SOFTWARE PROCESS IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. 2006; **11**: 229–238
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/spip.266

Software Process Improvement in Europe: Potential of the New V-Modell XT and Research Issues

Practice Section

Stefan Biffl[1],[*,†] Dietmar Winkler[1], Reinhard Höhn[2] and Herbert Wetzel[3]
[1] *Vienna University of Technology, Institute of Software Technology and Interactive Systems, A-1040 Vienna, Austria*
[2] *Knowledge Management-Associates GmbH, Lerchenfelder Gürtel 43, A-1160 Vienna, Austria*
[3] *IMG AG, Fürstenlandstrasse 101, CH-9014 St Gallen, Switzerland*

The goal of European industrial practice to support high-value software production in diversified domains has led to the development of a huge number of process model variants. However, these diverse models are hard to compare, which hinders efficient collaboration and software process improvement on a European level. Process managers see a growing need for approaches that support stakeholder collaboration, systematic process mapping, and transformation of processes to improve their leverage in software process improvement. In this article we present the V-Modell XT (VM XT), a flexible software process model approach that has recently been announced as the standard for public-sector IT projects in Germany, as promising opportunity to help provide a unifying European software process model 'umbrella'. On the basis of strengths of the VM XT, we suggest research directions for advanced support of software projects: (i) effective business value translation to engineering solutions that strengthens stakeholder collaboration, (ii) process mapping that enables collaboration in projects that have to reconcile several process models, and (iii) process 'product lines' to capture the variability of software processes on domain and company levels and thus systematically help investigate best-practice approaches to software construction. We discuss these concepts, the contribution of the VM XT, and conclude with next steps for research and validation.
Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: software process improvement; V-Modell XT; value-based software engineering; business value translation; software process mapping; process product lines

1. INTRODUCTION

A common goal in software engineering is the construction of very valuable high-quality software products. However, concepts of high quality and value depend on the needs of the involved project

* Correspondence to: Stefan Biffl, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Favoritenstraβe 9-11/188 A-1040 Vienna, Austria
† E-mail: stefan.Biffl@tuwien.ac.at

Copyright © 2006 John Wiley & Sons, Ltd.

stakeholders to be able to define 'business value' and translate it into software products (Biffl *et al*. 2005, Boehm 2003, Roel 1995, Sharp *et al*. 1998). A subsequent ongoing challenge in modern software engineering is to establish a balance between product quality, functional range, development duration, and development cost. Software processes define development methodologies, independent of individual projects, regarding (i) process steps, performed in predefined sequences, (ii) products, (iii) activities and responsibilities, and (iv) sets of methods and tools to support project work in order to improve project planning and execution.

A defined software process provides the basis to gather experience and data for systematic process improvement and can thus lead to better products. However, the wide range of available software process approaches, which were often developed to fit specific application domains, project complexities, and project sizes, make it a challenge to compare best-practice experiences and select a ready-made process for any given project context. Therefore, many organizations customize their software processes according to their individual requirements based on well-known process models, e.g. the *System Engineering Method* (SEM) of Siemens Program and Systems Engineering (PSE) (Austria)[1] and *Promet* (IMG, Switzerland) (Österle and Winter 2003).

However, if the underlying model needs to be changed, it is often very difficult to propagate these modifications to the models customized from the underlying model. Managers of individual projects and customers – especially in the public sector – may have a stake in the application of a specific software process model; in a multicompany project this often means reconciling several process models to define an effective process model for the project at hand. These process diversifications and the wide range of requirements fuel a need for a means to assess the comparability and compatibility of two or more software process models.

A first step can be to find out to what extent two process models fit together in their current form; a next step could be to find a meta-model that presents a unified process model to which existing process models can be mapped. On the European level this approach allows comparison and use of process models, which implement national regulations and/or domain-specific needs, under a common umbrella. Such an approach for unification that allows systematic diversification is in line with the EU Lisbon declaration[2] goals to support European knowledge management in the area of software processes: software process models contain knowledge on how to make successful products; they can be used for knowledge acquisition and transfer, e.g. by providing structure to e-learning approaches.

Such an umbrella could be derived from the V-Modell XT (VM XT) (Rausch *et al*. 2005, V-Modell XT 2005), a very flexible new software engineering process model that covers the whole life-cycle within a framework of IT solutions (systems engineering). After more than a decade of experience with previous releases (V-Modell 1997), the VM XT was released by the German Ministry of Interior in February 2005 as a mandatory process model for public-sector IT projects in Germany. The process model is supported with a set of open-source tools.

