Introducing S-BPM at an IT Service Providers

4

Marc Sprogies and Werner Schmidt

Abstract

IT Service consumers have a clear idea of agile, flexible and transparent service processes to quickly get their needs satisfied. For an IT service provider like WK EDV GmbH this arises the challenge of designing its procedures adequately. For that reason WK decided to consolidate and optimize their service processes. It ran a pilot project to analyze, redesign and newly implement the software deployment process which is part of their overall Application Lifecycle Management (ALM) process. The project team applied Subject-oriented Business Process Management (S-BPM) as methodology and the Metasonic Suite as the respective software toolset in order to gain insights into and experience with the S-BPM environment. This contribution reports on the course of the project, the results and the learnings.

4.1 Project Background and Initial Situation

WK EDV GmbH (short: WK) is a well-established medium-sized IT Service Provider based in the Ingolstadt area. With about 130 employees it offers managed IT services, consulting, software development and client engineering to a variety of international customers in many different industries, from automotive to retail.

M. Sprogies (✉)
WK EDV GmbH, Ingolstadt, Germany
e-mail: marc.sprogies@t-online.de

W. Schmidt
Technische Hochschule Ingolstadt Business School, Ingolstadt, Germany
e-mail: Werner.schmidt@thi.de

A. Fleischmann et al. (eds.), *S-BPM in the Wild*,
DOI 10.1007/978-3-319-17542-3_4

With a flat and agile structure, the organization is strictly oriented towards the needs and projects of its customers, offering flexible and scalable services. Each organizational unit is responsible for managing the services it provides and the related processes. As service consumers often specify their own requirements and influence the service process, WK's process landscape contains many variants and alternatives. Managing them turns out to be a major challenge for the organization, which was striving for increasing transparency and better control of all managed services and processes. Consequently the managing directors of the company aim for implementing a Business Process Management (BPM) environment which supports standardization with unique definitions of process cores and roles while keeping the flexibility to manage customer-specific process variations. This environment to develop should also include software support by a Business Process Management System (BPMS).

As a first step, management started a project for harmonizing and optimizing service processes of the business unit 'Managed IT Services', which are ordered in WK's overall Application Lifecycle Management framework (ALM). As Fig. 4.1 shows, the processes span the entire application lifecycle, embodying the typical plan/build/run scheme.

The first sub-project described here focused on the software deployment process within ALM (see Fig. 4.1). This process is one of the core competencies of the Managed IT Services branch of WK EDV GmbH. It serves to deliver software (applications) onto its customers' client computers on different operating system platforms. The process is pretty complex and instances can follow many different patterns depending on what customers specify in their order. In regular vendor evaluations customers stated their overall satisfaction, but also articulated potential for further improvement, because services and their delivery sometimes deviated contentwise and temporally from what was negotiated. The service owners on the provider side not only became aware of these facts as addressees of the questionnaires but also by their own perception. As a matter of fact they could not really monitor and control the process because of missing check and measuring points. Deployments resembled individual projects rather than instances of a standardized procedure. The service owners identified the following major reasons:

- The work procedure was roughly specified in a flow diagram, but process participants did not sufficiently follow this specification. The reason was missing IT support to force following the defined steps including communication both within WK and towards the customer.
- As a consequence it was not guaranteed that all necessary steps are performed, which negatively affected the quality of the process output. The resulting

Fig. 4.1 Application lifecycle management (ALM) at WK EDV GmbH

instances also have been intransparent and heterogeneous and could not be systematically monitored with process performance indicators (PPIs), making proper management of the process difficult. As an example, cycle time of a deployment could significantly exceed because of one pending step, a fact which might not have been recognized for quite a while.

In order to realize improvements the objective was to analyze, model and prototypically implement the process as an IT-based workflow applying S-BPM methodology and technology. Workflow execution in the resulting environment should allow achieving to-be values of PPIs, like reaching more than 85 % of all client computers in a deployment or more 98 % successful deployments on reached clients (see also Tables 4.2 and 4.5). The decision for S-BPM was not based on a comprehensive evaluation of methodology and tools, but on its assumed suitability for communication-intensive processes like the one in focus. Besides resolving the mentioned weaknesses the sub-project should improve process documentation, transparency and acceptance. It should also allow all participants to learn about how the subject-oriented approach could support the way to establishing and sustaining the pursued BPM concept in the organization. The lessons learned were intended to help get valuable experience for succeeding steps in organizational development.

