# An Agile BPM Project Methodology

Christian Thiemich and Frank Puhlmann

Bosch Software Innovations GmbH
D-10785 Berlin, Germany
{christian.thiemich,frank.puhlmann}@bosch-si.com

**Abstract.** Business Process Management (BPM) has become one of the most important management disciplines in recent years. In reality, however, many technical process improvement projects failed in the past and the expected benefits could not be established. In the meantime, the agile software development movement made massive progress in contrast to classic waterfall approaches which are still the foundational methodologies for many BPM projects. This paper investigates the combination of a traditional BPM methodology and agile software development to overcome the limitations of existing BPM methodologies. The main focus is on projects that cover the technical realization of processes based on modern Business Process Management Systems (BPMS).

## 1 Motivation

Business Process Management (BPM) helps companies focusing on value-adding end-to-end processes and supporting them with IT systems. Hence, it focusses on implementing process-based, long-running and wide-reaching business applications with a specific set of tools (Business Process Management System, abbr. BPMS). Additionally, it supports the organizational change management which is required to successfully roll-out these changes throughout the company.

In some of our projects, however, we discovered that the process implementation the business departments got is precisely what they once required, but not what they need anymore as the requirements changed over time. Traditional software engineering were faced with the same problems but came up with solutions to closely link the customers (e.g. the requirement providers) with the engineers who actually build the systems. The solutions are based around the principles of agile software development, such as "satisfy the customer through early and continuous delivery of valuable software", "Welcome changing requirements, even late in development", or "Deliver working software frequently" (see [3]).

To gain the benefits of the agile principles for BPM projects, we developed a new approach for the realization of BPM projects which is based on the principles of the agile software development manifesto. Therefore we classify the existing artifacts and methods of an existing BPM methodology with a traditional waterfall project execution (named IBPM, see [12]) and reframe the generic parts into an agile development methodology based on Scrum. The derived framework

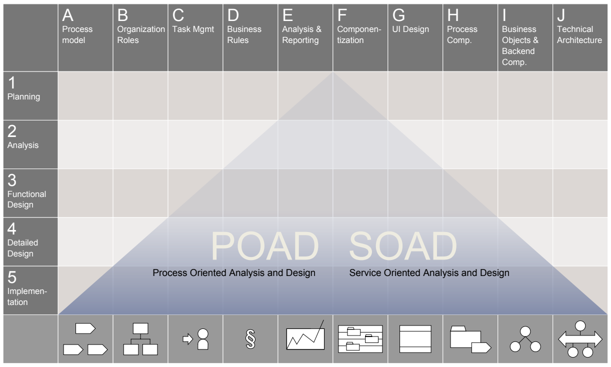


**Fig. 1.** The Integrated BPM Project Methodology Framework according to [12]

builds the foundation for an agile BPM methodology that provides guidance from early BPM project initiatives to the final implementation.

The paper is structured as follows. We start with an overview of the preliminaries in section 2, in particular IBPM as well as Scrum. Section 3 discusses the foundations of agile BPM projects. Besides providing an adaption of the agile principles to BPM, it also provides a guideline whether a BPM project should be implemented with the usage of the traditional lifecycle or by an agile methodology. Section 4 provides the core concepts of the agile BPM methodology, including a formal meta model, the agile lifecycle and key artifacts and methods.[1] Section 5 introduces a practical experience with the application of the agile methodology. We discuss related work in section 6 and conclude in section 7.

## 2 Preliminaries: IBPM and Scrum

This section introduces the basics of the Integrated BPM Project Methodology (IBPM) [12] and the Scrum software development framework. Both approaches have different purposes. While IBPM has a strong focus on analysis and design, Scrum delivers a holistic approach of how to get requirements implemented. IBPM furthermore provides best practices and artifacts to capture and discuss business requirements.

### 2.1 The Integrated BPM Project Methodology

The Integrated BPM Project Methodology enables BPM projects to be accomplished in a structured manner. The authors introduce different characteristics of BPM projects that are distinct to pure software development projects. BPM

[1] A complete description of the agile BPM methodology has been published as a Master Thesis (In German), available online at http://frapu.de/pdf/thiemich2012.pdf.

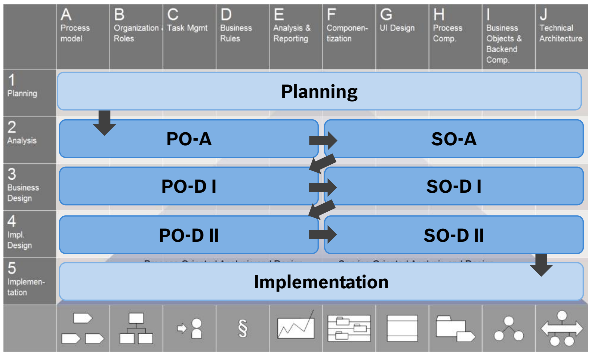


**Fig. 2.** IBPM Project Approach according to [12]

projects are to a large degree organizational projects. The challenge is to separate process flow and decision logic from software development aspects such as functions, thus establishing a well aligned and loosely coupled service-oriented architecture.

