Abstract: Nowadays, at least two types of software development processes exist. At first there are the formal, structured ones mainly based on waterfall- or spiral-like processes. On the other hand, pragmatic agile methods, like Scrum or Extreme Programming, become more and more important. People often say both approaches are exclusive. In this paper we present an approach to integrate a concrete agile method, the Microsoft Solution Framework 4.0 for Agile Software Development, into a precise, well defined and structured software development process model – the V-Modell XT.

Keywords: V-Modell XT, Microsoft Solution Framework, agile methods, process models, process engineering.

1. Introduction

Many projects fail of suffer in time or completeness [5]. Modern process models address special needs in state of the art technologies and methodologies putting them together in frameworks for operating complex projects. Two different types of process models are competing, today. On the one hand are the well known and widely accepted structured approaches, mainly based on waterfall- or spiral-models [1]. But during the past years, agile approaches [2] became more and more popular. Faster, cheaper and more customer-oriented – this is the philosophy of these approaches.

This paper contains the analysis of two process models: the V-Modell XT [7] and the Microsoft Solution Framework (MSF, see [6]). Both are process models appropriate for software development. The V-Modell XT is a generic, strongly structured process model, based on the German V-Modell 97. The MSF – especially the derivate for Agile Software Development – addresses software development for small teams. Both process models are based on a XML metamodel. Both define activity-, work-item- and role models, as well as being shipped with templates and guidance.

In several pilot projects [3] we have already worked with the V-Modell XT and learned that the generic nature of the V-Modell XT needs an embodiment with additional, concrete methodologies (e.g. the MSF Agile) for certain tasks. The integration of such a concrete methodology in the V-Modell XT, its requirements, pros and cons are the topics of this article.

We first give a brief introduction in both process models. Secondly we pay attention to the metamodels looking for integration capabilities and present a way to integrate the MSF Agile with the V-Modell XT by discussing two possible approaches.

2. The V-Modell XT and the Microsoft Solution Framework summarized

In this section we want to give a brief introduction into the V-Modell XT and the Microsoft Solution Framework. We will focus on the basic concepts of both approaches.

Since the MSF consists of different process model subtypes [6], we will only look at the sub-model for agile software development (MSF Agile).

2.1 The V-Modell XT – Overview

The V-Modell XT is a modular process model, obligatory for the German ministry of the interior and the federal armed forces. It’s a generic process model addressing different types of system development projects. The V-Modell consists of process modules (Fig. 1), similar to project disciplines, e.g. project management or software development. Each process module defines all products, roles and activities required for a particular discipline.

Process modules describe the what- and the who-dimension of a project but don’t dictate any order of execution. For the structured execution, strategies for project operation (SPO) are defined. They consist of a set of so called decision points (DP).

A decision point is a milestone and a quality gate, defining a set of products that have to be finished and quality-assured at a defined date. An operation strategy not only defines a set of decision points, but also an order over them (when-dimension).
Fig. 1. The V-Modell XT metamodel is showing all fundamental concepts. The grey-colored boxes show basic concepts; the non-colored boxes show additional concepts.

One of the most popular capabilities of the V-Modell XT is the automated tailoring-mechanism which enables a project manager to trim the whole V-Modell to his special project’s requirements. So for example if software is the matter of a project, no hardware-specific process elements are contained in the tailored, project-specific V-Modell.

Fig. 2 sketches the tailoring concept of the V-Modell XT. At the beginning of a project, the project is characterized by several so called project attributes. On the basis of these attributes, necessary activities and products for the specific project are identified. The resulting project specific V-Modell in general consists of a subset of the available process modules of the complete model. Thus only project relevant activities will be carried out within a project.

The V-Modell is written in XML. So it is easy to extend. New process modules can be added or existing ones can be altered. This means e.g. adding new products, new roles, new SPO’s and so on. Every element defined by the metamodel (Fig. 1) is available for extension and customization. Changes made to the V-Modell are available for Tailoring in-time, meaning if new items are added or existing ones are altered, supporting tools [9] can apply these changes immediately.