4.2 Course of the Project

4.2.1 Retrospective Overview

Due to high workload of all employees no regular staff member could take responsibility for the pilot project. Therefore WK management assigned it to a student, one of the authors, as a task for his bachelor's thesis (Sprogies 2014). Limited time of staff to contribute to the project by giving information input was the major constraint, paired with little explicit knowledge of BPM methodology and BPMS. The student at least had some basic skills gained in a university class, but no experience in Subject-oriented BPM. This was the 'playground' on which he started and drove the project as project leader (PL).

Figure 4.2 shows the course of the project with S-BPM lifecycle activities, results, involved S-BPM roles and software tools being assigned to different phases, which are then presented in detail.

4.2.2 Preparation Phase

After having been assigned the project order, the PL started some preparation steps in order to set the stage for action. The activities included

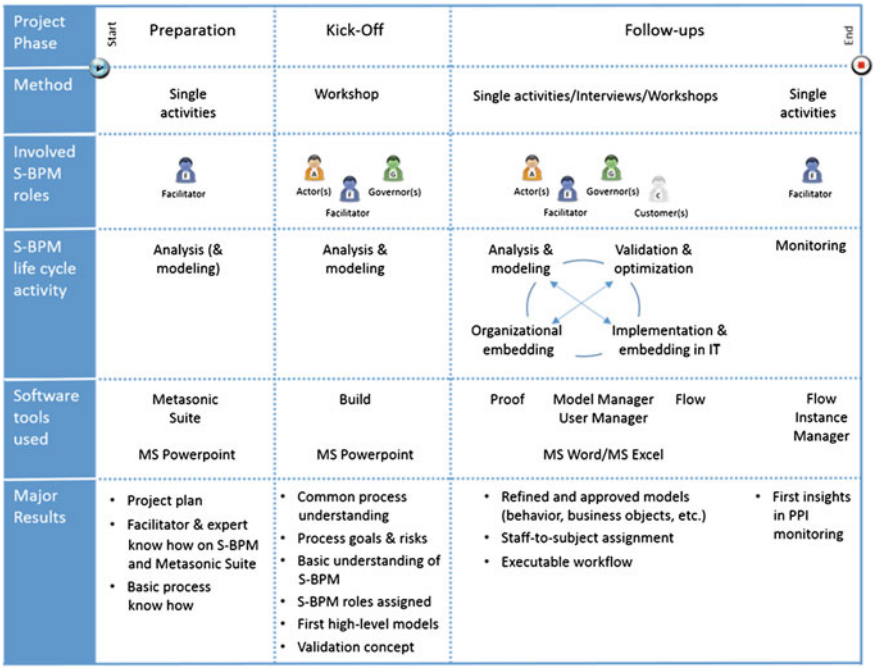

Fig. 4.2 Project overview

- Roughly planning the course of action, including the phases depicted in Fig. 4.2
- Familiarizing himself with S-BPM using the textbook by Fleischmann et al. (2012)
- Installing and familiarizing himself with the Metasonic Suite using the user manuals and the case study book *S-BPM Illustrated* (Fleischmann et al. 2013)
- Learning about the process by studying the existing flowchart (three pages) and making his own observations

Based on the knowledge he had gained the PL was ready to organize the kick-off meeting.

4.2.3 Initial Workshop

4.2.3.1 Workshop Preparation

In preparing for the kick-off workshop, the PL first defined the objectives and the time frame. The half-day meeting would serve to develop a common understanding of the process and to define the overall project frame. The PL identified and invited the participants (see Table 4.1). As input he prepared a presentation and handout for introducing S-BPM (overall approach, notation, S-BPM lifecycle, etc.) to the audience. A Word document was structured like jBook forms for subject-oriented

Table 4.1 S-BPM roles taken by WK employees (numbers in brackets)

S-BPM role (No.)	WK role taker
Governors (3)	Managing director Business unit manager 'Managed IT Services' Team manager 'Client Services'
Actors (3)	WK roles (5) in software deployment process (actors usually take more than one role): • Deployment requestor • Client management engineer • Quality verifier • Deployment coordinator • Deployment agent
Facilitators (2)	Team manager 'Client Services' Student (PL)
Experts (1)	Student (PL)

analysis in order to store online the workshop results as well as the outcome of the follow-up activities. In addition, the Metasonic Build was prepared for documenting results on the fly, in particular for creating process models.