The motivation behind IBPM was a desire to increase quality and efficiency as well as to reduce risks and costs by enabling people to ask the right questions and deliver the right artifacts at the right time. For this purpose IBPM uses best practices and consists of the following three core components: IBPM Framework, IBPM Pattern Catalogue, IBPM Project Approach.

**Preliminary 1 (IBPM Framework).** The IBPM Framework, shown in figure 1, defines ten thematic pillars, which combine the most important aspects from the process-oriented and the service-oriented perspective in a BPM project. Both perspectives are represented by five columns. Given that the process model is based on BPMN, it is the leading artifact for the process-oriented perspective. Additional pillars include Organization and Roles (B), User Tasks (C), Business Rules (D) as well as Process Monitoring (E). The same idea can be applied to the pillars in the service perspective. The Componentization (F) contains the leading artifact represented as an SOA-Map as well as four more refining pillars. The Framework is additionally based on five different levels of detail (Planning, Analysis, Business Design, Implementation Design and Implementation). The resulting matrix allows the structured discussion of artifacts and corresponding requirements.

**Preliminary 2 (IBPM Pattern Catalogue).** The IBPM Pattern Catalogue is based on best practices and provides project conventions for eight different aspects with predefined patterns. The patterns include BPMN modeling guidelines, UI templates, or patterns covering change management.

**Preliminary 3 (IBPM Project Approach).** The IBPM Project Approach (figure 2) is based on a waterfall approach and introduces different project phases (Planning, Analysis, Business Design, Implementation Design, Implementation)

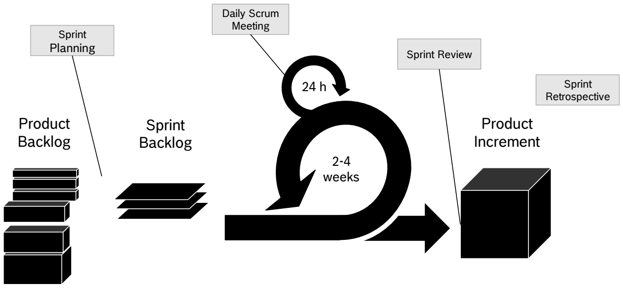


**Fig. 3.** The Scrum Development Process

and working packages, focusing either on Processes or Services with a strong emphasis on the Analysis and Design of a BPM project. IBPM splits the design phase into the Business Design and the Implementation Design. Comparing IBPM to generic project approaches, IBPM adds value by defining BPM-specific tasks, roles and artifacts.

### 2.2 The Scrum Software Development Framework

Scrum (see e.g. [10],[15]) is a lightweight development framework, which is easy to understand and provides defined roles and processes. Even though the framework was born the early 1990s for software development purposes, other disciplines, including general management, have adapted it more and more [4]. It represents the Agile Manifesto[2] and the associated principles by focusing on collaboration and interaction to satisfy the customer. Scrum is based on empirical process control theory and utilizes an iterative, incremental approach to control risks and optimize predictability. The main goal is to develop potentially releasable product increments in short, consistent and time-boxed iterations, so-called sprints.

Scrum is based around three roles (Product Owner, Scrum Master, Team), four meetings (Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective) and three artifacts (Product Backlog, Sprint Backlog, Product Increment). We introduce the product backlog, followed by the other artifacts and roles based on the meetings as shown in figure 3.

**Preliminary 4 (Product Backlog).** A "Product Backlog" describes the requirements for a product prioritized by their business values. In addition each entry in a Backlog is estimated (according to an abstract effort metrics based on "Story Points") and has a set of acceptance criteria.

**Preliminary 5 (Sprint Planning).** Within the first part of a "Sprint Planning" the "Product Owner" presents his goal for the following sprint and his priorities. The team will estimate the presented items and depending on the velocity (e.g. how many "Story Points" a team can currently implement in a

[2] See http://agilemanifesto.org/

predefined time frame), a certain amount of "Backlog Items" are selected for the sprint. In the second part of the "Sprint Planning" those requirements are broken down into "Tasks" which result as entries in the "Sprint Backlog".

**Preliminary 6 (Daily Scrum).** Over a predefined length of a sprint (usually 2-4 weeks) the Team will meet daily for the "Daily Scrum" to synchronize the current work and especially the problems (so-called impediments). The "Scrum Master" is responsible for solving those impediments as quickly as possible.

**Preliminary 7 (Sprint Review).** At the end of a "Sprint" the "Team", the "Scrum Master" and the "Product Owner" will meet again for the "Sprint Review". Based on the requirements and their predefined acceptance criteria the "Product Owner" will approve the requirements in the live system.

**Preliminary 8 (Sprint Retrospective).** The last meeting of a "Sprint" is the "Retrospective" where the "Team" and the "Scrum Master" (sometimes including the "Product Owner") reflect the result of the "Sprint" and define tasks to improve their Scrum process.

Scrum has a number of variants with additional steps or artifacts like Backlog Grooming, a Definition of Ready and a Definition of Done. The Backlog Grooming establishes a new meeting that helps the Product Owner to maintain his Product Backlog. He can get the assistance of the Team to make sure that the items in his Product Backlog are ready corresponding to the Definition of Ready. The Definition of Done is used to define general criteria which are needed in order for each requirement to be approved by the Product Owner.