2.2 The Microsoft Solution Framework – Overview

The Microsoft Solution Framework 4.0 (MSF) is a process framework similar to the V-Modell XT addressing software development. This process definition is available since 1993 and was condensed from many years of software development and infrastructure deployment. The former release [10] 3.0 and 3.1 of MSF contained Team and Process Models and several Disciplines as core components. Furthermore an interface for connecting with Microsoft Operations Framework (MOF) was available [10, 11]. The latest version 4.0 is the first fully tool-integrated MSF release. It is contained as integrated process in the Team Foundation Server, which is a new backend component for the Visual Studio development environment.

The MSF contains a set of fundamental principles: a team model, a process model and disciplines. The basic process
model is an iterative, governance-oriented approach. The structure and the contents of the MSF are based on a set of principles of a particular project. Some – but not all – principles are: “Stay agile and expect change” or “Always create shippable products” [4].

As shown in Fig. 3, the MSF only contains three basic elements: Roles, Activities and Work Streams. Roles are defined in a non-hierarchical team model - [4] speaks of peers. Work streams are assigned to roles. A work stream is a sequence of activities, such as working on the project plan. Each activity has products and roles assigned. The MSF Agile includes a simple, spiral-like iterative model for project operation.

2.3 Positioning the V-Modell XT and the MSF

After having introduced both process models, we are going to station them. The V-Modell XT is a generic process model. It is intended to address a wide audience and waives concrete methodologies. The V-Modell can be tailored to fit specific needs of a project (e.g. software development, embedded system’s development). But the tailored, project-specific V-Modell does not say anything about how to develop software for instance. A concrete methodology is needed at this point. To point it out again: The V-Modell provides a framework at the process level that gives guidance with respect to organizational questions and management. Further aspects, e.g. software development are covered on organizational levels, too. But there is no concrete methodology, e.g. Object-Oriented or Component-based Development implemented.

The MSF Agile is a highly specialized process primary covering agile software development using the Visual Studio 2005 (Team System) [4, 8], where it is directly integrated. On [6] a stand-alone version of the process guidance is available, too. So the MSF can be applied in a project without using the Visual Studio tools. This guidance contains all necessary concepts and descriptions for implementing the MSF in a software development project.

The MSF is a concrete methodology required for embodiment of a project-specific V-Modell project. Table 1 briefly compares both process models referring to some common criteria relevant for possible process integration.

3. Integrating the MSF and the V-Modell XT

In this section we give a summary of connection points appropriate for integrating the MSF with the V-Modell XT. After collecting information, we want to sketch out an integration process and describe a prototype.

Two ways for integration are possible. Extension bears all aspects extending the V-Modell without alteration of previous contents. This ensures strict compatibility. The second way, Specialization, alters the V-Modell by changing concepts, artifacts and the terminology to be used. This section mainly targets conceptual design questions without providing concrete solutions. A prototypical implemented scenario shows the possibilities and identified connection points in practice.

Table 1. Elements and concepts of both process models compared.

<table>
<thead>
<tr>
<th>Element/Concept</th>
<th>V-Modell XT</th>
<th>MSF Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team and People</td>
<td>Role model</td>
<td>Team Model</td>
</tr>
<tr>
<td>Activities</td>
<td>Abstract descriptions, using steps and activity diagrams</td>
<td>Work streams, activities and work items. Clearly defined using step-by-step instructions; tool-supported</td>
</tr>
<tr>
<td>Results</td>
<td>Products (documents, codes, etc.)</td>
<td>Work products (documents, models, codes, programs etc.)</td>
</tr>
<tr>
<td>Assistance</td>
<td>Process description, samples, specialized generated document templates</td>
<td>Step-by-step instructions, specialized documents, utilized project operation</td>
</tr>
<tr>
<td>Documentation</td>
<td>The V-Modell XT itself</td>
<td>Process guidance as HTML-based documentation (stand alone and integrated in project/team)</td>
</tr>
<tr>
<td>Tool support</td>
<td>OSS, commercial: for process authoring and project operation</td>
<td>Integrated in the Visual Studio 2005 Team System, some OSS tools covering authoring, export etc.</td>
</tr>
</tbody>
</table>
3.1 Analyzing the Metamodels

Both process models are based on a XML-metamodel. Aspects affected for integration are the iterations, roles, work item descriptions and the process guide.