The VM XT concentrates on the technical software engineering process with special attention to the 'call for tender' as a preliminary process step and an enhanced involvement of the customers within the project course. Additionally, the VM XT allows tailoring individual approaches according to different project types (including the implementation and improvement of software processes) and combines systematic process guidance with more flexibility than most other software process models: the VM XT provides a framework for software development, with the potential to grow into a European software process model.

On the basis of the strengths of the VM XT we suggest three research directions for advanced support of software projects.

1. *Business value translation that strengthens stakeholder collaboration:* Economically successful projects are based on understanding stakeholder value propositions (Biffl *et al*. 2005) and translating them into IT requirements and engineering solutions (see Figure 1). This remains a challenge, especially in IT projects with a public 'call-for-tender'. The VM XT pays special attention to this preliminary phase and integrates success-critical customer contributions. Research can enhance the technical focus of

[1] http://www.pse.siemens.at

[2] http://europa.eu.int/eur-lex/lex/LexUriServ/LexUriServ.do?uri=CELEX:52003PC0509:EN:HTML

Copyright © 2006 John Wiley & Sons, Ltd.

Softw. Process Improve. Pract., 2006; **11**: 229–238
DOI: 10.1002/spip

the VM XT with better integration of input from business process models, such as *Promet* (Österle and Winter 2003). Another challenge of value creation is to realign requirements and engineering solutions that evolve concurrently during the course of the project with proper synchronization mechanisms. While software process models propose synchronization mechanisms that work well for in-house projects, collaboration in distributed projects, potentially across several companies and countries, needs more advanced process support.

2. *Process mapping that enables collaboration in projects that have to reconcile several process models.* VM XT enables tailoring of particular process modules (PMs) during the project course, e.g. adding individual PMs. This flexibility eases mapping other process models to VM XT (e.g. for organizations, which want to use their in-house process model to bid for a public-sector IT project in Germany and have to show compliance with the VM XT). Further, VM XT can be used as a common representation for several other process models and support more efficient mapping between any of these models (e.g. for process managers in multicompany projects who have to derive a common process model that fits the project partners' needs and competences); a difficult task in many cases today.
3. *Process 'product lines'* that capture the variability of software processes on domain and company levels and thus systematically investigate best-practice approaches to software construction. Many projects have to consider domain-specific business requirements, e.g. dependable systems, financial transaction services, security-related applications, embedded systems (Klemen and Biffl 2004). Consequently, process managers see a strong need to describe domain-specific software processes (see also approaches in CMMI (Chrissis *et al*. 2003, http://www.sei.cmu.edu/cmmi Kneuper 2003) and SPICE (El-Emam *et al*. 1998, ISO/IEC 15504-x, Loon 2004, http://www.sqi.gu.edu.au/spice). We propose a process 'product-line' approach (Schmid and Biffl 2005) that uses VM XT as the basis for a software process meta model, and allows deriving process models that consider regulations from business domains, national standards, and company procedures. Systematically capturing process variability with such a product-line approach opens up the chance for context-specific empirical investigation of best-practice approaches to software construction.

The remainder of this paper is structured as follows. Section 2 describes the basics of the VM XT. Section 3 focuses on the contribution of VM XT to business value translation. Section 4 outlines the mapping and model translation approaches to support collaboration in projects that have to reconcile several process models. Section 5 describes enhancements to product lines to define process 'product lines' that capture the variability of software processes on domain and company levels. Section 6 concludes and outlines next steps for further research.

2. THE V-MODELL XT PROCESS FRAMEWORK

The VM XT (V-Modell 2005), successor of the V-Modell 97 (V-Modell 1997), is a software development framework for planning and execution of software processes. XT stands for *eXtreme Tailoring* due to the flexible customizing ability of the process framework to application needs, project types, and project complexity. In comparison to previous releases (V-Modell 1997), new features in VM XT cover regulations for hardware development, logistics, project management, and process improvement. Furthermore, VM XT supports a preliminary call for tender as a precondition for the technical product development process, including bidding scenarios for contractors and customers. VM XT assists three different project types: (i) system development from customer point of view, (ii) system development from contractor point of view, and (iii) implementation and maintenance of the organization-specific software process including continuous product and process improvement. The latter project type includes further development, extensions, and software process improvement of VM XT as well as implementation and customizing approaches for specific organizations.

