TUM

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

Software Engineering Process Metamodels

Ragna Steenweg, Marco Kuhrmann, Daniel Méndez Fernández

TUM-I1220

Technischer Bericht

Technische Universität München
Institut für Informatik

Software Engineering Process Metamodels

A Literature Review

Ragna Steenweg, Marco Kuhrmann, Daniel Méndez Fernández
Technische Universität München
Institut für Informatik, Software & Systems Engineering
Boltzmannstr. 3
85748 Garching, Germany
ragna.steenweg@tum.de, kuhrmann@in.tum.de, mendezfe@in.tum.de

Summary

Software processes help to structure and organize software projects. Since software projects are complex endeavors and continuously grow in terms of size, budget, and complexity, software processes are used to coordinate people and teams, to define interfaces in a multi-site project setting in global distributed development, and to provide a shared terminology and knowledge base. Since much process knowledge is available, appropriate tools are required to structure knowledge and to make it accessible. Metamodeling is an accepted technique to create structure and semantics and, finally, to allow for creating tools. However, metamodeling remains a frequently discussed topic in the area of software processes. There is a number approaches courting for the favor of the process users; ranging from small and situation-specific approaches, over vendor-based solutions/services, to generic process standards. The report at hands investigates Software Process Metamodels (SPMM) for the state-of-the-art, state-of-application, and tool support. The goal is to create a big picture of systematic software process engineering. We report on a comprehensive literature review for SPMMs (concrete metamodels, software processes that are built on an SPMM) and tool infrastructures to support process design, implementation, deployment, and management. We analyze the metamodels w.r.t. their appearance and their evolution, their acceptance, and how they are supported by tools.

This report includes all the results of the guided research project "*Metamodel-based Determination of Key Performance Indicators for Software Process Management & Improvement*".

Keywords

Sofware Engineering, Software Process, Software Process Metamodels, Literature Review

CR-Classification: D.2, D.2.9

Contents

List of Figures

List of Tables

1 Introduction

As software products are becoming more and more complex, it is necessary to organize the development process. To this end, we refer to frequently used software processes. Today, a lot of differnt software processes exist following different approaches, like rich and agile processes (e.g. RUP and Scrum), or artefact- and activity- based approaches (V-Modell XT and XP). Especially rich processes do not always meet individual needs of a project and consequently need to be tailored. To lighten the tailoring process, a structured representation of the process models concepts is necessary. This representation is done by metamodeling and this is why some of the rich software processes are based on metamodels, and there are also Software Process Metamodels as stand alone developments that provide the information to built new software processes. This report focuses on collecting the existing Software Process Metamodels (SPMM) and identifying the relevant ones.

1.1 Problem Statement

A lot of *Software Process Metamodels* (SPMM) and metamodel-based software processes can be found in the field of Software Engieering. However, not all of them are actually used for software development. Yet missing is the investigation of the important and relevant SPMMs that are used to create software processes for software development.

1.2 Contribution

We contribute an exploratory study on software process meta-models (SPMM). We conduct a literature survey to elaborate general/standardized and specific SPMM. After that the literature had been analyzed regarding to different categories.

1.3 Terminology used in this Report

The report at hands investigates the domain of software process metamodels (SPMM). Several terms are important and, therefore, need to be introduced:

Software Process. A *Software Process*[1] creates the environment for organizing, planing, and operating software projects. Usually, a software process is described in a blueprint style to allow for a systematic, repeatable, and measurable project organization and operation.

From a modeling perspective, a software process is represented as a model that consists of the following sub models:

Role model: A role model describes the roles—representing the capabilities and responsibilities of the involved personnel—that are necessary to operate a project.

1 Other frequently used terms: Software development process, software development process model, development process model, software process model.

Artifact model: The artifact model describes the artifacts being created in a project. Artifacts are (final) deliverables as well as tentative ones. A common and frequently used acronym is the term *work product*. The artifact model describes the artifact types, the structure of artifacts and, finally, dependencies among the artifacts.

Activity model: The activity model describes the methods used to create and finalize the artifacts. The activity model is usually not represented as an executable model, but as an informal description for the process users.

Process model: The process model forms the "executable" part of the process description, which can be used, for instance, as blueprint for the creation of project plans. The process model is also used to connect the process model's artifacts with planning elements to create, e.g., an order of artifact creation.

Process support model: The process support model contains all process-related elements that are essential or helpful the run a project. Such elements are for instance: norms and standards to be used, concrete methods to be applied, compliance-relevant norms, standards and regulations, or tools that should be used in a project.

Software processes can be designed, based on their conceptual focus, either *activity-oriented* or *artifact-oriented*. Activity orientation highlights the activity sub model and lays its focus on the description of activities. On the other hand, artifact orientation pinpoints the artifacts being produced in a project, often avoiding a detailed methodical embodiment that requires process users to select and apply their own methods. Those approaches are, however, not exclusive.

Software Process Metamodel. A *Software Process Metamodel* (SPMM) is a (formal) metamodel that defines the language in which a software process is created. Our notion of a metamodel complies with the one propagated by the UML [35, 37] whereby we consider a software process (model) to be an instance of a SPMM.

Software Process Life Cycle Model. The *Software Process Life Cycle Model* (SP-LCM) describes a framework in which the definition, implementation, evaluation, deployment and, finally, the overall management of a software process is modeled. Although there are some comprehensive approaches to systematically create a concrete software process (e.g., Münch et al. [33]), we use in this report a smaller, compact SP-LCM, which consists of the following phases:

1. Analyze
2. Conceptualize
3. Realize
4. Deploy

All phases are comprised into an overall management of a single process or a comprehensive software process line (SPL, [43, 27, 48]).

1.4 Related Work

In this section, we give a brief overview of related work. Bendraou et al. [6] several modeling techniques for software processes are compared, namely SPEM 1.1 and 2.0, DiNitto's approach, Chou's Approach, UML4SPM [5], and the PROMENADE Approach. The analysis addresses the key concepts and capabilities of the selected modeling approaches. A comparable study is done by Henderson-Seller and Gonzales-Prez [18], where OPEN/OPF, SPEM, OOSPICE, and the LiveNet approach were considered. Those contributions focus on the analysis of metamodels that can be used to describe software processes in general. They pay, however, not attention to the applicability/feasibility and the practical relevance.

Another study by Ruiz-Rube et al. [44] focuses on the feasibility of the SPEM metamodel and investigates the use of this metamodel in a mapping study. In this study, the authors investigate the use *and* the extension of SPEM and discuss more than hundred contributions.

Martínez-Ruiz et al. [30] also focus on SPEM. In contrast to the other contributions, they focus on the opportunities to model variability of software processes using SEPM, and present a case study.

Summarized, available studies focus either on general comparisons of basic concepts or on the evaluation of a particular SPMM. In this report, we focus on the analysis of the state-of-the-art in systematic process engineering using SPMMs in general.

1.5 Outline

The remainder of this report is strucutred as follows: In Sect. 2 we introduce the study design. In Sect. 3 we present the results of the study before concluding the report in Sect. 4. All detailed information regarding the collected data (data tables and numbers) can be found in the appendix.

2 Case Study Design

In this section, we present the study design. We discuss the research questions, the case selection, and the procedures for the data collection, the analyses, and for supporting the validity. This section is structured according to Runeson et al. [45].

Chapter Overview

2.1 Research Questions

This study aims at investigating software process metamodels (SPMM). The overall goal is to investigate which SPMMs are practically relevant. Therefore, we want to figure out what the current state-of-the-art of SPMM is including the evolution of SPMMs, tool support, practical use, and, finally, SPMM ecosystems for systematic software process design and management.

We formulate the following research questions:

RQ 1: Which software process metamodels do exist?

RQ 2: What is the acceptance of existing software process metamodels?

RQ 3: How are established software process metamodels technically supported?

Those research questions aim at identifying the state-of-the-art in systematic software process design and management. To answer RQ 1 we search and review the relevant literature. The results are used to build a "big picture" that contains SPMMs and their usage, and whether those SPMMs are based on a standard. To answer RQ 2, we perform an in-depth analysis of the gathered SPMMs focusing on their history/evolution. We conduct an in-depth research in which we analyze the roots of the particular SPMMs, create *family trees* to visualize standard SPMMs and their evolved versions, and, finally, variants that were created using standard SPMMs. To answers the third research question, we further analyze the data to investigate the technical support. Since we are interested in software process design, development, and management, we focus on tool support w.r.t. process authoring.

2.2 Case Selection

The case selection is opportunistic and is based on whether available approaches/contributions allow to answer following questions:

Q 1: Is a certain software process based on a metamodel? If yes, which one?
Q 2: Does the metamodel under consideration allow for tool-based software process design, development, and management?
Q 3: Is the metamodel under consideration a standard?
Q 4: What is the history of the metamodel under consideration?

Those questions need to be answerable, if a metamodel is included in the result set. In turn, a tool that is used to develop some kind of process, is explicitly excluded from our research to avoid vendor-specific tools that primarily *not* deal with widely accepted (industry) standards.