## 3 Foundations of Agile BPM Projects

This section discusses selected agile principles and how they affect BPM projects. We start by introducing the foundation of agile BPM projects by comparing the idea of BPM and the definition of a project. The idea behind BPM is to provide sustainable and continuous improvements, whereas projects are intended to be unique and to have defined goals.

From our experience, most BPM improvements are motivated by a project and are not continuous at all. BPM aims to make the business processes flexible and to improve them continuously. Even though agile projects are still projects, they have certain advantages in combination with BPM. They lead to an inevitable flexibility for BPM projects which embraces and welcomes changes. Additionally, when most projects start, their goal is often not exactly defined at a detailed levels. Agile approaches cut off the overhead of a multiple-month analysis and design phase and deliver visible results in early phases of the project.

**Best Practice 1 (Motivate the different approaches using the "Magic Triangle").** Use the "Magic Triangle", shown in figure 4, to explain the idea of the paradigm shift that accompanied an agile development approach. While the classic waterfall approach has a fixed project scope, it only delivers assumptions on time and budget. Turning the triangle to agile, in contrast, offers a fixed time and budget, but no limitation or restriction on the scope.

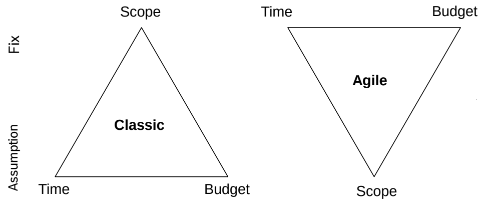


**Fig. 4.** Comparing the magic triangle

Until recently, a project without a clearly defined scope was not acceptable. Nowadays it is recognized that long-running projects with a fixed scope are often not successful and do not lead to end-user acceptance. To overcome this issue, the foundation of agile BPM projects is based on an adaption of the Agile Manifesto and its principles. In the following, the most relevant principles behind the Agile Manifesto are reflected from the perspective of a BPM project. We cite the principles according to [5]:

**Principle 1.** *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*

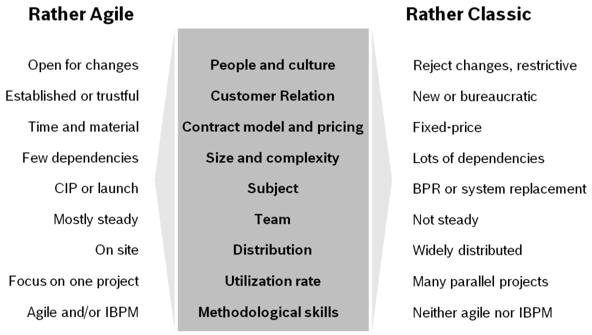
A number of BPM projects have a long analysis and design phase, including late presentation of visible results. Thus, the customer is not able to realign the direction of the project at appropriate times, leading to no continuous delivery at all. BPM projects are often accompanied by coarse-grained requirements. Since BPM and the agile approaches have similar preconditions, further research is necessary on how they fit together. Again, the idea of BPM is that of a sustainable improvement and support of the business' processes and not the accomplishment of a single project. This is a significant problem that is addressed via agile BPM projects.

**Principle 2.** *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*

Why do companies engage in BPM? They expect competitive advantages and want to reduce the time-to-market of their innovations and improvements. Our experience has shown that BPM projects are often delayed due to changing requirements. Agile methods welcome changes to satisfy customer requirements, which have evolved over time. This refines the process implementation, channeling it into the direction to help fulfill the customer's needs.

**Principle 3.** *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference towards the shorter timescale.*

This principle raises the question, 'what does working software mean for a BPM project?' Is it only the "living" process running in the BPMS or does it mean we have rolled out the process throughout the company, including training and documentation? A complete rollout cannot be achieved in one single sprint. In case of a technical BPM project we need to deliver a working process in our BPMS frequently. Depending on the project, we might have multiple releases



| Rather Agile | | Rather Classic |
|---|---|---|
| Open for changes | People and culture | Reject changes, restrictive |
| Established or trustful | Customer Relation | New or bureaucratic |
| Time and material | Contract model and pricing | Fixed-price |
| Few dependencies | Size and complexity | Lots of dependencies |
| CIP or launch | Subject | BPR or system replacement |
| Mostly steady | Team | Not steady |
| On site | Distribution | Widely distributed |
| Focus on one project | Utilization rate | Many parallel projects |
| Agile and/or IBPM | Methodological skills | Neither agile nor IBPM |

**Fig. 5.** Project parameters

every 5–10 sprints. In this case it is important to insure that the whole lifecycle of a process has been completed (incl. test/rollout/training).