4.2.3.2 Workshop Meeting and Results

From the S-BPM lifecycle perspective the meeting included analysis and modeling activities. The seven workshop participants spent approximately 2 h on the introduction of the S-BPM approach and on developing a common understanding of the process. They discussed for roughly another 2 h how to set the overall project frame. A fifth hour was used for separating sub-processes, identifying subjects and agreeing on future steps.

Many of the results reported below did not have to be developed from scratch. They were in parts formulated in advance by the PL based on his prior analysis and only needed to be discussed, elaborated and agreed upon in the meeting. This way the following results were achieved and mostly documented in the Word file and/or in model diagrams, with additional specifications in the Metasonic Build.
Methodology-related results:

- All participants had a basic understanding of S-BPM and the S-BPM lifecycle
- S-BPM roles had been assigned to WK representatives (see Table 4.1)
- Middle-out analysis and modeling by construction were considered to be the appropriate ways of (further) analysis and modeling
- Validation concept (detailed in Sect. 4.2.4.2)

Process-related results:

- Process goals (see Table 4.2)
- Process risks (see Table 4.3)

Table 4.2 Goals and metrics of software deployment process

Major goals	Metrics
Improved output quality through standardized process	>85 % of all client computers (deployment targets) are reached
Improved output quality through enforcing performance of all steps, particularly in quality assurance	>98 % successful deployments (on reached target systems)
Increased transparency	Stakeholders can access instance status information at any time and in real time
Reduced cycle time	<2 weeks (for standard deployments)
Minor goals	Metrics
Automated and detailed documentation of instances (logging)	Availability of detailed event logs
Improve response time in problem handling	Meet defined time constraints for (emergency) changes

Table 4.3 Process risks and counter measures

Risk level	Risk description	Counter measure
High	Faulty deployment scopes may endanger client function → roll-outs are critical and set dependent service user projects at risk	Documentation of scope development
		Early and intensive communication in case of scope deviation
		Early alerts and communication in case of error situations
Medium	Service user misinterpret modeled process	Early and intensive service user participation in process design
Low	Poor process performance caused by service user	Careful monitoring and quick action at execution time

- IT support of process tasks/activities
  The participants identified two IT systems supporting the process activities. LanDesk Client Management is a client engineering and software deployment system with functions for creating images or administrating the client landscape. MS Word, MS Excel and MS Powerpoint were selected to be used for activity check lists, protocols and reporting.
- Process network
  For a top-down view on the software deployment process and its positioning in the overall process landscape the participants created process network diagrams (PND). First they derived a PND from the ALM process chain in Fig. 4.1, by adding calls between processes in the form of input and output relations (see Fig. 4.3, upper part). Then they split the software deployment process into related sub-processes as shown in the lower part of Fig. 4.3.