In addition to the three different project types, the VM XT framework (V-Modell XT 2005) describes a set of 99 products, 18 PMs, 18 decision points (DPs), and seven project operation strategies:

Copyright © 2006 John Wiley & Sons, Ltd.
DOI: 10.1002/spip

1. *Products* are results of activities performed by defined roles, that apply a set of methods and techniques. Many of these products are supported by document templates for project application.
2. Process Modules (PMs) include a set of products (product-centric approach), corresponding activities, and basic method suggestions. Project operation strategies enable customizing and tailoring of PMs *to achieve appropriate process course.* Some basic PMs are mandatory for all project types (VM XT core components); others are optional depending on the application domain and the type of application.
3. Decision Points (DPs) are related to a subset of products and represent the state of treatment and the state of consistency.
4. Finally, a *Project Operation Strategy* (*POS*) is defined as a sequence of DPs for project course (including repetition, cycles, and skipping). According to individual requirements of the organization, project operation strategies may be customized, e.g. regarding incremental, agile, component-based development as well as migration and maintenance purposes.

This set of concepts is the basic process model, which can be adjusted to the business domain and project types. Therefore, there are three possibilities for model application: (i) immediate support for project course without change of the basic concept; (ii) application after tailoring according to several project criteria within the range of optional PMs, and (iii) extension and customizing of the basic model according to individual needs using an XML-based editor, provided by the developers of VM XT as an open-source tool.

Figure 1 displays the basic concepts of VM XT and its integration with business processes, including the call for tender regarding requirements engineering and negotiation, and offer scenarios. VM XT is embedded within the IT-system layer, which consists of three steps: (i) call for tender as preliminary process for public domain customers, (ii) the technical realization, and (iii) the operation and maintenance phase. The three VM XT project types and their involvement in the corresponding PMs can be identified as shaded marks in the process model. The software process schematically represents in the form of a 'V' the descending analytical development branch and the ascending synthetic development branch.

These horizontal interrelationships depict different views, e.g. implementation view (at the bottom level), architectural view, and user view (at the top level). Depending on the individual software processes of an organization, the schematic V-approach may be used as the actual project process or be replaced with an alternative software process, e.g. the rational unified process (RUP) (Kruchten 2004) or some company-specific software process model. Nevertheless, to meet the requirements of public domain customers and to determine strengths, weaknesses, and possible extensions of the process applied, one must be able to compare the VM XT and the alternative software process model (Rausch *et al.* 2005).

The following sections elaborate the need from project practice for process model support: (i) the translation of stakeholder value propositions to IT requirements; (ii) mapping approaches to compare software process models, and (iii) process 'product lines' for systematic customization and evolution of process frameworks such as the VM XT.

3. BUSINESS VALUE TRANSLATION AND STAKEHOLDER COLLABORATION

A key element of *value-based software engineering* (*VBSE*) (Biffl *et al.* 2005, Boehm 2003) is to identify success-critical project stakeholders, to elicit and negotiate their main value propositions, and to translate them into IT requirements and engineering solutions. Software process models usually concentrate on providing a framework for project organization. Roles in the software process define responsibilities and competence to enact processes. However, there is often little consideration on how to align the needs and competence of the stakeholders who give life to these roles, with the role(s) they should play (Standish Group International 2004). In particular, the project principal or customer is often represented by persons who are not routinely involved in developing software systems. Thus, it takes extra effort to help them play an active and constructive role; particularly at project inception in the context of a 'call-for-tender' that determines the opportunities of value creation in this type of project.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/spip