1st Way – Extension. The V-Modell contains the capabilities for integrating additional functionality. By adding additional process modules, new products, roles and activities may be introduced. New operation strategies can be added as well. An additional process module addressing the MSF contents might contain the role release manager, the product prototype and the topic scenario. This first way is described in the V-Modell and should be the preferred one if compatibility with V-Modell XT is required.

2nd Way – Specialization. The second way is to alter the V-Modell XT itself. This means creating an organization-specific V-Modell XT derivate. An additional process module has to be created, too. But MSF contents, comparable to existing V-Modell ones (Table 2 and 3), can be merged directly. As a consequence, standard V-Modell contents may be replaced by MSF contents, but some generic contents might be lost.

3.2 Connecting Project Operation Strategies

The first step of integration is the process itself. On the operational level, project managers have to deal with timeframes, budgets and resources. So organizing and guiding a project is very important. That’s why we will focus on the process at first. The V-Modell XT provides different operation strategies, e.g. for incremental, component-based and agile development. Fig. 4 shows the stated order of decision points (refer to Fig. 1) for the agile operation strategy. The upper part of Fig. 3 shows the integration of the MSF’s iterations [4] in the V-Modell’s operation strategy. The MSF is a
development-focused model and is therefore not supporting offering and contract conclusion. Thus the integration starts at the decision point iteration planned. The organizational MSF element Project Set Up is defined by the surrounding V-Modell. Development activities from MSF can be found around the decision points system elements realized and system integrated, where software-units and releases are created. The decision points system integrated and delivery completed provide the test results necessary for the MSF testing activities and the feedback. Feedback also comes from a change list/change set, created in the decision point acceptance completed. Finally the product’s release is carried out in accordance to the V-Modell XT.

Since the V-Modell operation strategy is rather a prototyping background than an agile, we also take a pure iterative approach into account as well (Fig. 4). Having a closer look on both competing approaches, the second one (iterative) should be preferred for integration. Conceptually it is the same, because the set of available decision points is defined by the V-Modell. Strategies for project operation only define an order over this set.

Since the V-Modell’s operation strategy is used for planning a project, the MSF iteration model has to be integrated. So a new operation strategy “System Development using MSF Agile” has to be introduced. The new operation strategy has to implement the MSF iteration model, e.g. based on the operation strategy for iterative system development.

Furthermore; despite of a set of V-Modell documents, Visual Studio models and their documentation have to be handled in the decision points of the new operation strategy, because the generated code based on the models is the result if using the MSF. This means that some V-Modell architectural documents may be replaced by Visual Studio Models (System Definition Model, Logical Data Center, Class etc.).

### 3.3 Connection on the Team Level

Both process models define a team or role model containing roles participating people should have. Since the MSF Agile only knows 6 roles [4], not all roles defined by the V-Modell are necessary or required separately if using MSF. The most important roles are listed in Table 2. The V-Modell roles software developer and system integrator for example coincide to the MSF developer. A role that is not explicitly defined is the MSF release manager. Other V-Modell roles like the project’s merchant are substituted in the MSF project manager. Only one new role has to be added, but other roles have to be restructured or merged on the V-Modell-side for the MSF’s integration. The MSF as well as the V-Modell contains profiles describing the people holding particular roles. These profiles have to be integrated. Extending the V-Modell by adding a new process module means to summarize all roles in the convention-part of the V-Modell and add the release manager as new role.