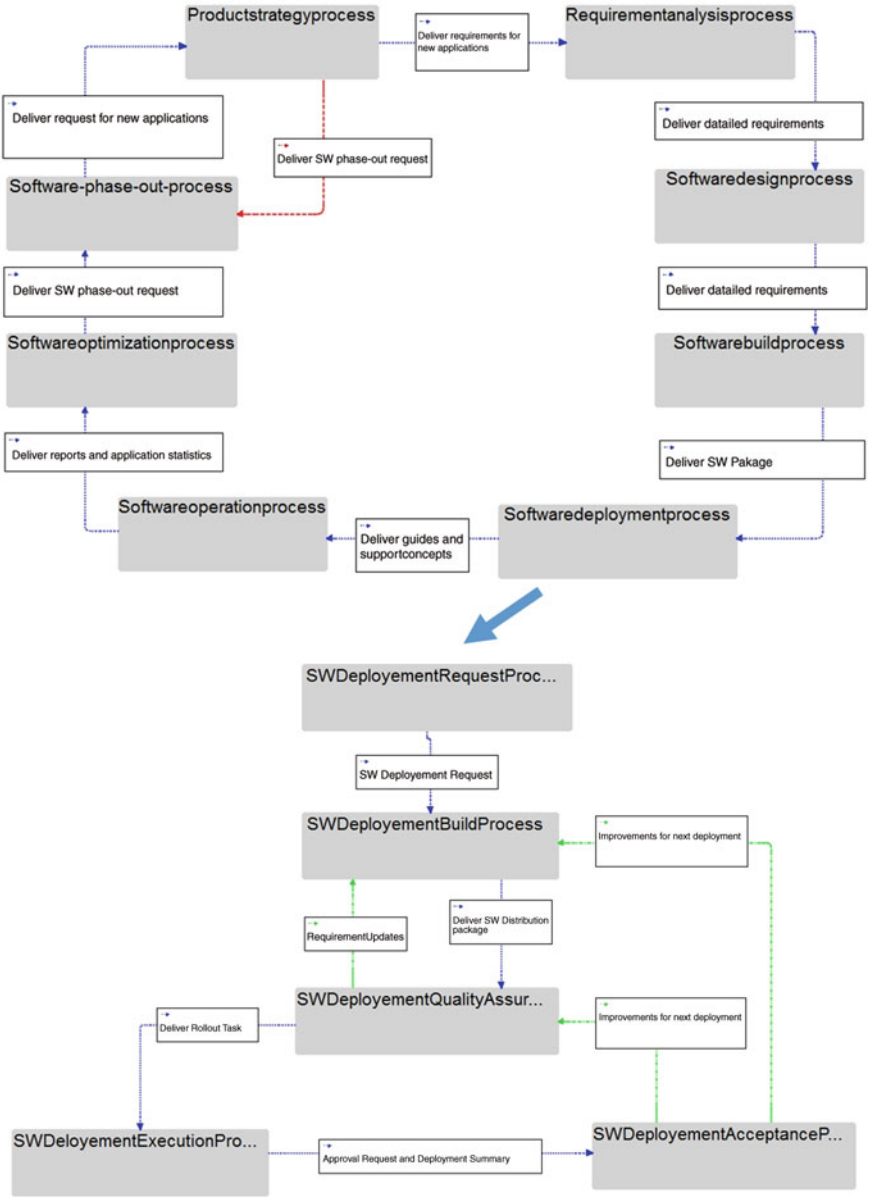

Fig. 4.3 Process network diagrams 'Application Lifecycle Management' and 'Software Deployment Process'

Software Deployment starts with the 'Request' sub-process, where a requestor defines the requirements for a deployment. 'Build' includes the creation and configuration of distribution packages and deployment tasks as objects in the

Table 4.4 Subject identification

Subject	Major activities in the process	Sub-process involvement (see Fig. 4.3)
Deployment requestor	Orders software deployment	Request
Quality verifier	Assures software deployment quality (checks scope definition etc.)	Request, Build, Quality assurance, Execution, Acceptance
Client management engineer	Creates scopes and installation packages	Build, Quality assurance, Execution
Deployment agent	Deploys software	Quality assurance, Execution, Acceptance
Deployment coordinator	Checks dependencies between deployments, checks reports and assures communication	Quality assurance, Execution, Acceptance

client management system (CMS). Testing these objects takes place in the 'Quality Assurance' sub-process, while in 'Execution' the CMS-based deployment, monitoring and reporting are accomplished. The sub-process 'Acceptance' organizes structured acceptance of the deployment and collects suggestions for improvement.

- Subject identification
  Based on the swim lanes of the existing flow diagram identifying the subjects only took minutes. They are listed in Table 4.4.

4.2.4 Follow-ups

After the initial workshop the PL, namely in his role as facilitator and expert (S-BPM method), planned and iteratively performed subsequent activities according to the S-BPM lifecycle to push the project on. This meant involving the stakeholders in interviews and workshops in order to refine, complete, implement and validate the process design. Such joint work usually was complemented by individual preparation work and a later elaboration by the PL.

