

In the graphical presentation of the communication flow the axis of abscissa represents the roles, the axis of ordinate the persons (see Figure 8). The displayed kind of communication is divided in planned (black, solid line) and unplanned (green, dashed line) communication. Unplanned communication is mostly based on socialisation in the tacit level of the process. This is in contrast with planned communication that is component of the explicit level. As soon as at least two persons are involved in the execution of a task or two persons are connected over an information flow, a planned communication takes place. The arrowheads show the direction and the trigger of the communication respectively.

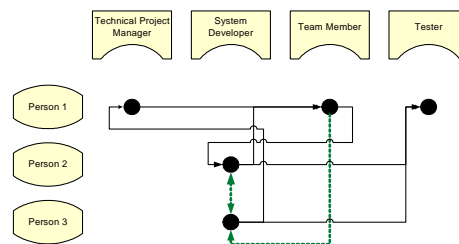


Figure 8: Communication View of the EF process instance

The View of Information and Knowledge Use aims at improving the usage of existing information and knowledge described in the KMDL® process model. It therefore matches task specific knowledge and information objects. Three degrees of

usage of this information and knowledge objects can be found in KMDL process models:

- Mandatory, the information or knowledge described by this object is required for fulfilling a task successfully
- Optional, using the information or knowledge object may enhance the results or efficiency of the task
- Unused, the information or knowledge described is not relevant for the task.

After generating this analysis view the current situation is presented by a view, consisting of all information and knowledge objects within scope listed on the axis of ordinate and each task within scope listed on the axis of abscissa. For each matching point a degree of usage is defined, as presented in Figure 9.

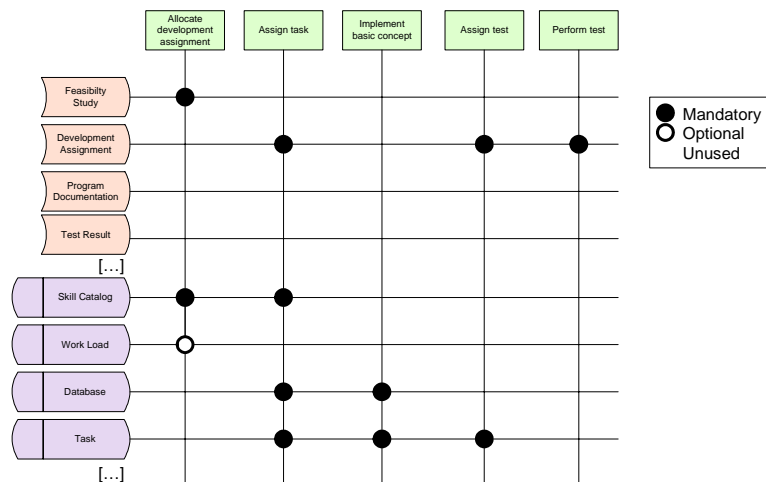


Figure 9: Knowledge and information use view (cut-out) of an EF process instance

In a second step possible benefits are identified of information and knowledge, e.g. increased task efficiency or improvements in quality of task results. Therefore, pairs of tasks and knowledge objects for all objects of the process model are examined. Through identifying the degree of relation between knowledge or information and the task, its value or relevance for the process is indicated. Further, suggestions for information flows or knowledge transfers can be derived.

In the example provided (see Figure 9) information and knowledge objects that were not used at all can be easily identified. This applies to the information object “program documentation” (which however may be used if analysing a different scope). An optional knowledge object, “workload”, is used to improve the outcome of the task “allocate development assignment”. It should be considered to make “workload” a task requirement or find other methods of making sure, that this knowledge is used and available for this task.

The analysis views confirmed previous indications of applied models of process instances: The communication structure view shows that important knowledge is

passed on between employees in an informal manner while using various different communication channels.

In addition to these formal approaches the assessment of additional weaknesses and potentials is performed by the process consultant of the project. The strength of this analysis is the integration of a greater background (e.g. knowledge gained during the interviews) and to overcome the limitations of context represented in models in general. Also, limitations of the method were identified and later used for enhancement of KMDL® and its formal analysis approaches (see section 4). In this case, it turned out that design documents and methods differ throughout the instances. Additionally, in instance two, an important potential was discovered while dealing with the customization of the billing function: during implementing the EF function as customer specific solution the decision was made to enhance the general product with the features of the function.