**Principle 4.** *Business professionals and developers must work together on a daily basis throughout the project.*

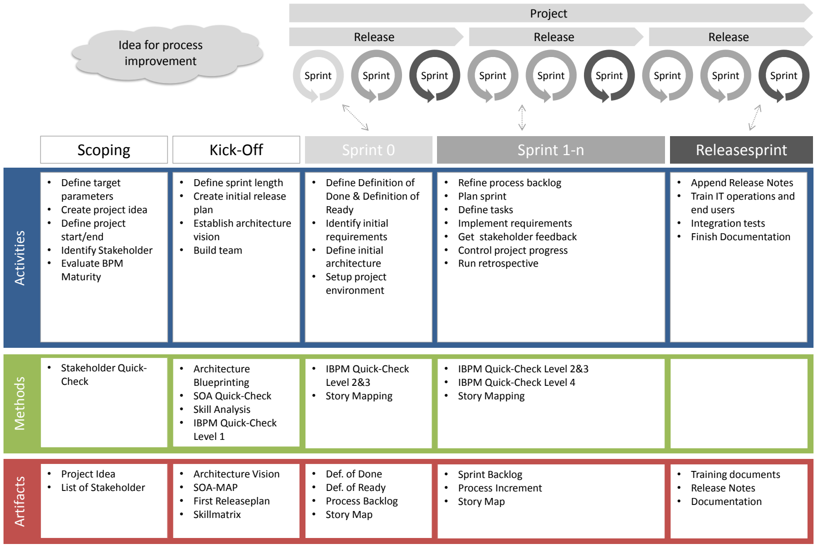
One of the major benefits of agile software development approaches is that they are run in short and frequent iterations. This enables a higher transparency for both sides. Problems become visible earlier than in classic approaches and can be addressed directly. Business professionals are often working on multiple projects at the same time. This leads to a problem regarding their availability and focus on one single project. In BPM projects, the business professionals need to be closely integrated, since the project is not only about changing software but also about organizational changes.

**Principle 5.** *Continuous attention to technical excellence and good design enhances agility.*

Besides needing a well-designed architecture, the technical implementation of a business process might differ from the business processes model. Therefore synchronization is needed. At this point, discussions often arise as to whether the process model or the (existing) technical implementation is leading. From our point of view, the process model should be the leading artifact. Nevertheless, to succeed with agile BPM projects it is necessary to focus on architecture first. An advantage of using modern BPMS is that there is a common base on which reference architectures can be used as a starting point for new projects.

**Principle 6.** *Simplicity—the art of maximizing the amount of work not done—is essential.*

When business people and IT people work together, it is important to keep it simple. The focus should always be on a minimum valuable process. That means that the details are only needed for the requirements that will be implemented in the next 2–3 sprints. By doing so, the approach stays conform to the agile principles that value generating business value over documentation.



| | Scoping | Kick-Off | Sprint 0 | Sprint 1-n | Releasesprint |
|---|---|---|---|---|---|
| Activities | • Define target parameters<br>• Create project idea<br>• Define project start/end<br>• Identify Stakeholder<br>• Evaluate BPM Maturity | • Define sprint length<br>• Create initial release plan<br>• Establish architecture vision<br>• Build team | • Define Definition of Done & Definition of Ready<br>• Identify initial requirements<br>• Define initial architecture<br>• Setup project environment | • Refine process backlog<br>• Plan sprint<br>• Define tasks<br>• Implement requirements<br>• Get stakeholder feedback<br>• Control project progress<br>• Run retrospective | • Append Release Notes<br>• Train IT operations and end users<br>• Integration tests<br>• Finish Documentation |
| Methods | • Stakeholder Quick-Check | • Architecture Blueprinting<br>• SOA Quick-Check<br>• Skill Analysis<br>• IBPM Quick-Check Level 1 | • IBPM Quick-Check Level 2&3<br>• Story Mapping | • IBPM Quick-Check Level 2&3<br>• IBPM Quick-Check Level 4<br>• Story Mapping | |
| Artifacts | • Project Idea<br>• List of Stakeholder | • Architecture Vision<br>• SOA-MAP<br>• First Releaseplan<br>• Skillmatrix | • Def. of Done<br>• Def. of Ready<br>• Process Backlog<br>• Story Map | • Sprint Backlog<br>• Process Increment<br>• Story Map | • Training documents<br>• Release Notes<br>• Documentation |

**Fig. 6.** Agile BPM Framework Overview

All discussed principles have their impact on a BPM project. In most cases, it is not possible to stay conform to all of them.

**Best Practice 2 (Parameters for Agile or Classic Project Approaches).** Evaluate the parameters shown in figure 5 to decide whether you should tend towards executing your project in an agile or more classic (waterfall)-oriented approach.

## 4 An Agile BPM Methodology

Since IBPM and Scrum are suitable methods for their purpose, the missing link is a value-adding combination of both frameworks. The goal of the current section is to combine BPM and Scrum to introduce a flexible framework for agile BPM projects. The proposed framework consists of three core aspects: ”Project approach”, ”Artifacts”, and ”Methods”. It enhances the reflected adaption of agile principles in BPM projects. Our focus is on the technical implementation of process-centric projects that are realized based on BPMS.

Agile BPM projects can be divided into the phases and types of sprints shown in figure 6. As discussed, we differentiate between artifacts, methods, and activities in each phase. Before a project starts, it must be scoped. Afterwards, the project kick-off starts the project and is followed by an adjustable number of sprints, until the project goal is achieved. Depending on the project, the number of releases will vary.