The criteria named above aim at identifying standards that are used to design software processes, which are represented as computable models and, therefore, support tool-based design, development, and management. SPMMs should be standardized and accessible—optimal case: as open source—to allow for broad application.

2.3 Data Collection Procedures

We opt an incremental approach for the data collection. The search is a combination of automated search strategies (as known from *systematic literature reviews* (SLR) according to Kitchenham et al. [24]) and "snow-balling" procedures. Due to the nature of the area of investigation, and the missing standardized terminology a pure SLR would be inefficient.

Therefore, we use SLR techniques to get initial result sets that will be filtered and used for an in-depth investigation.

To initialize the search, we use the following query:

metamodel ***or***
(metamodel **and** *software engineering)* ***or***
(metamodel **and** *development process)* ***or***
(metamodel **and** *software development process)*

The query has been used in the following databases:

- Web of Knowledge
- ACM Digital library
- IEEE Xplore
- Google/Google scholar

Having the initial set in place, we incrementally check the contained contributions, and, if a contribution deals with software processes and software process metamodels, we further check the reference sections of the papers. After this first screening, the initial result set is cleaned and contains only contributions relevant for further investigation (primary sources).

Each marked paper's reference section is then analyzed for further potentially relevant contributions[1] (snow-balling). If there is a relevant contribution in a reference list, the cleaned result set is checked, whether it already contains the newly considered paper. If the newly considered paper is not in the result set, it is appended to the list of relevant papers. The corresponding list item is attributed with *new*. The overall procedure is repeated until all list items in the result set are marked as *checked*.

From the collected papers, we extract the information shown in Table 2.1. Since we already expect the result set to be rather in-homogeneous, we do *not* expect all cells in the resulting spreadsheet to be properly filled. On the other hand, we expect more and manifold data for comprehensive standards and, thus, the corresponding cells of the spreadsheet may be too simple to catch all data. The spreadsheet is, however, the primary source for conducting the in-depth analyses.

2.4 Analysis Procedures

In the following, we describe our analysis procedure. To analyze the the gathered data the following analysis procedure is executed w.r.t. the research questions:

Step 1: Identify all metamodels from the collected data.

Step 2: Identify all metamodels that are computable.

Step 3: Investigate the evolution of the metamodel history.

Step 4: Create a time line from the contributions' publication dates.

Step 5: Analyze each paper for research type facets.

Step 6: Analyze each paper that deals with SPEM or one of its variants for use, deviation, or extension attributes.

Step 1 and step 2 are preparatory steps in which we screen the result set and investigate the contained metamodels as well as information w.r.t. computability and the history of the particular metamodels that is in particular investigated in step 3. In step 4, a time line for each of the identified metamodels is created. This time line will later be used to create

1 Beside of scientific papers, books and standards are included in the search when cited in the references of a paper.

Field	Description
Author	The authors of the paper
Title	The title of the paper
Reference	A reference link to the paper (data source, cite key, and so on)
Keywords	The key word list of the paper
Year_{C}	The publication year of the contribution
$\text{Year}_{\text{SPMM}}$	If the contribution deals with a certain SPMM, the version and the version's release date are stored in this field
Origin	The origin indicates, whether the paper is a scientific dissemination paper, an industrial or an inter-/national standard and, finally, who is the contributor of the standard
Business model	The contribution is analyzed for potential business models, e.g., an open and freely accessible standard is used, but the contribution itself deals with a commercial product. Such situations can be found, if tool-centric approaches are the matter of of contribution.
Paradigm	The process's and the SPMM's paradigm is analyzed, i.e. if the SPMM is activity- or artifact-oriented
History	If available, the history (in terms of version) of the process/the SPMM is analyzed
Variants	If available, variants and derivates are analyzed
Contribution facet	The papers are analyzed for their *contribution facets*: metamodel, model, methodology, tool, or technology
Research type facet	The papers are analyzed for there *research type facets*: evaluation research, solution proposal, position/opinion papers, and experience reports. Philosophical papers and validation research are not covered by our analyses.

Table 2.1: Data structure for the data collection (result set structure).

a complete "family tree" that plots the evolution of a certain metamodel with all variants and versions. In step 5, the result set is analyzed for research type facets, which are used to create—or at least to prepare—a mapping study. In step 6, finally, the study results by Ruiz-Rube [44] are analyzed and extended by our own data as our result set may contain additional contributions (see related work).

2.5 Validity Procedures

The aim of the study in this report is to conduct a first investigation of the state-of-the-art in SPMMs and their application in practice. As we rely on a literature review, we follow the validity procedures established to conduct a systematic SLR. We furthermore rely on researchers triangulation for the in-depth analysis performed to answer our research questions. Finally, as another validity procedure holds the combination of regular SLR techniques with snow-balling to overcome the problem of a blurry terminology given in respective area of investigation.

3 Study Results

In this section, we present the results of our work. This section is structured according to the research questions, which we introduced in the last section.

Chapter Overview

3.1 RQ 1: Which software process metamodels do exist?

In this section, we give a short overview over the software processes that met our selection criteria.

3.1.1 Metamodel-based Software Processes

This section contains a list of software processes that are designed and described using an SPMM. For each of the processes, we give a short description containing the purpose, the information w.r.t. the metamodel, the information w.r.t. the process's dissemination, and relevant references.

3.1.1.1 V-Modell XT

Purpose. The V-Modell XT is a comprehensive software process that supports a variety of project settings. It is a modular software process, which is focused on the artifacts being created in a project. Therefore, the V-Modell XT is a so-called *artifact-oriented* software process. The V-Modell XT that is considered here is a metamodel-based process since 2004/2005 when the former V-Modell 97 was completely revised.

Technology. Before the revision in 2004 the "classic" V-Modell was published as a book. After the revision, it has become a model described in XML. The metamodel [49] of the V-Modell XT is an XML schema.

Dissemination. The V-Modell XT is a standard software process, which is obligatory for the government authorities in Germany and their contractors. Therefore, the V-Modell XT is comparable to the Swiss *Hermes* [7] and the British *PRINCE2* [25]. Since the official standard is generic, a number of so-called V-Modell-Variants were created over the years. Especially the metamodel and the corresponding infrastructure, which was significantly updated in 2008/2009 allow for creating comprehensive process variants. The infrastructure of the process consists of a number of tools that allow for editing the process, managing the variants, publishing process documentation, and supports the initial tailoring (in terms of a project setup including the tailoring of the software process, creating consistent work product templates, and initial project plans).

References. The V-Modell XT is available online. The download package contains published process description, a set of pre-fabricated work product templates, and all sources that are required to perform own tailorings and process customization. The process is complemented with a number of documents describing installation procedures and hands-on examples. Furthermore, some books and articles are available that give an introduction into the V-Modell XT [13], introduce the customization process [28], and report on experiences.

3.1.1.2 Unified Process and its Derivates

In this section, we describe the *Unified Process* family that constitutes several software processes of which most famous ones are the *Rational Unified Process* and the free *Open Unified Process*.

Purpose. The Unified Process is an iterative and incremental software development process framework. It is use case and architecture driven. The evolution of the Unified Process starts in the 1960's as the Unified Process is based on the Ericsson approach. Over the years, many derivates appeared such as RUP—the Unified Process by Rational/IBM—or the OpenUP. The Unified Process offers an iterative and incremental development process.

Metamodel. Since 2000, the Unified Process is based on SPEM (cf. Sect. 3.1.2.2). The latest version of 2005, the Rational Unified Process 7.0, is based on SPEM 2.0 [36]. Most derivates, however, are based on SPEM 1.1.

Dissemination. The Unified Process is widely spread. IBM offers the Rational Unified Process (RUP) with the support of the Rational Unified Composer. This Composer helps to customize the RUP to the individual project needs. Besides there is a free version with the Open Unified Process. To customize it, there is a tool as well, named the Eclipse Process Framework (EPF [9]).

References. The Unified Process was published first in 1998 by Phillippe Kruchten [26]. Since the Unified Process was adopted for different purposes, a number of contributions were made, e.g., [3, 15]. Furthermore, since RUP is a widely implemented commercial product by IBM/Rational, a number of experience reports and variants is available that we do not further discuses in explicit.

3.1.1.3 Further Models

The aforementioned software processes are based on metamodels and therefore, allow for modeling, tailoring, and so on. From the perspective of a process engineer, those software processes are adequate points to start a software process improvement project. There are, however, more than the aforementioned software processes that are also based on "some kind of" metamodel. We distinguish software processes that are based on a metamodel, which is company-specific, and software processes that seem to be implicitly based on a metamodel, which is not formally described.

SE Book: The SE Book is a company-specific software process by T-Systems [8]. The SE Book is based on a formal metamodel that is—similar to the V-Modell XT—based on XML technologies. For confidentiality reasons not further information w.r.t. this software process can be given.

AE Modell: The AE Model is a company-specific software process that is used by some processes of the "Kreissparkasse". This process claims to be built on a metamodel on which, however, no information is available.

OEP: The OOSE Engineering Process [34] is a company-specific software process that was published as a book. This software process is rather detailed and uses UML to describe its basic concepts. Thus, the metamodel is only a supporting part of the process's documentation, the process is, however, not formally described by an explicit metamodel.