In this case study a total number of 18 weaknesses and potentials were identified, of which a few have been introduced for demonstration purposes. Additional weaknesses not mentioned include but are not limited to redundancy of work and unnecessary distribution of information.

### **3.5 Development of the Qualified Concept**

Each weakness bears a potential for improvement. In order to assess this potential necessary steps were derived individually for each potential. In a second step similar measures were grouped. Finally three central actions were suggested to the software company. Their relevance for the process was assessed by matching the potentials to the suggested measure. A summary of these arrangements is given below:

- Increasing the level and quality of documentation. Throughout the process this will provide improved orientation within EF, and even more important, after an instance of EF is implemented. It is not the aim to reduce the level of successful informal knowledge transfer. Rather the goal addressed here is to ensure a level of documentation, which allows for a reuse of code and already solved problems (and with that work) which have been addressed before. Therefore each EF should be formally documented (e.g. reason, target, solution) together with employees involved. Predefined templates and quality guidelines for these documents need to be designed and used to achieve a sufficient documentation quality.
- The introduced four instances show individual task sequences. Moreover, they differ in software engineering methods applied. This is due to the flexibility informal knowledge and information processing offers and typical for knowledge intensive processes. However, a balance between flexibility and structuring needs to be found that provides a minimum of orientation. This is not only to increase process transparency to employees but also a prerequisite for designing or customizing a software tool, which is a goal of the project. Last, designing a process guideline can be a vehicle to introduce additional process steps such as the decision to implement a function resulting of an EF as product improvement.
- Implementing a formal skill management for skill based task delegation. It was found that in multiple instances employees were assigned tasks that (a) others have addressed in previous EF or (b) that are delegated to employees

that need to gain specific knowledge first. This could be addressed by incorporating training needs or other actions to support this knowledge transfer.

In respect to the original goal of the project, the introduction of knowledge management software, the common ground of the process instances needed to be addressed. This was done by creating abstract process schemes. This degree of abstractness was achieved by switching to wider concepts (e.g. EF design specification instead of design of calculation function or other specifics to an instance). Subsequently a process scheme that integrates all process paths found in the instances was derived. This was done by virtually blending the instance models into each other keeping their individual task sequences as alternatives. This was followed by an analysis of path frequencies. By doing so, exceptional paths and instance specific activities could be identified and removed from the schematic model. As a result major activities were identified.

Based on these activities, combined with the introduced suggestions for improvement requirements for a software system to support dealing with EF were defined in a workshop by the software company:

- shared access to one system (number of communication channels)
- consistent communication structures and media
- well defined procedures
- simultaneous acquisition of problem and solution of EF
- easy access to detailed information about each activity (task) and EF
- structuring of tasks
- clearly defined interfaces to external systems (e.g. customer relationship software)
- description of software components and employee's responsibilities

### **3.6 Implementation of the Qualified Concept**

At current various standard software solutions for bug tracking or issue tracking are evaluated against these criteria. Besides the listed requirements this software needs to be customizable regarding the ability to integrate the identified typical task sequences of the EF process instances. The solutions should offer routing and handling of EF related tasks in a flexible, but transparent manner. Further, the software is used to create a documentation of each EF, adding basic information during EF implementation on a semi automated basis (such as employees involved and activities performed). Last, this software has to be integrated into existing software development solutions as well as the existing customer relationship software.

### **3.7 Evaluation of the Implemented Process**

It is the goal to maintain the identified strength of the software company that was determined within this project: the exceptional ability to share knowledge informally, which is a prerequisite for dealing with the various challenges of EF in a flexible and appropriate manner. The solution envisioned here shifts the balance between documentation and informal action towards codified knowledge. As explained,



several spots of weakness and potentials indicated that a minimum of documentation provides necessary information and therefore is expected to lead to an overall process improvement. However, this assumption can only be ultimately validated after the project solution is implemented.

#### **4 Evaluation of the preliminary Project Results**

This section addresses the impact of the overall project on the software company's ability to deal with EF. Further, it is addressed how modelling and analysis of knowledge intensive processes with KMDL® is appropriate to improve software engineering tasks.