#### Table 2. Role model comparison and integration

<table>
<thead>
<tr>
<th>V-Modell XT Roles</th>
<th>MSF Agile Roles</th>
<th>Responsible for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project manager</td>
<td>Project manager</td>
<td>Planning the project, Guide project</td>
</tr>
<tr>
<td>Requirements analyst</td>
<td>Business analyst</td>
<td>Capture project vision, Define requirement-oriented work packages</td>
</tr>
<tr>
<td>Software developer</td>
<td>Developer</td>
<td>Development and bug fixing tasks</td>
</tr>
<tr>
<td>System integrator</td>
<td>Architect</td>
<td>Architectural tasks</td>
</tr>
<tr>
<td>System architect</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Software architect</td>
<td>Tester</td>
<td>Testing tasks</td>
</tr>
<tr>
<td>Reviewer</td>
<td>Release manager</td>
<td>Product’s release</td>
</tr>
</tbody>
</table>

#### Table 3. Product comparison and integration (extract). V-Modell products are assigned to decision points (DP) and stationed to corresponding MSF products.

<table>
<thead>
<tr>
<th>V-Modell XT products</th>
<th>MSF Agile products</th>
<th>Product required in process stage/associated activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management documents</td>
<td>Vision statement</td>
<td>Project initialization and definition stages Management, e.g. “Capture project vision”</td>
</tr>
<tr>
<td>Project plan</td>
<td>Iteration plan</td>
<td>DP: Iteration planned Planning, e.g. “Plan an iteration”</td>
</tr>
<tr>
<td></td>
<td>Release plan</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project checklist</td>
<td></td>
</tr>
<tr>
<td>Software units</td>
<td>Code</td>
<td>DP: System elements realized Development, e.g. “Implement a development task”</td>
</tr>
<tr>
<td></td>
<td>Prototype</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unit test</td>
<td></td>
</tr>
<tr>
<td>Software architectures and specifications</td>
<td>Class diagram, QoS List, Scenario list, Storyboard Project release Test result</td>
<td>DP: System integrated Development and Releases, e.g. “Build a product”</td>
</tr>
<tr>
<td>Review spec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Realized software systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formal acceptance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptance Changeset</td>
<td></td>
</tr>
</tbody>
</table>


The convention-part is a specialty of the V-Modell supporting the mapping from other standards to the V-Modell. The second way (specializing) means renaming and extending the existing role profiles of the V-Modell with the ones from MSF and adding the new role release manager.

3.4 Connection on the Product Level

Both process models define some kind of products (see Table 1). If extending the V-Modell, overlapping products have to be included in a convention-part. New products/topics have to be defined in a new process module.

In Table 3 we provide a short comparison idea – some new products and topics are introduced, others are obsolete. For example the products prototype and scenario as typical agile-driven products/topics have to be introduced. Other V-Modell products, defined in a generic way, are specialized in the concrete MSF, e.g. functional and non-functional Requirements (V-Modell) are substituted by scenario, storyboard and QoS requirement.

Architectural products in the V-Modell can be found in several hierarchical specification documents. The V-Modell defines a so called system/product creation structure by default, which represents a hierarchical decomposition of a system to subsystems and so on. Every level in this hierarchy is represented by a set of products containing a system’s design and specification.

The MSF targets another kind of system development. Because of the tool integration, the MSF not only is an agile methodology, but a model-driven, too. The architecture of a system is stored in a set of models (datacenter or class models). These models and their documentation will complete or substitute the V-Modell architectural products. Adequate mappings between the design documents and the models have to be found.

3.5 Prototype

In the previous sections we analyzed connection points and provided examples and discussions. The next step was to implement a demonstrator.

We first identified the special setting for the implementation. Starting points were the current V-Modell XT (release 1.2, downloadable from [7]) and the Microsoft Solution Framework 4.0 for Agile Software Development (available at [6] or in Team Foundation Server – Process Template Manager, respectively). The goal was to integrate the MSF into the V-Modell. We use the V-Modell XT Editor [9] as authoring-tool to punctually extend the V-Modell with selected MSF contents.