Furthermore, several other (company-specific) processes re-use and adopt structures defined by SPMMs. For instance, the software process of Capgemini TS (former sd&m) is based on a metamodel, which is aligned with a company-specific RUP.

Another example is the OPEN Process Specification [17]/Process Framework [12]. This generic process framework combines a metamodel and concrete processes. OPEN is available at: `http://www.opfro.org`, but seems not to be maintained anymore. It seems to be more likely that OPEN was a major input when creating the ISO 24744 standard [23]. Especially the metamodel of OPEN shows a strong relationship to the ISO 24744 metamodel (cf. Sect. 3.2.3).

3.1.2 Software Process Metamodels Overview

As a result of our investigation, the following (standardized) metamodels were found. In the following, we give a brief overview of those metamodels. We do not provide a detailed description, bur refer to the corresponding literature.

3.1.2.1 V-Modell XT Metamodel

The V-Modell XT metamodel is the basic infrastructure component to build processes in the V-Modell XT environment. Since an independent non-profit organization is responsible for the maintenance of the entire V-Modell XT infrastructure, this metamodel is (implicitly) standardized.

Purpose. This metamodel is based on the idea of software process lines (SPL [43]). The metamodel allows for creating modular processes that might be part of a process line. The modularity concepts of the V-Modell XT are also reflected by the modularity of single self-contained processes and allow for project-specific tailoring. Comprehensive process lines can be built based on a stable core model—a reference model—by creating so-called process variants. A process variant can reduce, extend, and alter a given reference model using formally define variability operations. A tool, finally, computes a concrete process variant by merging the reference model and the extension parts.

Technology. The V-Modell XT metamodel is represented by an XML schema. Instances of this schema (concrete XML files) represent each a single process variant. Consequently, reference processes and extension parts are stored separately and can, therefore, developed and maintained in an independent manner. The metamodel is accompanied by a comprehensive platform, which consists of editors, environments that perform merging and generation/publication tasks, and tools that support project managers to tailor a process at the beginning of a project in order to meet concrete project requirements w.r.t. certain process configurations. The basic philosophy of the V-Modell XT metamodel is to highlight process artifacts that will represent artifacts of the later software process.

Dissemination. The V-Modell XT, which is a reference implementation of the V-Modell XT framework, is a national standard in Germany. It's maintenance is, however, outsourced to an independent non-profit organization that also deals with the improvement of the metamodel and the underlying process infrastructure. There is a number of comprehensive variants of the V-Modell XT (see Sect. 3.2.1). Furthermore, other derivates of the V-Modell XT of which we have little or no detailed knowledge at all, are in the field.

References. The metamodel of the V-Modell XT is documented in a technical report [49] (in German) on a conceptual level (e.g., using UML models). Furthermore the metamodel itself is open and free accessible as a (partially) documented XML schema. In addition to those resources, the book [28] introduces the method how to adopt the V-Modell XT and how to create processes using the V-Modell XT framework.

3.1.2.2 Software & Systems Process Engineering Metamodel

The Software Process Engineering Metamodel (SPEM) is a standardized metamodel, which is created, published and maintained by the OMG [36]. Similar to the V-Modell XT SPEM is the basis for a comprehensive process framework in which are metamodel descriptions, methodical components, and tools are comprised.

Purpose. SPEM is a metamodel that aims at addressing processes in general and is, therefore, a rather complex and highly integrated SPMM. The basic goals of SPEM are modularity, separation of contents and (concrete) processes, and flexibility in terms of creating arbitrary processes. SPEM basically follows the activity-oriented design approach, but also allows for a variety of dependencies among the different process elements. Similar to the V-Modell XT, SPEM allows for tailoring software processes, but implements a different strategy.

Technology. SPEM is first and foremost a technical specification, which is published as a OMG specification [36]. However, SPEM is also specified in a formal manner represented by a set of technical models and metamodels[1] that can be used with the Eclipse environment. The data exchange format of SPEM is realized using the XMI specifications and, therefore, SPEM-based processes are usually represented as a set of interconnected XML files. Although SPEM is a "general" metamodel for designing arbitrary processes and also contains some artifact orientation concepts, it is focused on the activity-oriented process design paradigm.

SPEM-based processes can be implemented, e.g., using the free Eclipse Process Framework (EPF, Sect. A.2).

Dissemination. SPEM is a widely spread and accepted industry standard and, thus, many (scientific) contributions are available. Ruiz-Rube et al. [44] conducted a comprehensive study on SPEM and its usage. Summarized, more than 100 contributions are available in which experience reports on the application, new SPEM-based derivates, and SPEM extensions are discussed. Consequently, similar to the V-Modell XT a process family exists, which we show in Sect. 3.2.2. Some examples of SPEM-based software processes that were implemented using EPF can be found in Sect. A.2.

References. The metamodel as the source for all SPEM-related contributions is published by the OMG [36]. Also the technical parts are available as free downloads. An overview of the majority of SPEM derivates, variants, and extensions can be found in [44]. The report at hands adds some contributions to the result set of Ruez et al. that are discussed in Sec. B.

3.1.2.3 ISO 24744 (SEMDM)

The *Software Engineering Metamodel for Development Methodologies* (SEMDM [23]) is an ISO standard, which is based on ideas proposed by OPEN (see Sect. 3.1.2.4) and (Situational) Method Engineering. This standard is published by the International Organization for Standardization and is an international (industry) standard.

1 See the SPEM website `http://www.omg.org/spec/SPEM/2.0` for further information.

Purpose. SEMDM is a SPMM that addresses software process design in general. Its concepts are based on a set of core concepts, which have their origins in Method Engineering. One basic idea of SEMDM is *not* to strictly differentiate between a model and a corresponding instance. The process metamodel is based on "power type" pattern and on a set of so-called "Clabject" constructs. Consequently, SEMDM also addresses process engineering (in general, but with a focus on Software Engineering methods), but uses a different conceptual approach compared to SPEM or the V-Modell XT.

Technology. SEMDM is based on the UML. Since there is no technical implementation, no concrete and standardized data exchange formats are defined. Furthermore, as no data exchange is defined, no basic data model is defined that realizes the process language. SEMDM is in its current state the documentation of a standard. To the best of our knowledge, no technical (reference) implementation is available.

Dissemination. SEMDM is covered in only a few scientific contributions in which basic principles are explained, and some experiences are reported. There is, however, no self-contained and published process model available that is built on SEMDM. A detailed evaluation of the dissemination is, therefore, not possible.

References. SEMDM is a ISO standard [23]. In addition to the standard documentation, few contributions deal with SEMDM, i.e. [16].

3.1.2.4 OPEN Process Framework

The OPEN Process Specification [17]/Process Framework [12] is defined (partially) using UML. Furthermore, OPEN provides potential users with a structured web-based knowledge library and therefore, a reference "implementation" is available. However, to the best of our knowledge no practically relevant application of OPEN is documented. On the other hand, the OPEN metamodel has a strong relationship to the ISO 24744 metamodel. We therefore consider OPEN to be the predecessor of ISO 24744.

3.1.2.5 MetaME

Purpose. MetaME is a method, to build software engineering methods by combining product and process models [10]. MetaME is based on (Situational) Method Engineering [41] and provides a proposal of how to create Software Engineering methods. It follows the idea that one should build up the methods and not tailor a big process, so it is a bottom-up approach:

- 0: Define domain and discipline
- 1: Produce domain model of software engineering concepts
- 2: Select notations
- 3: Define artifact types
- 4: Define the software engineering process model
- 5: Select tools, techniques and utilities

Technology. MetaME is specified using UML.

Dissemination. MetaME is proposed, however, validation work is yet not available.

References. MetaME was published in 2010 by Gregor Engels and Stefan Sauer [10].

3.2 RQ 2: What is the acceptance of existing software process metamodels?

The "acceptance" of an SPMM is reflected by its use and, thus, available contributions regarding the concrete application, experiences reports, improvement proposals for certain metamodel releases, or variant creation. To get an overview, we perform an in-depth investigation in which we analyzed the identified SPMMs in order to create a set of *family trees* that show the evolution of a particular SPMM. For certain results, background information going beyond the analysis is given in footnotes.

3.2.1 V-Modell XT

The family tree of the German V-Modell XT is given in Fig. 3.1. Before 2004, the official release of the V-Modell 97 was published as a book. The first release that was built on a metamodel was the internal release 0.9[2], which was used for testing. The official release was done in 2005.

Starting with the official release, the V-Modell XT metamodel was also officially released so that tool vendors could use it for creating appropriate tools. Figure 3.1 shows the evolution of the metamodel on the left. The figure shows that the first update of the metamodel was done in 2006; the next update was done in 2009. After that a branch can be observed. In 2010 and 2012 there were two metamodel variants created for which holds:

$MM\ 1.3 \subseteq MM\ 1.3B \subseteq MM\ 1.3Z$

In parallel the original metamodel "MM 1.3" was further developed and adopted some improvements that were made in the branch. However, the latest version of the branch is not fully compatible with the latest metamodel version "MM 1.4".