At customer sites, the agile life cycle often leads to a discussion regarding modeling vs. implementation. This is one of the central differences between tra-

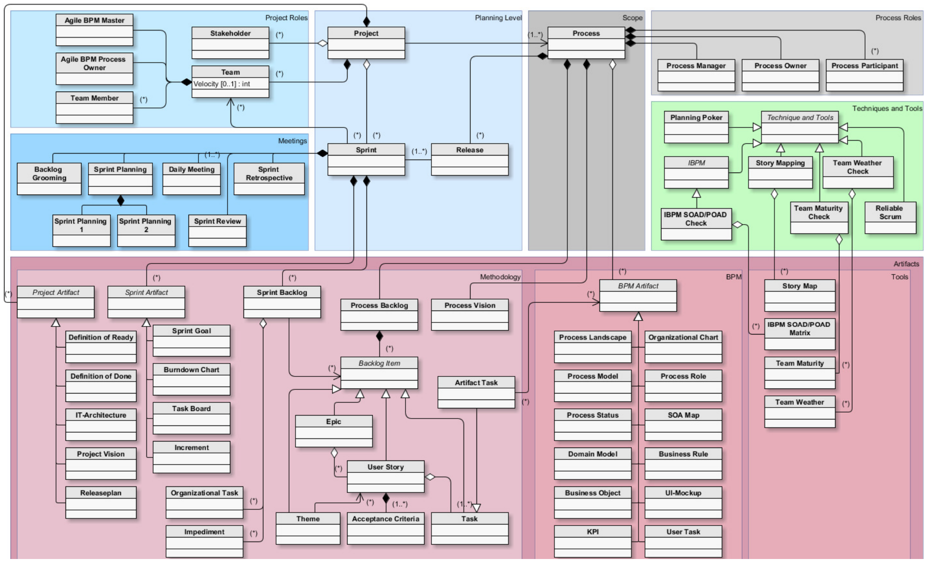


**Fig. 7.** Agile BPM Meta Model

ditional agile software development projects and BPM projects. So, how does the documentation and modeling of business processes fit into the idea of agile software development? Though it is obvious that the code itself is a kind of documentation, how can we argue the value of business process models? Since a BPM project is mostly running on different abstraction levels (from management, business professionals and IT), we need to clarify a language that is understood by both worlds. A process model helps to communicate the needs of the business people to the IT. Furthermore, a modern BPMS allows the implementation of the processes based on their models.

**Best Practice 3 (Clarify Modeling vs. Implementation).** We believe that in a BPM project, both—the process model and the implementation—generate business value. Furthermore, BPM projects are often connected with organizational change that is based and communicated using the different models.

### 4.1 Agile BPM Meta Model

Given that a process model is adding value to the project and needs to be specified before the process can be implemented, we generate additional dependencies that have to be managed. A common problem in a software development project using Scrum and User Stories is that the big picture or strategic fit is often lost. The combination of IBPM and Scrum in an agile BPM project helps to keep this big picture. Therefore several of methods and artifacts are used in our approach.

In the following paragraphs, we will give an overview and some examples to highlight the linkage between both approaches.

The meta model of our proposed approach, shown in figure 7, depicts the different artifacts, methods and activities, and indicates how they are connected with the processes. Not all relations are shown at the given level of granularity. As can be seen, the model is divided into different main subjects, according to figure 6. From the BPM perspective, we assume that a process has a manager, an owner and multiple participants. Processes exist in different releases. A process improvement will be initiated with a new project, where a project has different roles. It always has Stakeholders and depending on the project size, a project might consist of one or multiple teams working together. Each team, aligned to the idea of Scrum, has its Agile BPM Master, an Agile BPM Process Owner and a 3–5 member team. We chose a different wording to highlight that those roles need additional competencies.

The Agile BPM Process Owner is responsible for identifying the customer's needs. He has to extract and transfer them into appropriate requirements in the process backlog. The prioritization and estimation of the backlog is also his responsibility. Besides that, he is accountable for the Return on Invest of the project. It is necessary that he is always available for the team, in case questions and problems arise during a sprint. The Agile BPM Master takes care of the compliance to the Agile BPM Methodology, the project approach and different responsibilities. He is familiar to BPM as well as agile principles and approaches. Particularly in the early phases of the project he is the trainer and mentor for the whole project team and moderates the meetings. He is the first contact person in case of any ambiguity and for problems that may threaten the target achievements. The team has the same role it has in Scrum. Since it is a BPM project, the required skills are different to those in software development projects. Projects have multiple sprints and these sprints are associated to releases of the appropriate processes. Every sprint is run the same way. At this point our approach is very similar to Scrum. We decided to establish the Backlog Grooming as a mandatory activity within our projects, since we discovered out that most of the projects use something similar to help the product owner to prepare the backlog for the next sprints.

### 4.2 Tools and Techniques

We identified a set of tools and techniques that support us in delivering successful BPM projects. In the following we describe an excerpt of these tools and techniques as best practices.

Given that we work with user stories in our process backlog, we developed three IBPM Quick Check Levels based on the IBPM Matrix (see figure 1) for different purposes and phases (see figure 6). They establish the big picture and help to identify the BPM artifacts needed to achieve the defined targets and goals. The goal of those different Quick Checks is to ask the right questions depending on the detail level and project approach, e.g.