As mentioned earlier, extension means adding a new process module, containing products, roles etc. and connecting it to the existing V-Modell XT contents.

Creating a new process module. The first step for the integration is the creation of a new process module called “SW-Development with MSF”. The new process module is not a part of the so called V-Modell Core (the V-Modell XT Core [7] contains process modules mandatory for all projects), because it is only necessary for software or software-related development projects.

Since the V-Modell already contains some structure and content related to system development, the MSF concepts are add-ons. Common structures are implemented in the process module System Development, which is now the basis for the new process module. Thus the added process module is stationed on the same level as the standard SW-Development process module defined by the V-Modell itself.

Creating new elements. The second step is the integration of the selected products, roles and activities. We chose two scenarios to implement for demonstration, as shown in Fig. 5: 1). Release Management 2). Scenario-based requirements engineering

The first scenario introduces the MSF role Release Manager and the product Release Plan and the corresponding activity Planning Releases. With respect to the V-Modell, this is an addition of new contents.

The second scenario does not introduce a new role but refers to roles already defined in the V-Modell XT. This scenario introduces the product Scenario and the activity Brainstorm Scenarios.

Both newly introduced products are located in product groups already defined by the V-Modell. Hence, no product group (Produktgruppen) has to be created.
tors. Selecting these properties during the tailoring of the V-Modell enables the process module’s availability. The 2nd point to define the further application is the V-Modell project type (Fig. 5: “Projektypen”) definition section. Selecting the project type definition for contractor’s system development projects, the new process module has to be added as optional process module.

3.6 Application in a Project

Up to this point, new contents were introduced and basic connections to the V-Modell are made. When using the V-Modell, this is the very first step. The next one is to tailor the whole process model to fit a concrete project’s requirements. On [9] the V-Modell XT Project Assistant is available as reference implementation of tailoring tools.

Fig. 6. After inserting the defined criterions, the new process module is available for selection, immediately.

Having changed the V-Modell as described above, the changes are available to the tool immediately (Fig. 6). During the tailoring process, the new process module can be selected. This includes all products, roles and activities defined there in the project specific V-Modell. The steps of the tailoring are completed.

Workstreams. A workstream is a concept the V-Modell does not know, directly. It is a group of activities that flow together. But this is not an activity group as defined by the V-Modell, which is only an instrument for structuring and putting activities together. Let us give an example. The workstream “Guide Project” contains five activities: “Review Objectives”, “Assess Progress”, “Evaluate Test Metric Thresholds”, “Triage Bugs” and “Identify Risks”. Currently, if including products in the V-Modell, we use these activities for connecting with products. Currently we pay no attention to the workstreams. The V-Modell does not know this kind of activity collection. It defines something similar by its decision points (DP). The project stage that ends with a certain decision point can be seen as workstream. This works e.g. for “Create a Scenario” stage before DP: “System specified”, “Guide Iteration” stage before DP: “Iteration planned” or “Release a product” stage before DP: “Delivery completed”.

If integrating both processes, adequate mappings for the concept workstream have to be found.

The V-Modell XT System Model. Another point that has to be taken into account is the system model of the V-Modell XT. As discussed in section 3.4 the V-Modell defines a hierarchy of products. Specifications, designs and deliverables are defined and connected on several abstraction levels. The V-Modell implements a philosophy of hierarchical decomposition of a system. A system consists of segments or units. A segment consists of segments or units. A unit consists of components, which themselves consists of modules. This is the vertical dimension. Additionally, the V-Modell defines a horizontal dimension, too. The horizontal decomposition leads to the system itself, supporting systems and logistics. This structure is strict.

The MSF has to be integrated with this structure as well. Thus the software system’s design is stored in a set of models, only these models have to be stationed as first. So a Visual Studio “Application Model” might map to the system- or the segment-level, while Visual Studio “Class Models” map to units or components. The horizontal cut can be mapped as well. So a Visual Studio “Deployment Model” can be understood as part of the logistics and so on.