A deeper look in the metamodels shows that there is no "real" version and configuration management at all for the metamodels. The metamodels underwent an evolution, but do not contain a version number. That's why we named the metamodel version in relation to the version number of the resulting reference model (e.g., "MM 1.3" means the metamodel that was used to build the version 1.3 of the reference process).

Although the metamodel is not a "first class citizen" on the first sight, a number of process versions an variants are available. Our research names (at least) seven variants that are built on the reference model, each containing a couple of Thousands of process elements. The family tree in Fig. 3.1 shows that the number of variants increased in 2009/2010. The reason was the metamodel "MM 1.3" that was built on principles of software process lines [43, 27, 48] and allows for easier creation of variants by using explicit variability mechanisms.

3.2.2 SPEM

Figure 3.2 shows the family tree of the SPEM metamodel. The figure also shows exemplarily assigned concrete processes and tools. The initial release of SPEM was done in 2001; an improvement to a version 1.1 was done in 2005. The release of 2005 was the

2 This release was used to evaluate the newly created process. Compared to a software project this version was the *Beta* release.

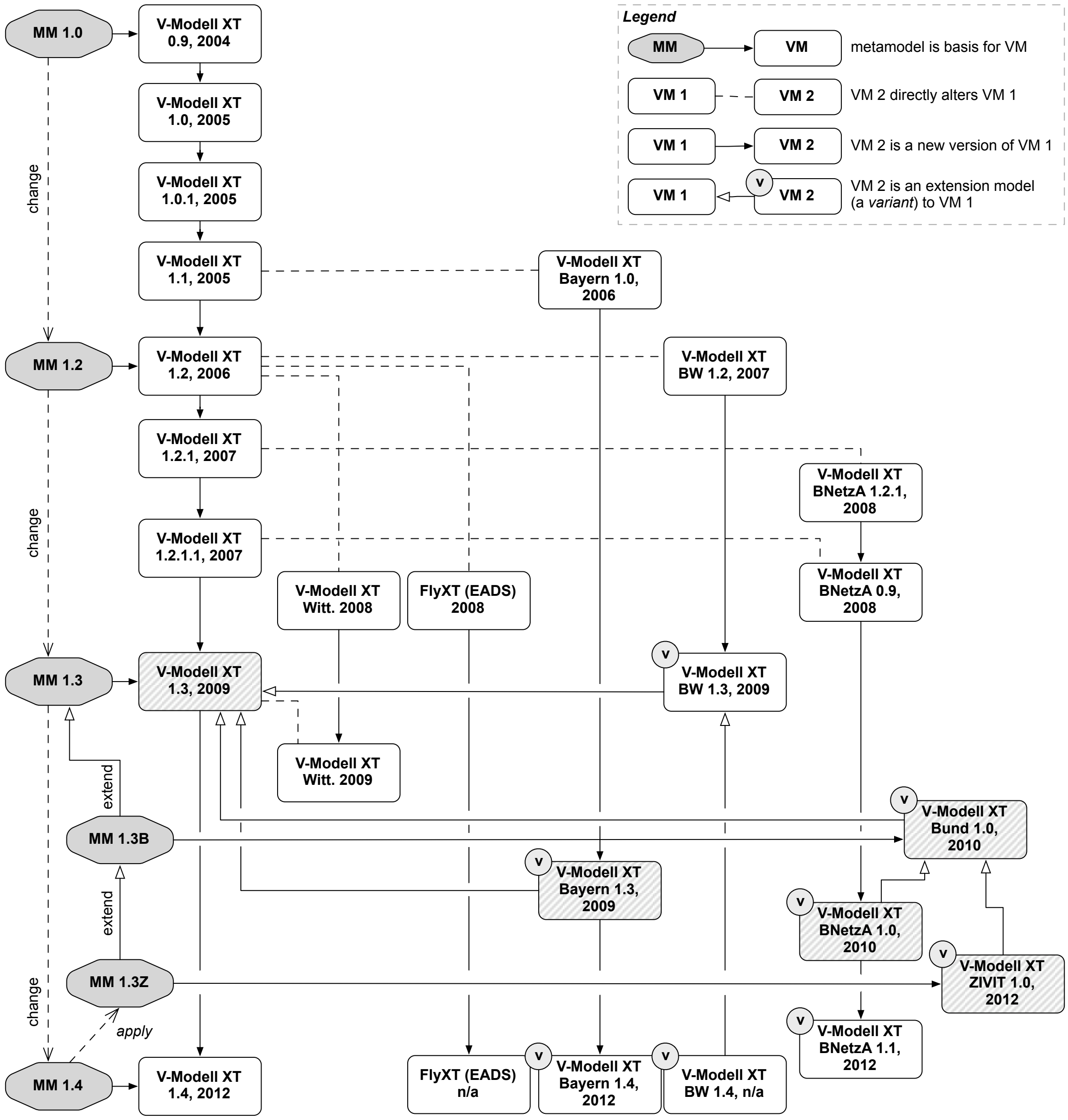

Figure 3.1: V-Modell XT metamodel and concrete process versions and variants

basis for further discussion and improvement. The Unified Method Architecture (UMA) and two RFPs fostered the discussion on SPEM and influenced its further development. On the one hand, these discussion leads to a comprehensive tool infrastructure (the Rational Method Composer in 2005, which was release as open source in 2006 as part of the Eclipse projects), and, on the other hand, a number of concrete processes was created (starting with the (commercial) RUP, the free Open Unified Process, to several smaller methods, such as Scrum).

The standardization work leads to SPEM 2.0 (released in 2008), which is a basis for a number of processes and also for research in the field of software processes. Figure 3.2 shows in lower part the different metamodel extensions that were developed around SPEM 2.0.

Due to the high number of contributions on SPEM, we do not include them in the figure. A comprehensive analysis can be found in Ruiz-Rube [44] and in Appendix A.1.

In difference to the V-Modell XT metamodel SPEM is under version and configuration management. Although there is a number of derivates available, only three major ver-

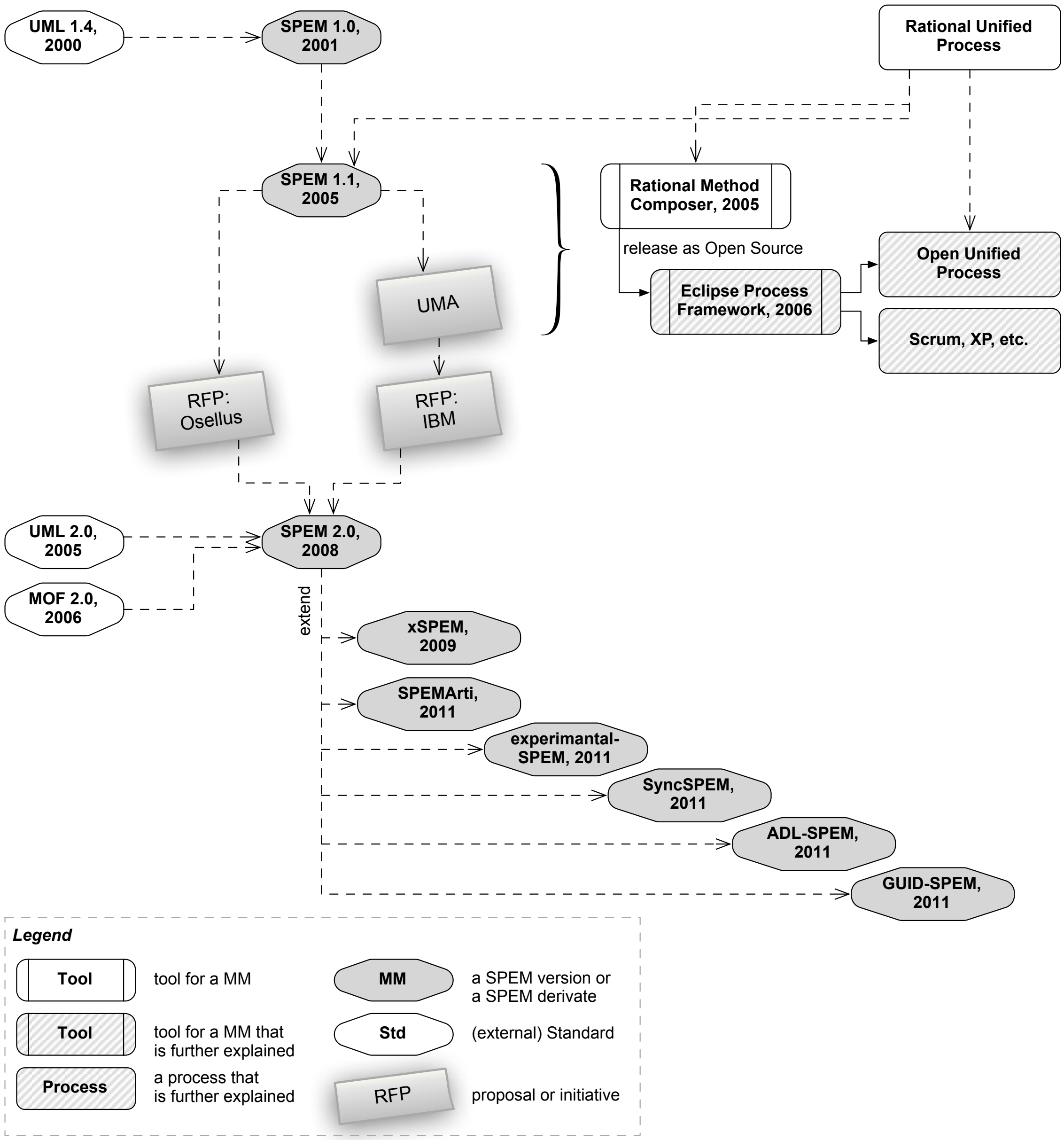

Figure 3.2: The SPEM metamodel and concrete process versions and variants

sions of SPEM can be found. The explanation is that SPEM is a standard published by the OMG and, therefore, underlies the regulations of maintenance, improvement, and further development.