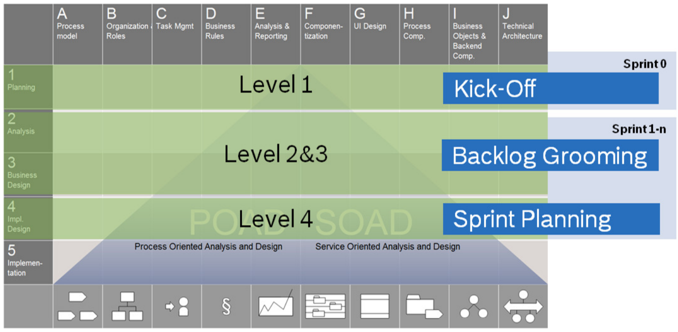


**Fig. 8.** IBPM Quick Check Level

– ”Do we need a UI? If so, how should it look like?”
– ”Are there any business rule candidates within the process?”
– ”Which non-functional requirements have to be considered?”

**Best Practice 4 (Use the IBPM Story Check).** You should use the IBPM Story Check (see figure 9) in the early phase of a new user story to identify which subjects and dependencies the specific story will generate. The goal is to classify new requirements and to identify the relevant IBPM pillars. Even though it is a very simple enhancement to a story card, it helped in our projects to handle dependencies and to maintain the big picture. Lets say we have a story card that says: *”As a purchasing agent I want to create a new purchase requisition to initiate the purchase process”*. This story will get a priority and some acceptance criterias. In our framework, we additionally map this story into a specific process step. In this case it is *”Create purchase requisition”*. This process step can be found directly in our process model. Doing this for each user story will help us to keep track of our end-to-end processes. A best practice is to offer three options for the relevance of a specific pillar: Existing artifacts, Artifacts needed, Not needed. Based on this information we can derive specific tasks and artifacts that have to be delivered before we can put the story into a specific sprint. We highly recommend to tie this method into the Definition Of Ready.

**Best Practice 5 (Use IBPM Quick Checks).** Use the different IBPM Quick Checks levels shown in figure 10 to decide on important steps within a Sprint. Since the IBPM Quick Check Level 1 is only used within the project kickoff, the other three are used within the preparation of the next sprints. The Backlog Grooming is supported by the Level 2 and 3. The goal is to make the requirements achievable for the team. Therefore it is necessary to identify the missing BPM artifacts. If there are any required artifacts missing, they have to be addressed within the next sprint. Since IBPM specifies a lot of artifacts, it is not mandatory to deliver all of them. In agile BPM projects, the associated user story is used as guideline, which artifacts should be considered. The team defines which of the artifacts generate value for the business and IT alignment.

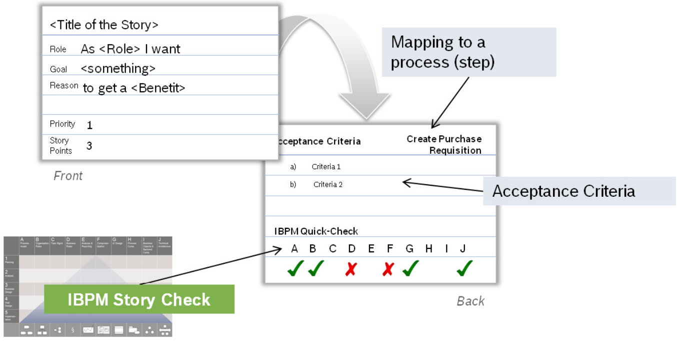


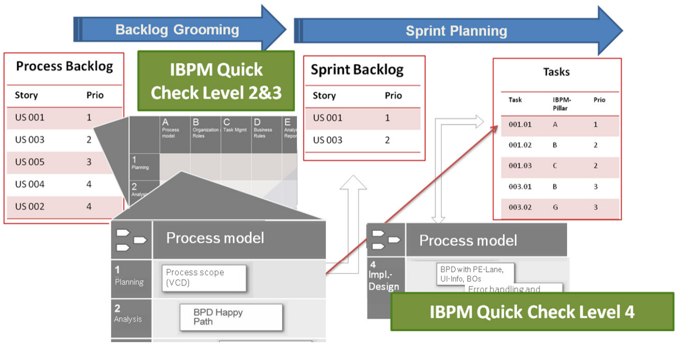
**Fig. 9.** IBPM Story Check

All of these checks generate a broad set of artifacts. The artifacts are linked to either a process, the project or methods. To handle the complexity and to understand their relations, we use the meta model shown in the beginning of this section.

**Best Practice 6 (Customize the Framework).** Take the time to think about a project-specific customization carefully. Depending on specific problems or questions there are ways to solve issues by adapting some of our best practices. One point that sometimes leads to questions is the feasibility of requirements within one sprint. To make sure that any requirement that gets into a sprint can be achieved successfully, the "Definition of Ready" defines what needs to be delivered prior to the realization. It is a best practice to establish the Backlog Grooming to identify the information demand for the next one or two sprints. In case the team and the Agile BPM Process Owner identifies a missing BPM artifact, there is enough time to gather or create the required information.