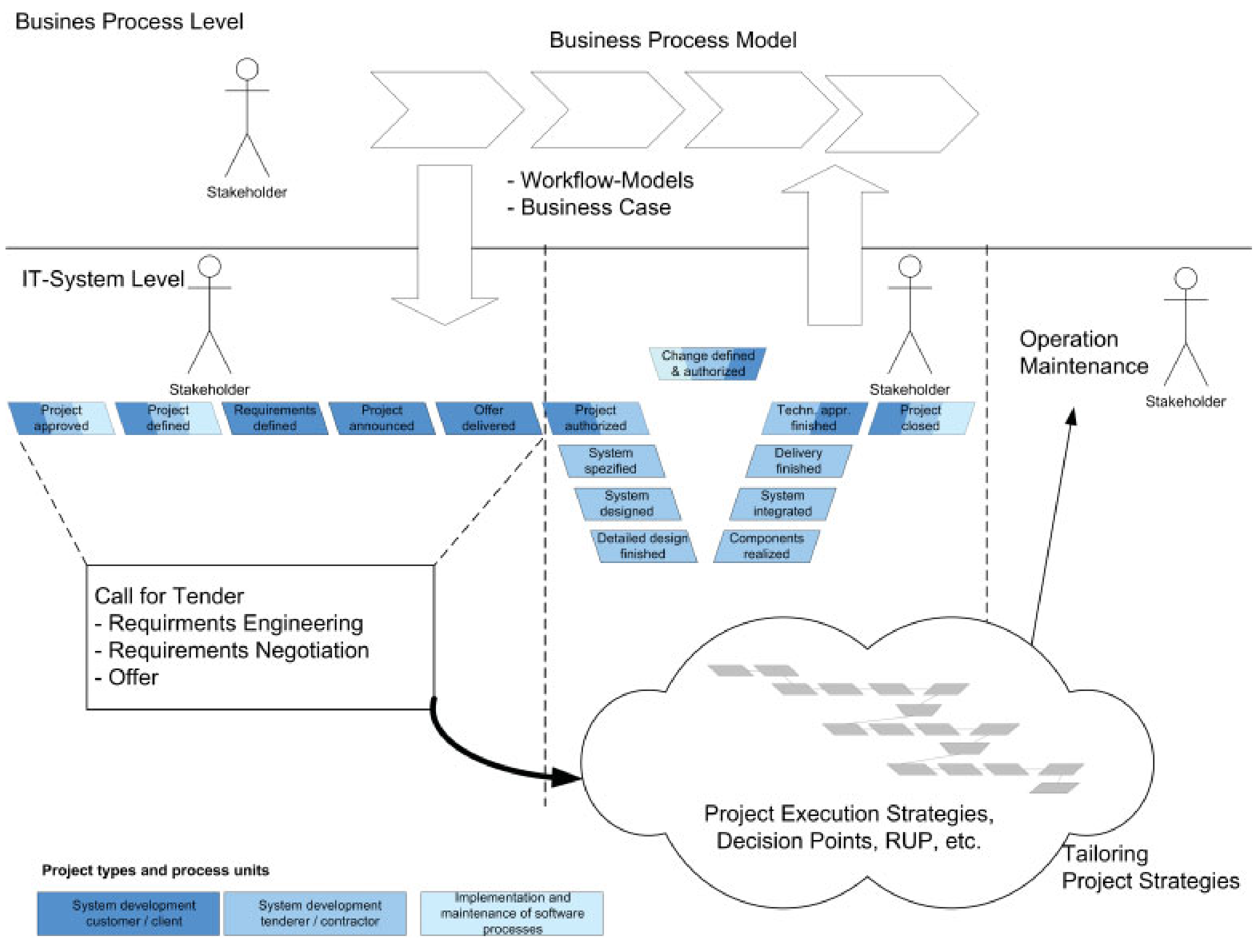

Figure 1. Business process integration and basic components of VM XT

The VM XT is one of the very few process models that explicitly deals with the responsibilities of the customer and thus contributes better by including this crucial stakeholder in the project. However, we see the contribution of the VM XT as a start to better address VBSE considerations; Figure 1 shows the VM XT in the context of value creation: in a business environment, the value that can come from a software development project is generated mostly from the application of the project results on the business process level. Thus, there is a research need to enhance mainly the technical focus of the VM XT with better integration of input from the business process level, e.g. from results of *Promet*, a well-known open-source business process model (Österle and Winter 2003). The business value, expressed in business process modeling, needs to be translated to IT requirements that are the project basis, e.g. in a 'call for tender'. The next step is to develop an engineering solution that fulfills the IT requirements based on a software process tailored to the project needs. Finally, the operation of the IT system supports value creation due to new and improved business processes.

An important aspect of most effective value creation is that the stakeholders understand enough of the software development process options to describe requirements in a way that can be aligned with effective engineering solutions. This requires some way of communicating the scope of engineering solution options during requirements elicitation and negotiation (Schmid and Biffl 2005). Better integration of software process models and business process models can help to open up the currently typical one-way communication into an overall

Copyright © 2006 John Wiley & Sons, Ltd.
DOI: 10.1002/spip

more effective two-way communication between stakeholders who understand the value of business requirements and engineering solutions.