3.2.3 OPEN and ISO 24744

The OPEN Process Framework was also identified to be a process metamodel for designing software processes. When applying the method of building family trees to investigate the evolution of a standard, the result in Fig. 3.3 is found.

There is several work around OPEN, but not experience report neither a tool support can be found. Analyzing the contributions assigned to OPEN most of them are in the area of (Situational) Method Engineering, i.e. [41].

In Fig 3.4, the results for ISO 24744 (SEMDM) are shown, which are similar to the re-

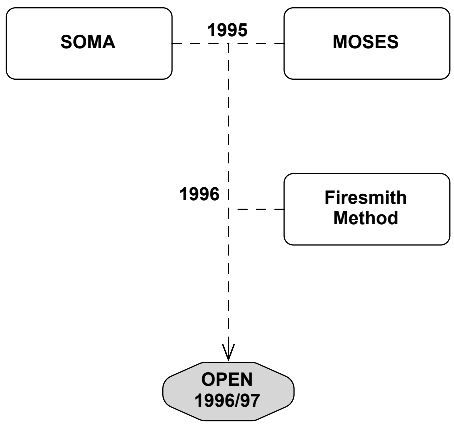

Figure 3.3: The OPEM metamodel and concrete process versions and variants

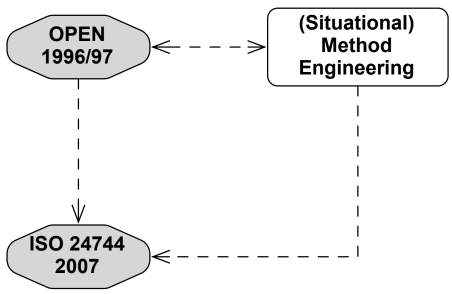

Figure 3.4: The ISO 24744 metamodel and concrete process versions and variants

sults for the OPEN metamodel. A detailed investigation of the description of OPEN and SEMDM shows the relationship of both metamodels. Furthermore, SEMDM is also highly influenced by the (Situational) Method Engineering community and is often exemplary used of discussing certain SME topics.

For both metamodels, OPEN as well as SEMDM, we could not find any comprehensive tool infrastructure supporting the development of concrete software processes. As mentioned in Sect. 3.1.2.3 and 3.1.2.4, a detailed analysis of the dissemination and the acceptance is, therefore, not possible.

3.3 RQ 3: How are established software process metamodels technically supported?

In the last research question, we address the technical support for SPMMs. Since our selection criteria excluded vendor-specific solutions, such as Stages[3], we searched for process frameworks that can be used to create arbitrary processes.

Our results show that there are only two comprehensive *and* established process frameworks available:

- Eclipse Process Framework (EPF)
- V-Modell XT

Both frameworks provide a tool set that allows for implementing and publishing software processes. EPF, for instance, is used as the basis for a number of software processes.

3 Stages, Method Park: `http://stages.methodpark.de`

Prominent examples are the Rational Unified Process[4], OpenUP, Hermes, and Scrum.

The V-Modell XT infrastructure is used to implement the V-Modell XT reference model and its variants, but it is also used for the T-Systems SE Book, and several research prototypes. The SE book is an interesting use case for the V-Modell XT infrastructure. The infrastructure is used to edit the T-Systems process, but does not use the V-Modell XT metamodel. T-Systems developed an own metamodel, which has certain relationship to the V-Modell XT metamodel, but also has significant differences.

Regarding a *Process Engineering Tool Chain* both, EPF-based processes as well as V-Modell-XT-based processes, are further supported by a number of tools and extensions. Hermes, for instance, which is based on SPEM provides a V-Modell-XT-like tailoring that is implemented as an EPF plug-in. The V-Modell XT, for instance, is subject to research in the field of process enactment.

Detailed information regarding the tools and tool support can be found in the appendix of this report.

4 For the implementation of RUP the commercial version of EPF, the Rational Method Composer, is used. However, the underlying metamodel and the basic software infrastructure is the same.

4 Conclusion

4.1 Outcomes

The report at hands collects and structures the first results of a investigation of the state-of-the-art in systematic software process design, development, and management. The results show that there are only a few systematic approaches, namely:

- Software & Systems Process Engineering Metamodel (SPEM)
- Software Engineering Metamodel for Development Methodologies (SEMDM)
- MetaME
- V-Modell XT

The results further show that only SPEM and the V-Modell XT are frequently and continuously used to implement software processes.

In this report, we summarized all data that we conducted in our (exploratory) literature review. In particular, we collected data w.r.t.:

- Common information, use, and the evolution of SPEM-based processes
- History and evolution of the Rational Unified Process (RUP)
- History and evolution of V-Modell XT (and its variants)
- Common information and history of EPF, incl. EPF-based processes

4.2 Interpretation

Since our major goal with this report is to collect and structure data in a first step, we only provide a short and *tentative* interpretation.

We interpret the data regarding OPEN and ISO 24744 (SEMDM) that especially SEMDM has no practical relevance. Besides some academic contributions that are mostly of a philosophical nature or a solution proposal, no practically relevant implementation is known. Furthermore, SEMDM has no tool support that is comparable to the EPF and V-Modell XT environments.

Another new and interesting approach is MetaME that is based on sound and structured concepts. However, it remains unclear what the relevance of MetaME (replacement of existing standards, new way of thinking, extension of existing standards) is. Furthermore, MetaME is yet not evaluated and, therefore, remains so far a solution proposal.

The V-Modell XT community is rather active in terms of improving the process framework and the reference processes. However, the V-Modell XT is a national standard with a "proprietary" platform (compared to OMG initiatives such as SPEM or an ISO standard). In consequence, the dissemination of the V-Modell XT concepts is limited. The major drawback is hereby that the key documentation is only available in German which limits the international relevance.

Our outcomes show that SPEM (in combination with EPF) is a frequently used process framework. A considerable number of contributions is available in which the application of SPEM is shown as well as proposals for extension are made. However, as shown in Sect. 3.2.2 the development of proposals to extend the metamodel seems to be chaotic.

Although a number of such proposals were made, there is no progress obvious neither a strategy w.r.t. the future of SPEM.

Furthermore, the data shows that there were no new releases of EPF-based processes since 2011. Even the EPF Composer only got a minor refresh in 2012. Especially the agile methods were probably not maintained (last releases: Scrum – 2008, XP – 2007). Quite interesting: There was a proposal to implement Feature-driven Development (FDD) as another agile method. This particular method is implemented as plug-in, but is not mentioned in the community at all. Other processes and process proposals were proposed between 2006 and 2008. Just HERMES as a national standard is continuously maintained. This could be a hint that implementing new software processes using EPF in particular and systematically develop software processes in general is nowadays not a "hot topic".

4.3 Future Work

Our research shows that there is still much work to do. Among all contributions we could only identify two SPMMs that are in practical use and that have an appropriate maturity (in terms of tool support, evolution of the platform, management of the platform, and so on).

Nevertheless, both platforms are not compatible. At the same time a number of contributions was made to improve SPEM; also the metamodel of the V-Modell XT is a subject of continuous improvement. Future work therefore contains deeper investigation of currently available metamodeling concepts and a harmonization of the most frequently used concepts to make software processes exchangeable. Such work should also pay attention to existing, but not standardized proposals for extending metamodels, i.e. in terms of process enactment.

Another aspect should also be respected: There are wildly accepted software processes such as PRINCE2 that, however, are not implemented in any process framework. The use of comprehensive software process frameworks can be fostered if such software processes would be implemented in an appropriate environment.

A Data Tables for SPEM-based Processes and Metamodels

This appendix contains all data tables that are related to the SPEM metamodel. In particular, this appendix contains the data tables w.r.t. the SPEM metamodel usage and also addresses concrete implementations that are based on the Eclipse Process Framework (EPF).

A.1 SPEM-based Metamodels

In the following all data table for the analyses are presented. The following tables summarize the survey's data. The columns contain the following information (based on Ruiz-Rube [44]):

Number of contributions: All contributions counted per year.

Use: Number of contributions that use a particular version of the SPEM metamodel.

Devise: Number of contributions that devise a particular SPEM version. A contribution in the "devised" category uses a new metamodel, which is—itself—based on a particular SPEM version.