**Best Practice 7 (Use interleaved Teams for Business and Technical Perspectives if needed).** Use interleaved parallel teams to generate "ready" backlog items with an offset of one sprint if the stories become too complex. One team represents the business perspective and shapes the requirements which will then be implemented by the second team. This approach collides with the agile principles, but sometimes helps to get at least some of the concepts into a project. [14]

**Best Practice 8 (Adjust stories from horizontal to vertical as the project progresses).** From the product development perspective, user stories have to be vertical (e.g. one feature/activity in depth). Otherwise they can't produce any benefit. In contrast, BPM projects are not about product development. We recognized that there are often horizontal user stories which are predominantly at the beginning of a project (e.g. a click-dummy of the process). They help people understand how their process will look and feel like. Afterwards it is much easier to identify which aspects must be considered vertically.



Process Backlog

| Story | Prio |
|---|---|
| US 001 | 1 |
| US 003 | 2 |
| US 005 | 3 |
| US 004 | 4 |
| US 002 | 4 |

Sprint Backlog

| Story | Prio |
|---|---|
| US 001 | 1 |
| US 003 | 2 |

Tasks

| Task | IBPM-Pillar | Prio |
|---|---|---|
| 001.01 | A | 1 |
| 001.02 | B | 2 |
| 001.03 | C | 2 |
| 003.01 | B | 3 |
| 003.02 | G | 3 |

**Fig. 10.** Sprint preparation with IBPM Quick Checks

Adjust the splitting of the user stories from horizontal to vertical as the project progresses.

## 5 Experience

In our professional experiences, we have analyzed multiple projects and their lessons learned. Most of the less successful projects suffer from the same problems. Waterfall oriented approaches were not able to use the benefits of modern BPMS. The time between the initial planning, the analysis, design phase and the final implementation causes a reduced end-user acceptance, a lack of management attention and often changed business requirements.

The first practical experiences of applying the presented framework are very promising. We had several projects that were successfully accomplished based on the described framework. We observed that the classification of the project environment is one of the key issues for successful BPM projects (see best practice 2). In projects where agile approaches are used for the wrong reasons, for example to only seem state-of-the-art, we recognized that they were neither successful, nor did they emphasize the agile principles. Talking to project members clarified that it only left "‘scorched earth"’. One of our goals is to establish guidelines that help BPM projects to decide between agile and classic approaches. In the following we introduce one of our recent projects that highlights how the sketched framework can be adapted for specific requirements.

### 5.1 Service Portal Project

In September 2012, our company set up a project to build a remote service portal based on our core products. Before the project started, we considered the parameters that would influence the project. The outcome of our classification (based on figure 5) stated that it matched our expectations of an agile environment. The project was defined by a fixed timeline and the allocated resources.

Since the project was based on our core products, the technical architecture was also well defined in the early stage of this project. Another important aspect was fuzzy requirements at the project start which often resembled more a vision than concrete issues. The basic parameters of the project can be summarized as follows. We had one single team which was cross-functional and international but had no former experience with agile methodologies. In this project we began by coaching the team and filled the position of the Agile BPM Master. Additionally we supported the Agile BPM Product Owner by getting the stories ready according to our definition. Since the team had neither experience in IBPM nor in Scrum, we chose to start with a reduced set of artifacts to begin with. Besides a lack of experience in the Agile BPM area, we had to face cultural differences. The people who formed the team were from Singapore, USA and Germany. Nevertheless, we passed through the defined phases (see figure 6), which helped the team to structure their work several times, especially in the initial workshops and sprints.

The team chose to start with a sprint length of one week. They started with this high frequency for several reasons. First of all, they wanted to learn more about the Agile BPM approach and expected a faster adaption. On the other hand they had fuzzy requirements and wanted to remain flexible in the face of a highly changing process backlog. At the end of the project, their decision has proven of value. During the project the stakeholders were highly satisfied by the results and the project approach. Before the project started there was a certain suspiciousness among the stakeholders. They were doubtful how a newly formed and international team should produce any usable outcome in a short running project like this. The results from our project have shown that most of the issues can be overcome, based on the application of the agile principles explained earlier.

### 5.2 Lessons learned

We have learned that a clearly defined process for collaboration accelerates the knowledge transfer and the efficiency. From our experience we know that it works well to start with agile projects in smaller scopes. The most important thing to start with agile approaches is to find a project initiative that matches the requirements for an agile approach. Do not try and implement the agile approach in a project just to say that you are doing so; in most cases this will not work out well.

We additionally found out that IT departments often think in short terms. Unfortunately, in a lot of cases, the business departments are not ready to think this way yet. Their organization does not support early and frequent feedback since they are working in many different projects and can not devote their full attention to one project. At this point our approach can not deliver a general recommendation, since every company is differently organized. We can only demand that the stakeholders from the business departments must have at least half of their time available for a particular project, since otherwise it will not be possible to align the pulse of business and IT. Also keep in mind, that one of the most important things is to keep it simple. It is not about following our framework in a dogmatic way but rather about finding a pragmatic, customized path. When we start working with our approach, we consider the people and their knowledge and then successively bring in methods, tools, and artifacts.