Another challenge of value creation is to realign business requirements and engineering solutions that often evolve concurrently during the course of the project. For reasons such as technical development difficulties, finding inconsistencies of requirements, and differences in the real world, changes to requirements and development plans are common in software projects. Synchronization mechanisms during the project help to assess the need for realignment and provide concrete steps to renegotiate for more consistent products and plans. Software process models propose synchronization mechanisms that work well for in-house projects, such as simple DPs and review procedures. However, collaboration in distributed projects, potentially across several companies and countries, needs more advanced process support (e.g. effective tracing for requirements management and proactive communication between different stakeholders to avoid risky 'hidden consequences', such as interfering project course, with respect to possible changes of their interests) to deal with concurrent work and changes on different levels. The application of stakeholder-related principles from VBSE can help to address these value creation and collaboration aspects in software process models. Empirical analysis of collaboration risks in (distributed, multicompany) software projects can describe the gap between the needs and the currently available solutions in practice. The VM XT provides an advanced software process framework to anchor this research.

4. SOFTWARE PROCESS MAPPING

The major goal of European industrial practice is the production of high-value software in diversified application domains. To meet this goal, several different process models were developed according to domain specific and company related requirements; these approaches led to a huge number of process model variants. Usually, an organizational unit selects an appropriate software process that fits its business best at a certain point in time, and then adapts this process according to the local requirements. As a consequence, a wide range of different software process models, architectures, and methodologies are in use (Bernus *et al*. 1998; 2003). These diverse models are hard to compare (e.g. regarding structure and semantics), which hinders efficient collaboration and software process improvement on a European level. From an academic point of view, systematic mapping of process variants is a prerequisite for a conceptual unification and systematic construction of process models according to predefined needs.

Process managers, especially in the growing segment of multicompany software development see a need for approaches that support stakeholder collaboration, systematic mapping, and transformation of processes to improve their leverage in software process improvement. Process mapping enables collaboration in projects that have to reconcile several process models. VM XT enables tailoring of particular PMs during the project course, e.g. adding individual PMs. This flexibility eases mapping other process models to VM XT (e.g. for organizations that want to use an in-house process model and have to show compliance with the VM XT).

Further, VM XT can act as a common representation for several other process models and support more efficient mapping between any of these models (e.g. for process managers in multicompany projects, who have to derive a common process model that fits the project partners' needs and competences); in many cases today a time consuming task that comes with many new projects.

Mapping to compare software processes requires the investigation of the specific software process model according to components, internal structure, products, activities, etc.

Figure 2 depicts the basic schema for software process mapping, based on VM XT as the bridging

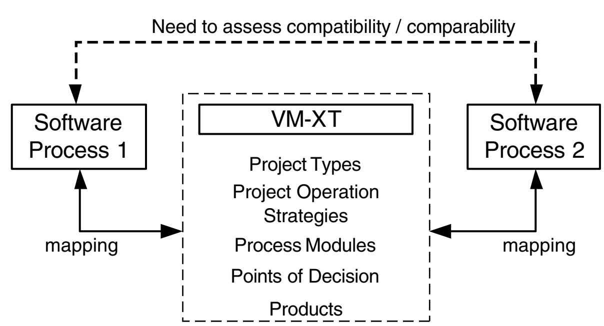

Figure 2. Basic process mapping approach for comparability and compatibility

Copyright © 2006 John Wiley & Sons, Ltd.

Softw. Process Improve. Pract., 2006; **11**: 229–238
DOI: 10.1002/spip

intermediate process model. Following this mapping schema, the analysis of a software process consists of two phases. The first phase is to map each software process to the VM XT in three steps: (i) *structure analysis:* identify basic components of a process model, (ii) *initial mapping:* compare this structural information to VM XT elements according to syntactic and semantic elements in order to find similar elements, completely different elements, and mixed elements; and (iii) *refine mapping:* continue to analyze elements with status 'mixed' on a finer level of detail until the complete software process has been successfully mapped or a defined level of detail is reached. The result set contains a set of matching process elements and opens issues for further investigation. This analysis allows determining the degree of compliance of the software process with the VM XT structure.

The second phase analyzes the compatibility of the two software processes and builds on the results of the first phase. For each structural element, the process manager has the following options: Choose one of the elements of the process models or to choose an element of the VM XT. Using the VM XT as the intermediate model has two major benefits: (i) the VM XT provides a consistent structure for the comparison and (ii) if the manager needs to compare many process models over time, the effort for structural analysis grows in a linear rather than an exponential way. Following this mapping approach, small and medium enterprises (SMEs) can adapt their software processes according to VM XT more easily in order to achieve competitiveness (e.g. to bid for public IT projects in Germany) and to produce software in collaboration with large companies.