Extend: Number of contributions that extend a SPEM version. A contribution in den "extend" category is based on an extended SPEM metamodel version.

Table A.1 summarizes all contributions, but leaves out the UMA-based contributions. Since UMA is an "intermediate" in the SPEM family, we do not provide an explicit classification, but list the UMA-based contributions in Table A.2.

For all other SPEM versions we provide detailed classification, which are based on ruiz-Rube [44] and extended by our own research. Table A.3 summarizes the data for the recent SPEM version 2.0, Table A.4 contains the data for SPEM 1.1, and, finally, Table A.5 shows the contribution that is based on SPEM 1.0.

Year	Number of Contributions	Use	Devise	Extend
2003	2	2		
2004	3	3		
2005	10	8		2
2006	13	8	2	3
2007	14	9	2	3
2008	14	11	1	2
2009	15	11		4
2010	29	19	4	6
2011	19	12	2	5
2012	2	2		
	121	85	11	25

Table A.1: All contributions on SPEM usage without UMA

Year	Number of Contributions	Use	Devise	Extend
2007	1	1		
2008	2	2		
2009	1	1		
2011	1	1		
	5	5		

Table A.2: All contributions using UMA

Year	Number of Contributions	Use	Devise	Extend
2005	1			1
2006	2	1	1	
2007	4	2		2
2008	6	4		2
2009	12	8		4
2010	23	15	2	6
2011	17	11	2	4
2012	2	2		
	67	43	5	19

Table A.3: All contributions using SPEM 2.0

Year	Number of Contributions	Use	Devise	Extend
2003	2	2		
2004	3	3		
2005	9	8		1
2006	10	7	1	2
2007	10	7	2	1
2008	8	7	1	
2009	3	3		
2010	6	4	2	
2011	2	1		1
	53	42	6	5

Table A.4: All contributions using SPEM 1.1

Year	Number of Contributions	Use	Devise	Extend
2006	1			1

Table A.5: All contributions using SPEM 1.0

A.2 Concrete EPF-based implementations

A popular implementation of SPEM (and also of a number of UMA concepts) is given by the Eclipse Process Framework (EPF). Especially the EPF-based implementation of SPEM became popular, hence it also provides the infrastructure for the *Rational Unified Process* (RUP), and is, furthermore, an open source project hosted by the Eclipse Foundation. In this section we summarize the evolution of the EPF infrastructure as well as several EPF-based processes[1].

A.2.1 EPF History

In the following, the history of the Eclipse Process Framework is summarized. Tabel A.6 summarizes the releases and the corresponding release dates. The latest EPF release is numbered *1.5.1.4*.

EPF Release	Release Date
1.0	2006-10-02
1.0.1	2006-10-27
1.0.1.2	2006-12-13
1.0.2	2007-03-05
1.2.0	2007-08-01
1.2.0.1	2007-09-14
1.2.0.2	2007-11-20
1.2.0.3	2008-03-28
1.2.0.4	2008-06-25
1.5.0	2008-08-25
1.5.0.1	2008-10-28
1.5.0.2	2008-12-18
1.5.0.3	2009-07-16
1.5.0.4	2009-10-09
1.5.1	2010-09-30
1.5.1.1	2010-11-19
1.5.1.2	2011-06-10
1.5.1.3	2011-11-30
1.5.1.4	2012-06-01

Table A.6: History and releases of the Eclipse Process Framework

A.2.2 EPF-based Standard Processes

Based on EPF, a number of so-called standard processes was implemented, e.g. Scrum [46] or XP [4]. In this section we give an overview over such standard processes and add release information where possible.

A.2.3 Further EPF-based Processes

Besides the concrete software processes that are implemented using EPF further processes use this infrastructure for modeling software processes. Some examples are:

1 Have in mind that not all listed processes are officially supported by the community, some of those implementations were—unfortunalety—rejected by the community for “unknown” reasons.

OpenUP Release	Date	Scrum Release	Date	XP Release	Date
				0.1	2007-07-24
0.9 (basic)	2006-10-02				
1.0	2007-08-01	1.0	2007-02-12		
		1.1	2007-07-24		
		1.2	2007-07-26		
		1.3	2007-08-17		
		1.4	2007-11-02		
1.5	2008-08-21	1.5	2008-08-20		
1.5.0.1	2008-10-22				
1.5.0.2	2008-10-27				
1.5.0.3	2009-02-27				
1.5.0.4	2009-10-08				
1.5.1	2010-09-28				
1.5.1	2010-10-07				
1.5.1.1	2010-11-30				
1.5.1.2	2011-06-10				

Table A.7: Standard processes implemented using EPF

- The Agile Database Techniques Plug-In for OpenUP/Basic[2]
- TOGAF 9 Method Plugin for the Eclipse Process Framework Composer[3]
- Agile Business Rule Development EPF plugin[4]
- ITIL (in Rational Method Composer)[5]
- An enterprise process framework defined and delivered with IBM Rational Method Composer[6]
- Feature-driven Development Plug-In[7]
- ICONIX Process[8]
- KAOS β[9]
- HERMES PowerUser Release 2.1.1[10]

Since there is no central process repository nor a central list of processes or process variants that use EPF, no information is available which of those processes is maintained, used, or evaluated. Furthermore, beyond the above listed processes a number of scientific contributions is available that use EPF, e.g., to implement prototypes. Ruiz-Rube provided a list of contributions that make use of SPEM-based software processes in [44].

2 `http://www.ambysoft.com/epf`; v0.3 (Beta), Nov. 30th, 2006; requires: OpenUP
3 `http://www.opengroup.org/architecture/togaf/epf_intro.html`; 2008; requires: EPF Composer 1.2.0.3 (or newer)
4 `http://www.agileitarchitecture.com/2008/02/agile-business-rule-development-epf.html`; v1.0 (pre-release), Feb. 27th, 2008; requires: OpenUP
5 `http://www.ibm.com/developerworks/rational/library/dec05/james/index.html?S_TACT=105AGX15&S_CMP=EDU`; Dec. 15th, 2005
6 `http://www.ibm.com/developerworks/rational/library/feb07/snyder/index.html?S_TACT=105AGX15&S_CMP=EDU`; Feb. 15th, 2007
7 `http://www.planeteclipse.net/mhonarc/lists/epf-dev/msg02520.html`; Dec. 4th, 2007
8 `http://iconixprocess.com/?s=roadmap`; Jan. 28th, 2007
9 `http://etsmtl.ca/Professeurs/rchampagne/documents/epftutorial/Eclipse-process-Framework-Step-by-step-example.pdf`; March 2010
10 `http://www.hermes.admin.ch/ikt_projektfuehrung/poweruser/hermes-power-user-release-2.0`; Feb. 25th, 2011

Those processes—as they are of a scientific nature—are usually implemented using EPF as it is for free.

The only software process that is "officially" released and maintained is *HERMES* since it is the standard development model for Switzerland government authorities (comparable to the German V-Modell XT).

B Data Tables for Survey Replication and Extension

B.1 Replication and Extension Data

Table B.1 lists all those contributions that resulted from our search and that are not contained in the result set of Ruiz-Rube [44]. For each of the paper we did a classification of the research type facet. Furthermore, we integrated those contribution into the data tables of App. A.1.

		Research Type			
Title	**Reference**	Evaluation	Solution	Position	Experience
SPEM on test: the SODA case study	[1]				X
SynchSPEM: A Synchronization Metamodel Between Activities and Products Within a SPEM-based	[42]		X		
SysPEM: Proposing A Coherent Model For Systems Engineering Processes	[22]		X		
SPEM2XPDL: Towards SPEM Model Enactment	[11]		X		
Deriving Software Process Simulation Model from SPEM-based Software Process Model	[39]		X		
Modelling software process variability: an empirical study	[30]	X			
Methodologies for self-organising systems: a SPEM approach	[40]		X		
Constructing Tool Chains Based on SPEM Process Models	[29]		X		
UML4SPM: An Executable Software Process Modeling Language Providing High-Level Abstractions	[5]		X		
Towards a rigorous process modeling with SPEM	[2]		X		

Table B.1: Additional SPEM-related contributions and their classification according to research type facets.

B.2 Further Survey Resources

This section contains the papers that were found during the survey and that met all our criteria as listed in Sect. 2.2. This result set also contains the standards for software process metamodeling as well as further documented metamodel-based approaches. Therefore, Table B.2 only distinguishes between standards and "general" (scientific) contribution. The category *Standard* contains national, industrial, and international standards.

Title	Reference	Type: Standard	Type: Contribution
Software Engineering Metamodel for Development Methodologies	[23]	X	
Software & Systems Process Engineering Meta-Model Specification	[36]	X	
Das V-Modell XT 1.3 Metamodell	[49]	X	
Business Process Definition Metamodel Volume 1	[38]	X	
An Approach for the metamodel of the framework for a partial agile method adaption	[32]		X
A comparison of four process metamodels and the creation of a new generic standard	[18]		X
A Meta-Method for Defining Software Engineering Methods	[10]		X
Bridging the Gap from Process Modelling to Process Assessment: the OOSPICE Process Specification for Component-Based Software Engineering	[20]		X
A method to build information systems engineering process metamodels	[21]		X
Comparing collaborative and process semantics for cooperative information systems	[19]		X
The Metamodel: A Starting Point for Design Processes Construction	[47]		X
FMESP: Framework for the Modeling and Evaluation of Software Processes	[14]		X
PIT-ProcessM: A Software Process Improvement Meta-model	[31]		X

Table B.2: Further contributions on metamodels and metamodel-based software processes.