The second question here is the way of creating the required documentation out of the models. [14] describes a possible approach.

Integrating both processes on this level is complicated, because the operational (executable) V-Modell parts (strategies for project operation, Fig. 4) go hand in hand with the system model, described above. The system model is strict and should not be dramatically changed.

Project Operation. The strategies for project operation defined by the V-Modell contain a lot of decision points (milestones). With respect to the MSF Agile, which is an agile method – meaning many, but short iterations – there are too much decision points.

During our analysis we identified two strategies for project operation that might meet the requirements of the MSF Agile. Our favorite, the iterative/incremental approach (Fig. 4), defines at whole 13 decision points and 9 for development and planning tasks per iteration. Having in mind that an iteration of an agile process is regularly between 2 and 6 (max.) weeks in length, this means passing a milestone every 4-5 days (based on a 6-week calculation). This is with respect to quality assurance impossible.

If integrating the MSF, a new strategy for project operation has to be designed and included.
4. Conclusion

In this paper we stationed the V-Modell XT and the MSF 4.0 for Agile Software Development. We briefly introduced the metamodels of both processes and discussed operation strategies, products and roles. Furthermore we presented and discussed ideas and variants for integrating both process models: Extension vs. Specialization.

Extending the V-Modell XT means (simple) adding MSF contents as additional functionality, what would result in redundant, non-optimized products and processes. Specializing on the other hand leads to a highly optimized V-Modell version but may cause serious problems, because branching in derivates complicates versioning and implies other problems.

Nevertheless, integration would provide a benefit: A specialized V-Modell XT including the MSF is compatible with the V-Modell standard and profits from wide tool integration. On the other hand the Visual Studio-integrated software development process MSF becomes compatible to the V-Modell XT. Users also profit from a concrete, slender methodology without generic overhead. If necessary, the MSF itself can be customized, too [12]. An integration, where both processes are adapted to each other, is possible. So the best of the V-Modell can be combined with the best from MSF.

In a small prototype we presented the extension-scenario. We designed some products, activities etc. and demonstrated the way of integrating the new elements in the V-Modell and the in-time availability of the new contents for further work.

Currently some efforts are made to create organization-specific derivatives of the V-Modell XT [3]. Regarding to the German armed forces, a first integration of their “Customer Product Management”-process (CPM) [13] in the V-Modell was successfully performed and finished in the first quarter 2006. Thus the CPM is not a development model, but a management frame for procurement, we looked for a small development model to integrate. Our analysis of the MSF started in mid 2005. The results we’ve got until now and the final availability of the MSF-related tools motivates a wider implementation which has already started in summer 2006.

The first implementation (based on the prototype presented here) only provides an additional process module showing how MSF work items implemented in the V-Modell would look like. The next step starting in autumn 2006 will provide a full integrated MSF Agile process including a wide range of products, activities and optimized strategies for project operation.

At this point, software development using the V-Modell XT can be done either using the standard mechanism (process module: “SW-Development”) or using the new process module “SW-Development using MSF Agile”.

References


Author Bios

Marco Kuhrmann born in 1978. He studied Computer Science at University of Potsdam. Since 2004 he is working as research assistant to Prof. Dr. Dr. h.c. Manfred Broy at Technische Universität München, Institute of Informatics – Software & Systems Engineering where he was a member of the V-Modell XT development team. In the period 2004-2006 he was coaching several pilot projects using the V-Modell. 2005 he was working on the integration of the German federal armed forces management and procurement process “CPM” in the V-Modell XT. Currently he works on process engineering question especially on questions related to process integration, process authoring and construction methodologies and tool support for modular process models.

Thomas Ternité born in 1979, studied industrial engineering with emphasis on computer sciences and is now research assistant to Prof. Dr. Andreas Rausch at the Technische Universität Kaiserslautern. He is a member of the V-Modell XT development team, working for maintenance and further development of the V-Modell XT. 2005 he was working on the integration of the German federal armed forces management and procurement process “CPM” in the V-Modell XT.