Taking a broader view on software process improvement, quality management systems often provide a company-wide framework for continuous process improvement, e.g. CMMI (Chrissis *et al*. 2003, http://www.sei.cmu.edu/cmmi, Kneuper 2003), SPICE (ISO/IEC 15504) (ISO/IEC 15504-x, Loon 2004, http://www.sqi.gu.edu.au/spice/), and ISO 9000 (ISO 900x 2000). VM XT and its derivate process models may be embedded within a quality management system as a central business process for software construction. Because of these integration possibilities, VM XT fits well in to quality management systems, including bidding, parts of purchasing processes, and commissioning.

Initial research using this process mapping approach identified the challenges of how best to represent the mapping results and whether a simple model can cover all major process models. These difficulties point out the broader need for a consistent description of software engineering and software process semantics, e.g. building up an ontology that helps to converge the semantic variants and document differences in meaning between process descriptions.

5. EXTENSION OF THE VM XT FRAMEWORK TO PROCESS 'PRODUCT LINES'

Many projects have to consider domain-specific business requirements, e.g. dependable systems, financial transaction services, security-related applications, and embedded systems (Klemen and Biffl 2004). Consequently, process managers see a strong need to describe domain-specific software processes (see also approaches in CMMI (Chrissis *et al*. 2003, http://www.sei.cmu.edu/cmmi/2005, Kneuper 2003)).

There are several dimensions of formal requirements, such as regulations from business domains, national standards, and company procedures, which need tailored software process models. Currently, such tailoring occurs typically in an *ad hoc* fashion. Typical follow-up problems from this approach are limitations of comparability, high effort for maintenance, and architectural deterioration, very similar to *ad hoc* reuse in software engineering. The flexible structure of the VM XT makes it well suited to derive a wide range of software process variants. For deriving future software processes, the VM XT can provide a common starting base, similar to a software library.

A major development step would be to take up the product-line paradigm from software engineering to software process modeling. This approach allows capturing the variability of software processes on domain and company levels. Such a framework can provide decision making processes for project and process managers to derive an appropriate software process model to achieve robust and high-quality projects. Figure 3 illustrates decision levels of a product-line approach based on VM XT. The model consists of four levels: (i) *generic level:* the basic VM XT, (ii) *business level*: describing

Copyright © 2006 John Wiley & Sons, Ltd.
DOI: 10.1002/spip

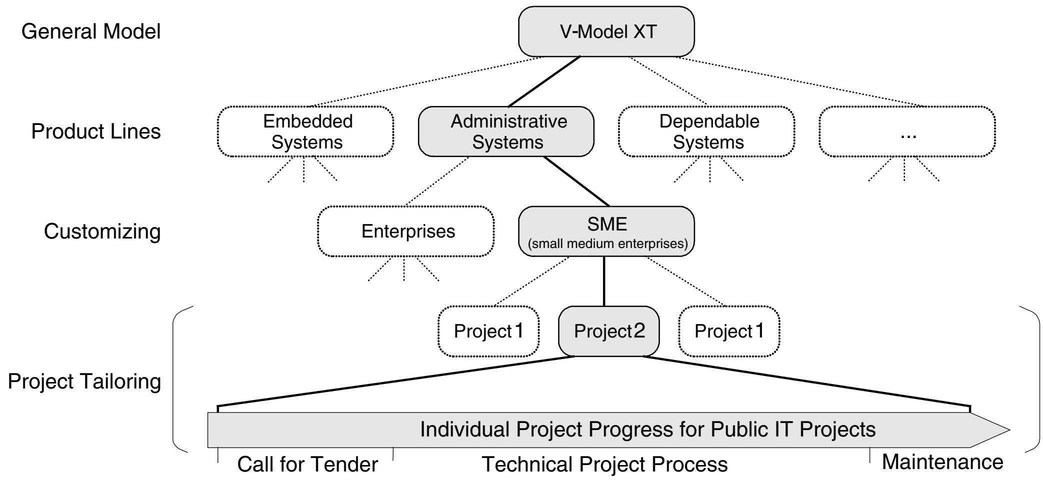

Figure 3. Product line configuration

VM XT variants for various application domains, (iii) *company level:* representing customizing of VM XT approaches; and (iv) *project level:* including tailoring for specific projects within an enterprise. Process product lines – a set of software processes derived from the basic process model – promise a repeatable and efficient configuration of process models with respect to specific business domains and application areas, such as dependable systems, embedded systems, and administrative systems (Abran and Moore 2004). Further benefits come from integration of state-of-the-art techniques and best practices due to comprehensive tools and method support.