B.3 The RUP History

The Rational Unified Process (RUP [26]) is a comprehensive software process comparable to the German V-Modell XT or the Swiss Hermes. RUP can be considered to be the major representative of the Unified Process (Sect. 3.1.1.2). Starting in 2007 RUP uses SPEM to structure and describe its contents. In Fig. B.1 the history of RUP is shown.

The figure shows the evolution of the entire process model. In the early days of RUP a lot of consolidation work was done to create the initial releases of RUP (e.g., integration of the Unified Method, Objectory Process, and the Rational Approach). In the following years, a continuous improvement was done. For instance:

- *RUP 5.0, 1997:* Objectory UI design, Data Engineering, and UML 1.1 were added as concrete technologies; Configuration & Change Management, Business Engineering were added as methodological support.
- *RUP 5.5, 1998:* SPC/PMI-like Project Management was added as methodological support.
- *RUP 2001, 2000:* The concepts of Architecture and Proof-of-Concept were added as methodological support.

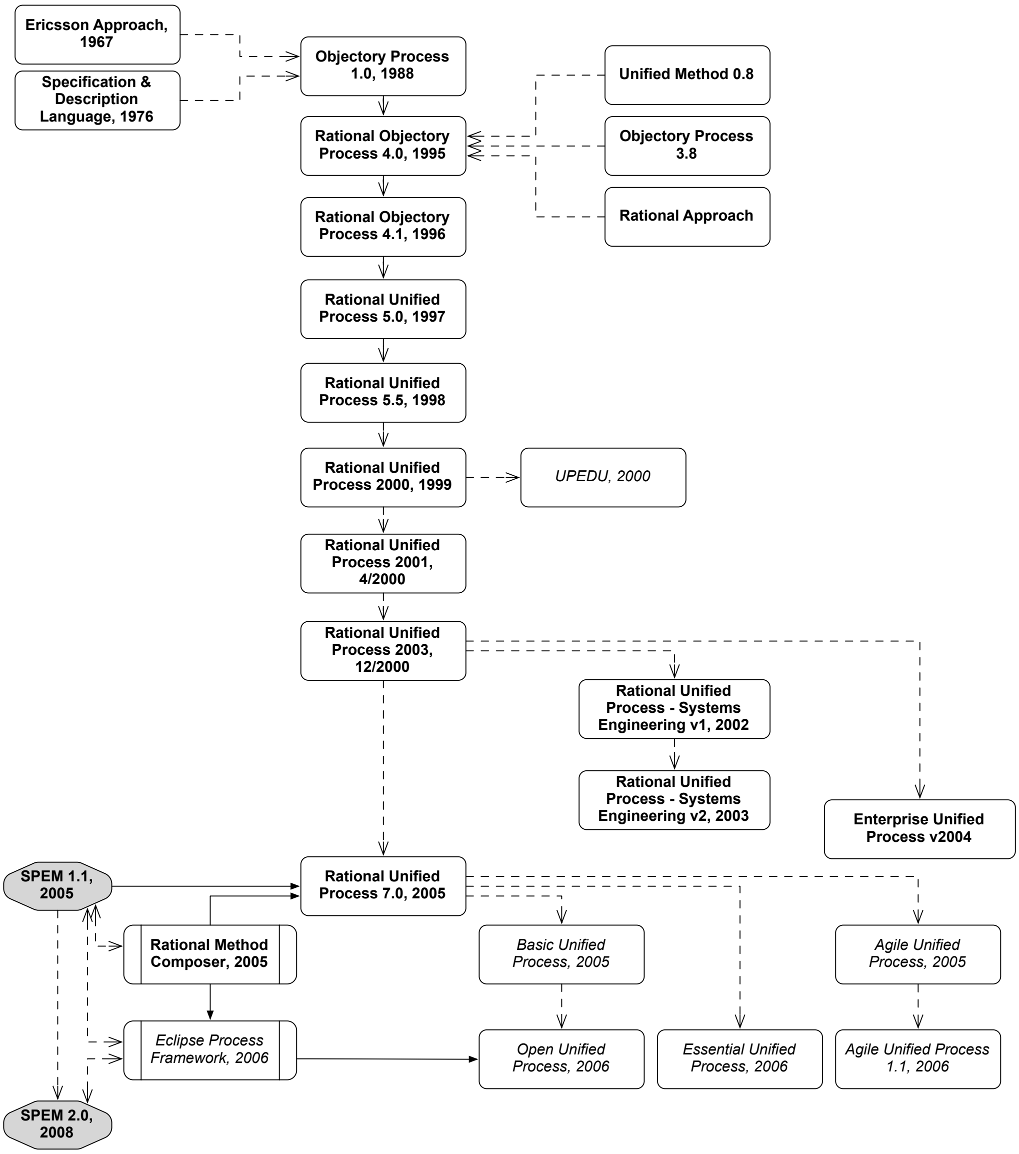

Figure B.1: The history of RUP including variants and tool hot spots.

In 2005 the *Rational Method Composer* (RMC) was released. At this time there was a shift in the style of designing and improving RUP, since RUP now was completely implemented in a tool. RMC itself was (partially) moved to the Eclipse community and there released as *Eclipse Process Framework Composer* in 2006.

Remark. Also starting in 2005/2006 the relationships among the different UP derivates becomes blurry. Some process derivates—at least variants—were published. Especially the EPF community grew and fostered an number of process implementation projects (examples can be found in Sect. A.2.3).

B.4 V-Modell XT Technology

Similar to SPEM the V-Modell XT is also complemented with a comprehensive tool infrastructure. In App. A.2.1 an overview over the development of the Eclipse Process

Framework (EPF) Composer was given. In this section we add the corresponding information w.r.t. the V-Modell XT tools.

The Table B.3 lists the V-Modell XT tools. The infrastructure consists of two major components: The *V-Modell XT Editor* is the authoring component, similar to the EPF composer. The *V-Modell XT Project Assistant* is a tool that can be used to set up a project. The assistant supports project managers in tailoring a process, creating work product templates, and provides simple planning capabilities to create initial project plans. Table B.3 lists all available information regarding both tools[1] and shows which tool configuration was included in the process's major releases.

Editor Release	**Release Date**	**Project Assistant Release**	**Release Date**	**V-Modell XT Major XT Release**
2.1.2				
3.0.0	2005-02-04	1.0.0	2005-02-04	1.0
3.0.2		1.0.1		
3.1.0		1.0.2		
3.1.2		1.0.3		
3.1.3		1.0.4		
3.1.4		1.0.5		
3.1.5		1.1.0		
		1.1.1		
		1.1.2		
		1.1.3		
3.2.0	2009-03-10	1.2.0	2009-03-10	1.2
3.2.1				
3.2.2				
3.2.3				
3.2.4		1.2.1		
3.2.5		1.2.3	2006-10-02	
3.2.6				
3.3.0	2007-10-30	1.2.4	2007-10-29	
3.3.3	2009-01-08	1.3.1	2009-01-08	1.3
3.3.4	2009-03-09	1.3.2	2009-03-09	
3.3.6	2009-11-11	1.3.3	2009-08-10	
3.3.7	2009-12-04	1.3.4	2009-10-27	
3.3.8	2009-12-18	1.3.5	2009-12-04	
		1.3.6	2011-02-25	
3.4	2012-06-04	1.4	2012-06-04	1.4
		1.4.1	2012-10-02	

Table B.3: History and releases of the V-Modell XT Tools

In addition to the tool releases from Table B.3 both tools were also customized. In Sect. 3.2.1 the family tree for the V-Modell XT and its variants is shown. Especially for the variant *V-Modell XT Bund* both the editor and the project assistant tools were customized. The customization included:

- Minor changes in the tools to enable them to use the evolved metamodel
- A branding that includes new logos and so on

1 Since there was a repository migration during the development, not all information is available.

The technical improvements were reintegrated into the main tools and are available since editor (rel. 3.4) and project assistant (rel. 1.4) for all users.

Remark. An aspect that is important to know: The V-Modell XT infrastructure requires OpenOffice.org and/or LibreOffice as backend components. During the development of the infrastructure components several (tentative) releases were necessary to deal with improvements of OpenOffice.org/LibreOffice that cause serious incompatibilities. The concrete required OpenOffice.org/LibreOffice versions that are required to operate a concrete tool configuration is left out here. The installer packages each contain appropriate OpenOffice.org/LibreOffice backends.

Bibliography

[1] ANDREA OMICINI, E. D. Spem on test: the soda case study.

[2] B. COMBEMALE, X. CRÉGIT, A. CAPLAIN, AND B. COULETTE. Towards a rigorous process modeling with spem.

[3] BALDUINO, R. Introduction to OpenUP (Open Unified Process). Online, `http://www.eclipse.org/epf`, 2007.