## 6 Related Work

BPM methodologies can be clustered based on their focus (e.g. enterprise, process, or project level) or purpose (e.g. business process redesign or continuous process improvements). Enterprise-BPM [12] has a focus on the enterprise-wide establishment of BPM. It helps to identify project initiatives, to manage complex application landscapes and to improve BPM based on a risk-oriented approach. Six-Sigma' DMAIC, the RummlerBrache Methodology [11], or Jeston and Nelis [9] are focussing on the process level. In addition to those methodologies , there are a number of other approaches. Nevertheless, most focus on the process level and help to identify possible process improvements. Existing BPM methodologies that focus on the project level, such as the Integrated BPM Methodology (IBPM) [12], help business departments to precisely state their requirements as process models and corresponding artifacts. The traditional BPM-lifecycle (model, implement, execute, analyze) [1] defines how to hand over the documented requirements to the IT department for implementation.

Evaluating the agile methods and approaches, we focussed on the most popular methods, e.g. Scrum and Kanban [13][6][2]. Scrum is originally a framework for developing complex software products. We chose Scrum in this approach, because it fits very well with the IBPM approach.

During the last year, we recognized a stronger interest in this topic in the academic sector and the industry as well. Parallel to the thesis on which this paper is based on, there had been some investigations into the subject. One of them is a study confirming that agile approaches are progressively adapted to process centric projects[6]. Wauch and Meyer illustrate an approach to establish an agile BPM organization within the company in their conference paper[14]. Another topic is the current rise of cloud services and their combination with BPM and agile methods [8][7].

## 7 Conclusions

This paper introduced an agile BPM project methodology that has been based upon an existing BPM methodology and an agile software development framework. In contrast to existing BPM methodologies, the proposed agile methodology focuses on the customer needs first, by tightly integrating them into the process implementation.

As a result, the first two steps of the traditional BPM lifecycle (e.g. modeling and implementation) are merged together, resulting in "better" processes, since the business departments iterate in close cycles together with the IT departments to implement the processes that are really needed. Since the business gets a better understanding of what they really want in each iteration, the fuzziness of the to-be processes is cleared up in early stages. Due to the high frequency of feedback cycles and the daily synchronization of the progress, we discovered that the agile approaches establish a very transparent project environment.

While the first practical results are very encouraging, in our ongoing research we will analyze more agile BPM projects and investigate the results to come up with recommendations on how to handle those difficulties. If needed, the proposed framework will be adapted to a more fine-grained support of different project causalities.

## References

1. van Aalst, W.d., Ter Hofstede, A., Weske, M.: Business process management: International conference, BPM 2003, Eindhoven, the Netherlands, June 26-27, 2003 : proceedings. Springer, Berlin and and New York (2003)
2. Anderson, D.J.: Kanban: Evolutionäres Change Management für IT-Organisationen. dpunkt, Heidelberg and Neckar, 1 edn. (2011)
3. Beck, K., Jeffries, R., Highsmith, J., Grenning, J., Martin, R.C., Schwaber, K., Cunningham, W., Sutherland, J., Mellor, S., Thomas, D.: Manifesto for agile software development (2001), http://agilemanifesto.org/
4. Denning, S.: The Leader's Guide to Radical Management. John Wiley and Sons, San Francisco, CA (2010)
5. Kent Beck et al.: Twelve Principles of Agile Software (2001), http://agilemanifesto.org/principles.html
6. Komus, A.: Studie: Status Quo Agile: Verbreitung und Nutzen agiler Methoden (2012)
7. Kruba, S., Baynes, S., Hyer, R.: Bpm, agile, and virtualization combine to create effective solutions. CoRR abs/1208.3887 (2012)
8. Krumeich, J., Werth, D., Loos, P.: Knowledge management and business processes learning on the job; a conceptual approach and its prototypical implementation. In: Malzahn, D. (ed.) eKNOW 2013, The Fifth International Conference on Information, Process, and Knowledge Management. Nizza, France. pp. 1–7. International Academy, Research, and Industry Association (IARIA), ThinkMind (2013)
9. Nelis, J. (ed.): Business Process Management: Practical Guidelines to Successful Implementations. Taylor & Francis (2008)
10. Pichler, R.: Agile product management with Scrum: Creating products that customers love. Addison-Wesley, Upper Saddle River and N.J (2010)
11. Rummler, G.A., Brache, A.P.: Improving Performance: How To Manage the White Space on the Organization Chart. The Jossey-Bass Management Series. ERIC (1995)
12. Slama, D., Nelius, R., Breitkreuz, D.: Enterprise BPM: Erfolgsrezepte für unternehmensweites Prozessmanagement. dpunkt-Verlag, Heidelberg, 1 edn. (2011)
13. VersionOne: State of agile survey 2011: The state of agile development: 6th annual (2011)
14. Wauch, F., Meyer, S.: Agilität als Wegbereiter für lebende Prozesse. In: Engstler, M., Oestereich, B. (eds.) IT-Projektmanagement 2012+ im Spagat zwischen Industrialisierung und Agilität? pp. 49–62. dpunkt-Verlag, Heidelberg (2012)
15. Woodward, E., Surdek, S., Ganis, M.: A practical guide to distributed Scrum. IBM Press, Upper Saddle River and NJ (2010)