We see the benefits from an extension of VM XT to product lines in three ways: such an extension (i) allows to encompass different application domains, (ii) more systematic insight in the courses of projects, and (iii) a larger set of predefined methods and tools. Project and process managers may choose from a pool of application specific VM XT variants (broader view) for special purposes, e.g. web applications, embedded systems (comparable to the Siemens PSE SEM process family)[3]. Therefore, preliminary customization of the basic process for individual application is necessary. This will lead to a set of process model approaches including a domain-specific selection of PMs of the basic VM XT. The second benefit comes from a deeper insight into software process enactment, i.e. the software process enhancement toward a more detailed view on specific PMs. This approach enables a better guidance through the software development process. The third benefit comes from method and tool support of VM XT. The current version of VM XT provides a small set of methods and tools for application. To follow best-practice approaches, project managers need tool and method support including guidelines on which individual method or tool to select. Product-line guidelines support the systematic creation of processes and tool support as they provide the ground to define a 'market' for these products; further, they help project and process mangers to select appropriate methods and tools from the 'market'.

In the basic VM XT framework, software process improvement approaches are derived from the underlying model (including continuous improvement with respect to further development and improvement of the VM XT basic model). The VM XT development team provides a VM XT specific editor to customize the basic definitions of the software process model according to the individual requirements of an organization. Additionally, customized software process models may be extended with respect to special needs of the process users for state-of-the-art approaches and new methods and techniques. Extensions may include additional modules, products, DPs, new project operation strategies, additional methods and tools. Therefore, this concept allows the construction of VM XT variants with respect to different application

[3] http://www.pse.siemens.at

Copyright © 2006 John Wiley & Sons, Ltd.

Softw. Process Improve. Pract., 2006; **11**: 229–238
DOI: 10.1002/spip

and business domains, other types of software processes, and process improvement approaches. With such a product-line approach it is possible to gather information on decisions in customizing a process model and to systematically evaluate and learn from them.

Owing to their potential for learning from projects in similar domains and business areas, product lines promise repeatable and efficient configuration of process models. The product-line approach supports (i) systematic investigation of best-practice approaches to software construction and at the same time (ii) provides a structure to document process variability and capture the lessons learned.

Best-practice approaches should be used to extend the basic model and all derived business domain-specific versions of VM XT product-line approaches. Because of the basic model structure (XML data that describes process elements), experiences and empirical evaluation data can be used for training proposes, e.g. to set up an e-learning strategy. Product lines allow the consideration of national and domain-specific regulations and experiences. Therefore, VM XT and the proposed product-line concept provide a promising direction towards a broader usage of VM XT on a European level.

Research is needed on how best to structure the product-line refinement concept: what should be invariants, and what are the variants. An empirical analysis can determine types of software projects that would be best addressed with the VM XT product-line approach. While the product line can provide the structure of a derived software process model, a challenge can be that the derived processes still exhibit too many degrees of freedom to effectively support a project manager. The support for process enactment with templates and examples needs interfaces to provide feedback from project practice. A good example for successful process support on this level is the Siemens Austria PSE *stdSEM process family*, which consists of six major processes for project types, and examples for tailoring to support the project manager at work.

6. SUMMARY AND OUTLOOK

The mandatory nature of the VM XT for national public domain IT-Projects provides a significant impulse to the German software development scene: As consequence, all organizations have to meet the requirements of this process model. As Germany is an important industrial partner for many European countries, and as a market specifically targeted by companies in EU New Member States, the VM XT is likely to become a leading software process model with the opportunity to grow to a European software process approach. The current version of VM XT is available in German and English to help spread VM XT across Europe.

During a workshop at the Vienna University of Technology in April 2005, researchers from several European countries found in the VM XT potential for proposing a common European process model framework to support systematic software development and to build up empirical know-how on the best practices in Europe. The structure of the new software process and the simple implementation within a company enable SMEs to better meet competitive and demanding bidding requirements in the software market across European national boundaries, e.g. for public projects in Germany.

In 2005, the Austrian Computer Society[4] established a special interest group on 'Software Processes' with a focus group on 'Process Improvement with the V-Modell XT' in cooperation with the Vienna University of Technology and the University of Linz. A major goal of the special interest group is the application and advancement of process models compatible with the VM XT framework according to research topics, practical application, including knowledge acquisition and transfer. Critical success factors for software process improvement are (i) balancing stakeholder interests by translation of business value into appropriate software engineering solutions, (ii) integration of best-practice approaches, and (iii) development of product lines to software processes to capture the variability of software processes on domain and company levels and thus systematically investigate best-practice approaches to software construction. A combination of the top-down product-line approach with the bottom-up input from currently used processes (using the mapping and synchronization mechanisms) can get the most from both sides: systematic derivation of the software process framework combined with practical know-how from concrete

[4] http://www.ocg.at

Copyright © 2006 John Wiley & Sons, Ltd.
DOI: 10.1002/spip

project work. The concepts of the new VM XT support these critical success factors as evolutionary parts of the VM XT framework.