The bottom-up driven approach of modelling multiple instances of a knowledge intensive process has proven to be a working approach to assess details of procedures in their existing variety. After all, the described method to derive an abstract process model using a frequency analysis allows for a systematic identification of typical task procedures of knowledge intensive processes. With the given properties of knowledge intensive processes this already is an insight. In this case the interview technique has proven to be a reliable concept to assess knowledge related activities as well as to actually gain information on the informal level. In applying KMDL® the iterative modelling procedure has proven to be a valuable instrument to create a shared understanding of the process between interviewer and process actors. However, results from the interviews heavily depend on the level of trust. The level of detail obtained partially depends on the time between the instance and the interview. Finally, this method is rather time consuming.

Similar to other process modelling methods, KMDL® models contain only a fragment of reality. One of the major purposes of such methods is to help to clearly communicate the processes [Kagermann, 05]. In addition, even though KMDL® objects are clearly defined, process models may differ slightly depending on the modeller. With that said, possibilities of formal analysis approaches are generally limited. They are to be considered as a tool, but not as a fully automated process analysis. The informal analysis through experts from the company as well as through KMDL® consultants allows assessing context and background, which are not represented in the model. This analysis leads to less systematic recognition of weaknesses and potentials.

Based on the experience from the case described, alterations in the specification of the KMDL® method were derived (and verified in practical application), that include:

- Development of a form for structured interviews that can be customized to the requirements of a company or scenario
- Development of modelling guidelines as well as style guidelines to decrease dependency of the consultant
- Semantic structuring of knowledge objects and task requirements
- Specification of formal methods of analysis (as described)

Further, additional improvements of the KMDL® method are planned that enhance the 'fragment of reality' represented in the model. Other improvements are

concepts for generalization and specification as well as aggregation of objects [KMDL, 05].

Current subject of research is, amongst others, a methodology to perform a ranking of potentials as well as measures defined.

Regarding the software company significant findings have already been derived from the KMDL® project. These include gaining critical knowledge and a significantly increased level of transparency on the ability to deal with EF. At current, major strengths as well as indicators for corporate culture have been identified. In addition drawbacks of predominantly informal knowledge transfers are realized. This allows, even at this early project stage, to focus on critical aspects of EF implementation.

## 5 Further Considerations for Application

This contribution shows a path for improving software development processes by applying a process oriented knowledge modelling method. In order to assess and understand the weaknesses and to use the strengths of the company under consideration the KMDL® provides a useful framework.

Nevertheless, the remaining challenge is to support knowledge intensive processes properly. This is due to process flexibility and their uniqueness. In addition to a formal process definition there are informal processes in companies. These processes work independently, i.e. they are only affected peripherally by formal specifications of procedures. The communication of process participants, existing social networks, and the company's culture are important factors for the project success. Traditional modelling methods do not consider the specialities of human factor in the business process. For knowledge intensive processes the individual is only conditionally replaceable.

The KMDL® focuses on knowledge. Knowledge is always bound to a person. Therefore, modelling this knowledge also captures the informal processes. The process description does not stick to the organisational level. The individual level for knowledge creation and distribution should not be underestimated within the corporate knowledge strategy.

## References

[AA, 01] Agile Alliance: "Manifesto for Agile Software Development" <http://agilemanifesto.org/>.

[Abecker, 02] A. Abecker, K. Hinkelmann, H. Maus, H-J. Müller: Integrationspotenziale für Geschäftsprozesse und Wissensmanagement (in German), In: A. Abecker, K. Hinkelmann, H. Maus, H.J. Müller (eds.): Geschäftsprozessorientiertes Wissensmanagement (in German), Springer-Verlag, Berlin Heidelberg, New York, 2002, 1-22.

[Allweyer, 98] T. Allweyer. Knowledge Process Redesign, Saarbrücken, 1998.

[Andresen, 04] K. Andresen: Developing Adaptability in the domain of Information Technology. In Röck, H. (edt.): Perspectives in Business Informatics Research. Shaker Verlag 2004, 15-21.