[4] BECK, K. *Extreme Programming*. Addison-Wesley, 2003.

[5] BENDRAOU, R., GERVAIS, M.-P., AND BLANC, X. Uml4spm: An executable software process modeling language providing high-level abstractions. In *2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)* (2006), IEEE, pp. 297–306.

[6] BENDRAOU, R., JÉZÉQUEL, J. M., GERVAIS, M. P., AND BLANC, X. A Comparison of Six UML-Based Languages for Software Process Modeling. *Software Engineering, IEEE Transactions on 36*, 5 (2010), 662–675.

[7] CONFÉDÉRATION SUISSE. The HERMES Method. Online: `http://www.hermes.admin.ch`, 2011.

[8] COSTACHE, D., KALUS, G., AND KUHRMANN, M. Design and Validation of Feature-based Process Model Tailoring - A Sample Implementation of PDE. In *Proceedings of the 8th European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEM/FSE 2011)* (sep 2011), ACM Press, pp. 464–467.

[9] ECLIPSE FOUNDATION. Eclipse Process Framework (EPF). Online, `http://www.eclipse.org/epf`, 2010.

[10] ENGELS, G., AND SAUER, S. Graph transformations and model-driven engineering. Springer-Verlag, Berlin, Heidelberg, 2010, ch. A meta-method for defining software engineering methods, pp. 411–440.

[11] F. YUAN, M. L. Z. W. Spem2xpdl-towards spem model enactment.

[12] FIRESMITH, D., AND HENDERSON-SELLERS, B. *The OPEN Process Framework. An Introduction*. No. ISBN-13: 978-0201675108. Addison-Wesley Longman, 2001.

[13] FRIEDRICH, J., HAMMERSCHALL, U., KUHRMANN, M., AND SIHLING, M. *Das V-Modell XT - Für Projektleiter und QS-Verantwortliche kompakt und übersichtlich*, 2. ed. No. ISBN: 978-3-540-76403-8 in Informatik im Fokus. Springer, 2009.

[14] GARCÍA, F., PIATTINI, M., RUIZ, F., CANFORA, G., AND VISAGGIO, C. A. Fmesp: Framework for the modeling and evaluation of software processes. *Journal of Systems Architecture 52*, 11 (2006), 627 – 639. <ce:title>Agile Methodologies for Software Production</ce:title>.

[15] GAU, T. Uma und epf: Einführung und anwendung in der praxis. *Objekt Spektrum*, November/Dezember 2006 - 6 (2006), 42–47.

[16] GONZALEZ-PEREZ, C. *Situational Method Engineering: Fundamentals and Experiences*, vol. 244 of *IFIP — The International Federation for Information Processing*. Springer, 2007, ch. Supporting Situational Method Engineering with ISO/IEC 24744 and the Work Product Pool Approach.

[17] GRAHAM, I., HENDERSON-SELLERS, B., AND YOUNESSI, H. *The Open Process Specification*. ACM-Press. Addison-Wesley, 1997.

[18] HENDERSON-SELLERS, B., AND GONZALEZ-PEREZ, C. A comparison of four process metamodels and the creation of a new generic standard. *Information and Software Technology 47*, 1 (2005), 49 – 65.

[19] HENDERSON-SELLERS, B., AND HAWRYSZKIEWYCZ, I. Comparing collaborative and process semantics for cooperative information systems. *International Journal of Cooperative Information Systems 17*, 02 (2008), 155–176.

[20] HENDERSON-SELLERS, B., STALLINGER, F., AND LEFEVER, B. Bridging the gap from process modelling to process assessment: the oospice process specification for component-based software engineering. In *Euromicro Conference, 2002. Proceedings. 28th* (2002), pp. 324 – 331.

[21] HUG, C., FRONT, A., RIEU, D., AND HENDERSON-SELLERS, B. A method to build information systems engineering process metamodels. *Journal of Systems and Software 82*, 10 (2009), 1730 – 1742. <ce:title>SI: YAU</ce:title>.

[22] JAKJOUD, A., ZRIKEM, M., AYADI, A., BARON, C., AND AURIOL, G. Syspem: Proposing a coherent model for systems engineering processes. In *2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)* (2011), IEEE, pp. 367–372.

[23] JOINT TECHNICAL COMMITTEE ISO/IEC JTC 1, SUBCOMMITTEE SC 7. Software engineering – metamodel for development methodologies. Tech. Rep. ISO/IEC 24744:2007, International Organization for Standardization, 2007.

[24] KITCHENHAM, B., BRERETON, O. P., BUDGEN, D., TURNER, M., BAILEY, J., AND LINKMAN, S. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology 51*, 1 (2009).

[25] KÖHLER, P. *Prince 2 - Das Projektmanagement-Framework*. No. ISBN: 3-540-29181-4 in Xpert.press. Springer, 2006.

[26] KRUCHTEN, P. *The Rational Unified Process: An Introduction*, 3 ed. Addison-Wesley Longman, 2003.

[27] KUHRMANN, M. *Konstruktion modularer Vorgehensmodelle*. PhD thesis, Technische Universität München, 2008.

[28] KUHRMANN, M., TERNITÉ, T., AND FRIEDRICH, J. *Das V-Modell XT anpassen*. Informatik im Fokus. Springer, 2009. (to appear).

[29] M. BIEHL, M. T. Constructing tool chains based on spem process models.

[30] MARTÍNEZ-RUIZ, T., GARCÍA, F., PIATTINI, M., AND MÜNCH, J. Modelling software process variability: an empirical study. *IET Software 5*, 2 (2011), 172.

[31] MARTINS, P., AND DA SILVA, A. Pit-processm: A software process improvement meta-model. In *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the* (29 2010-oct. 2 2010), pp. 453 –458.

[32] MIKULĖNAS, G., BUTLERIS, R., AND NEMURAITĖ, L. *An Approach for the Metamodel of the Framework for a Partial Agile Method Adaptation*, vol. 40. INFORMATION TECHNOLOGY AND CONTROL, 2011.

[33] MÜNCH, J., ARMBRUST, O., SOTO, M., AND KOWALCZYK, M. *Software Process Definition and Management*. Springer, 2012.

[34] OESTEREICH, B., SCHRÖDER, C., KLINK, M., AND ZOCKOLL, G. *OEP - OOSE Engineering Process*. No. ISBN: 3-89864-407-3. dpunkt.verlag, 2006.

[35] OMG. Unified modeling language (uml): Infrastructure – version 2.1.1. Tech. rep., Object Management Group, 2007.

[36] OMG. Software & Systems Process Engineering Metamodel Specification (SPEM) Version 2.0. Tech. rep., Object Management Group, 2008.

[37] OMG. Unified modeling language (uml): Superstructure version 2.2. Tech. rep., Object Management Group, 2009.

[38] OMG. Business Process Model and Notation Version 2.0. Tech. rep., Object Management Group, 2010.

[39] PARK, S., CHOI, K., YOON, K., AND BAE, D.-H. Deriving software process simulation model from spem-based software process model. In *Deriving Software Process Simulation Model from SPEM-based S* (2007), IEEE, pp. 382–389.

[40] PUVIANI, M., DI SERUGENDO, G. M., FREI, R., AND CABRI, G. Methodologies for self-organising systems: A spem approach. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* (2009), IEEE, pp. 66–69.

[41] RALYTÉ, J., BRINKKEMPER, S., AND HENDERSON-SELLERS, B., Eds. *Situational Method Engineering: Fundamentals and Experiences*. Springer, 2007.

[42] ROCHD, A., ZRIKEM, M., AYADI, A., MILLAN, T., PERCEBOIS, C., AND BARON, C. Synchspem: A synchronization metamodel between activities and products within a spem-based software development process. In *2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)* (2011), IEEE, pp. 471–476.

[43] ROMBACH, D. Integrated Software Process and Product Lines. In *International Software Process Workshop (SPW 2005)* (2005).

[44] RUIZ-RUBE, I., DODERO, J. M., PALOMO-DUARTE, M., RUIZ, M., AND GAWN, D. Uses and Applications of SPEM Process Models. A Systematic Mapping Study. *Journal of Software Maintenance and Evolution: Research and Practice 1*, 32 (2012).

[45] RUNESON, P., AND HÖST, M. Guidelines for conducting and reporting Case Study Research in Software Engineering. *Empirical Software Engineering 14*, 2 (2009), 131–164.

[46] SCHWABER, K. *Agile Project Management with Scrum*. Microsoft Press, 2004.

[47] SEIDITA, V., COSSENTINO, M., HILAIRE, V., GAUD, N., GALLAND, S., KOUKAM, A., AND GAGLIO, S. The metamodel: a starting point for design processes construction. *International Journal of Software Engineering and Knowledge Engineering 20*, 4 (2010), 575–608.

[48] TERNITÉ, T. *Variability of Development Models*. PhD thesis, Technische Universität Clausthal, 2010.

[49] TERNITÉ, T., AND KUHRMANN, M. Das v-modell xt 1.3 metamodell. Tech. Rep. TUM-I0905, Technische Universität München, 2009.