The strategic positioning of the VM XT framework and its strengths to support process mapping and derivation hold the opportunity to provide important impulses to software process improvement on a European scale, especially for the integration of software development efforts in the EU New Member States, and thus can make a significant contribution to realize major goals of the EU Lisbon declaration.

ACKNOWLEDGEMENTS

We want to thank the participants of the VM XT research strategy workshop carried out at Vienna University of Technology in April 2005 and our research partners at Czech Technical University for their valuable discussion and comments to the research agenda presented in this paper.

REFERENCES

Abran A, Moore JW (eds). 2004. *Guide to the Software Engineering Body of Knowledge*. IEEE http://www.swebok.org.

Bernus P, Mertins K, Schmidt G. 1998. *Handbook on Architectures of Information Systems*. Springer: Berlin, ISBN: 3-540-64453-9.

Bernus P, Nemes L, Schmidt G. 2003. *Handbook on Enterprise Architectures*. Springer: Berlin, ISBN: 3-540-64453-9.

Biffl S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (eds). 2005. *Value-Based Software Engineering*. Springer Verlag: Heidelberg, Germany.

Boehm BW. 2003. Value-based software engineering. *Software Engineering Notes* **28**(2): p1–12. ACM SIGSOFT.

SEI. 2005. *Capability Maturity Model Integration*, Carnegie Mellon University, Software Engineering Institute: Internet: http://www.sei.cmu.edu/cmmi/, 2005.

Chrissis MB, Konrad M, Shrum S. 2003. CMMI. Guidelines for process integration and product improvement. ISBN 0-321-15496-7. Boston. Addison Wesley.

El-Emam K, Drouin J-N, Melo W. 1997. Spice. The Theory and Practice of Software Process Improvement and capability determination. Los Alamitos, IEEE Computer Society Press.

ISO 900x. 2000. International Organization for Standardization.

ISO/IEC 15504-x. Software Process Improvement and Capability Determination. Part 1–7, Standard, International Organization for Standardization.

Klemen M, Biffl S. 2004. Economic aspects and needs in IT-security risk management for SMEs, *Proceedings of the 6th International Workshop on Economics Driven Software Engineering Research (EDSER-6) at the International Conference on Software Engineering*, Edinburgh, Scotland.

Kneuper R. 2003. *Capability Maturity Model Integration*. ISBN 3-898-64185-6. dpunkt verlag.

Kruchten P. 2004. *The Rational Unified Process–An Introduction*, 3rd edn. Pearson ISBN 0-321-19770-4.

Loon Hvan. 2004. *ISO/IEC15504-Process Assessment Standard*. Springer, ISBN 0-387-23172-2.

Österle H, Winter R. 2003. *Business Engineering*. Springer. Auf dem Weg zum Unternehmen des Informationszeitalters. Berlin, ISBN 3-540-26405-1.

Rausch A, Bergner K, Höhn R, Höppner S, Wetzel H. 2005. *Das V-Modell XT, Grundlagen*, Tailoring, Anwendung. Springer: Berlin, ISBN 3-540-26405-1.

Roel W. 1995. *Requirements Engineering*. John Wiley & Sons Ltd.

Schmid K, Biffl S. 2005. Systematic management of software product lines. *Software Process Improvement and Practice* **10**(1): p61–76. John Wiley & Sons Ltd.

Sharp H, Finkelstein A, Galal G. 1998. Stakeholder identification in the requirements engineering process. In *10th International Workshop on Databases and Expert Systems Application*, Florence, Italy IEEE Computer Society Press: 387–391.

SQI 2005. *Software Process Improvement and Capability dEtermination*. Griffith University, Software Quality Institute: Australia, Internet: http://www.sqi.gu.edu.au/spice/, 2005.

Standish Group International. 2004. Chaos Report.

V-Modell. 1997. http://www.v-modell.iabg.de.

V-Modell XT. 2005. http://www.v-modell-xt.de.

Copyright © 2006 John Wiley & Sons, Ltd.

Softw. Process Improve. Pract., 2006; **11**: 229–238

DOI: 10.1002/spip