- [Bahrs, 05] J. Bahrs; N. Gronau; C. Müller, S. Schmid: Modellierung von Wissenskonversionen entlang wissensintensiver Geschäftsprozesse (in German). BIT: Banking and Information Technology, 1, 2005, 21-31.
- [BMI, 05] Federal Ministry of Interior: Das V-Modell (in German). <http://www.kbst.bund.de/-279/V-Modell.htm>.
- [Boehm, 05] B. Boehm, R. Turner: Balancing Agility and Discipline. 2<sup>nd</sup> ed. Addison-Wesley Boston 2005.
- [Eclipse, 05] eclipse.org, 2005, <http://www.eclipse.org/>.
- [Gronau, 04a] N. Gronau, E. Weber: Management of Knowledge Intensive Business Processes. In: J. Desel, B. Pernici, M. Weske, (edt.): Business Process Management, Springer, Heidelberg, 2004.
- [Gronau, 04b] N. Gronau; E. Weber: Modeling of Knowledge Intensive Business Processes with the Declaration Language KMDL. In: Mehdi Khosrow-Pour (ed.): Innovations Through Information Technology, Proceedings of the 14th Information Resources Management Association International Conference, Idea Group Inc., 2004.
- [Gronau, 04c] N. Gronau, C. Müller, M. Usler: The KMDL Knowledge Management Approach: Integrating Knowledge Conversions and Business Process Modeling, In: D. Karagiannis, U. Reimer (eds.): Practical Aspects of Knowledge Management, Springer-Verlag, Berlin, Heidelberg, 2004, 1-11.
- [Gronau, 04d] N. Gronau, M. Usler: Antipattern zur Potenzial-Analyse mittels KMDL in wissensintensiven Prozessen im Software Engineering (in German). In: N. Gronau, B. Petkoff, T. Schildhauer (eds.): Wissensmanagement - Wandel, Wertschöpfung, Wachstum: Tagungsband zur KnowTech 2004, München, 18. - 19. Oktober 2004 im Rahmen der Systems 2004, Internationales Congress Center München, GITO-Verlag, Berlin, 2004, 233-246.
- [Heisig, 02] P. Heisig. GPO-WM - Methode und Werkzeuge zum geschäftsprozessorientierten Wissensmanagement (in German), In A. Abecker, K. Hinkelmann, H. Maus, H.J. Müller (edt.): Geschäftsprozessorientiertes Wissensmanagement (in German), Springer-Verlag, Berlin Heidelberg, New York, 2002, 47-64.
- [Heisig, 03] P. Heisig: Wissensmanagement in industriellen Geschäftsprozessen (in German). Industrie Management 3, Gito-Verlag, Berlin, 2003, 22-25.
- [IBM, 05] IBM: Rational Unified Process. <http://www-306.ibm.com/software/awdtools/rup/>.
- [Java, 05] Sun Microsystems, Inc.: Java Technology 2005 <http://java.sun.com/>.
- [K-Modeler] K-Modeler. <http://www.k-modeler.de>.
- [Kagermann, 05] H. Kagermann, P. Zencke. Die Renaissance des Geschäftsprozess-Managements – Von Modellierungswerkzeugen zur Prozessplattform für den Wandel (in German). In: IM Information Management & Consulting, spezial issue 2005, imc information multimedia communication AG, Saarbrücken, 2005, 6-12.
- [KMDL, 05] Knowledge Modeling and Description Language. <http://www.kmdl.de>
- [Krallmann, 02] H. Krallmann, H. Frank, N. Gronau: Systemanalyse im Unternehmen (in German). 4<sup>th</sup> ed. Oldenbourg Verlag München Wien 2002.
- [Krüger, 98] W. Krüger: Management permanenten Wandels (in German); In: H. Glaser, E.F. Schröder, A.V. Werder (ed.): Organisation im Wandel der Märkte (in German), Wiesbaden 1998, 227-249.

[M-WISE, 05] M-WISE 2005 <http://www.m-wise.de>.

[Nonaka, 95] I. Nonaka, H. Takeuchi: *The Knowledge-Creating Company. How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, New York, 1995.

[Polanyi, 58] M. Polanyi: *Personal Knowledge - Towards a Post-Critical Philosophy*, The University of Chicago Press, Chicago, 1958.

[Remus, 02a] U. Remus. *Integrierte Prozess- und Kommunikationsmodellierung zur Verbesserung von wissensintensiven Geschäftsprozessen* (in German), In A. Abecker (ed.): *Geschäftsprozessorientiertes Wissensmanagement* (in German), Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[Remus, 02b] U. Remus. *Process oriented knowledge management, Concepts and modeling*. PhD thesis, University of Regensburg, Germany, Regensburg, 2002.

[Sommerville, 01] I. Sommerville: *Softwareengineering*, 6<sup>th</sup> ed. Pearson München, 2001.