The more or less sequential order of the following description does not exactly reflect the actual chronological sequence. As is typical for the open S-BPM lifecycle, the course of action was characterized by sometimes simultaneously and iteratively performed activities.

4.2.4.1 Analysis and Modeling

At first the PL interviewed the identified representatives (actors) of each subject about their work procedures in the sub-processes. The information gathered was first documented in the Word file as in Table 4.4, but more detailed, and per sub-

Fig. 4.4 Subject interaction diagram for sub-process 'Quality assurance'

process. It was used to clearly separate steps between subjects and to identify interaction points.

The latter formed the basis for modeling the communication structure in the Metasonic Build, for each sub-process revealing the message exchange between subjects and their linking via interface subjects. In small workshops the PL discussed and validated (see Sect. 4.2.4.2) each of the five resulting subject interaction diagrams (SID) with the concerned actors and governor (here: team manager 'client services'), ending up with 37 message types. Figure 4.4 depicts the SID for the 'Quality assurance' sub-process as an example.

Together with the existing flowchart the verbal description of subject activities also served as input for modeling the subject behavior in the Metasonic Build by the PL. In order to refine the models he also observed and participated in the processing of real software deployment instances, taking the roles of the different subjects (apprenticing). The drafted subject behavior diagrams (SBD) were used for discussions with the actors in order to correct and complete the behavior specification (see Sect. 4.2.4.2). Figure 4.5 depicts a part of the SBD for the 'Client Management Engineer' in the 'Quality assurance' sub-process.

While observing real instances the PL could also identify business objects transferred with the exchanged messages (e.g., forms, documents, checklists),

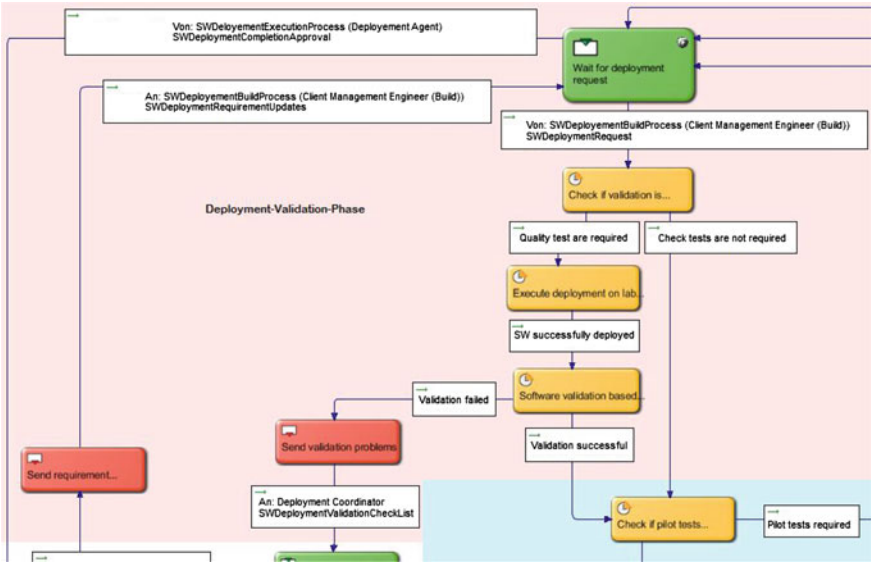

Fig. 4.5 Subject behavior diagram for 'Client Management Engineer' in sub-process 'Quality assurance' (clipped)

BO name	SWDeployment Request		
Data element	Type	Mandatory	Remark
Deployment Requestor	String	Yes	
Title	String	Yes	Rollout Title
OS	Enumeration	Yes	Options: • Windows • MAC OSX
Due Date rollout	Date	No	
Pilot phase needed	Boolean	No	

Fig. 4.6 Data structure of the business object 'Software deployment request' (clipped)

including their data structures. Again, the information obtained was documented both in the Word file (see Fig. 4.6 for a data structure) and in the Metasonic Build and then evaluated in workshops with the actors. Modeling in Build also included the specification of layouts for and views on business objects (see Fig. 4.7), later at runtime controlling the access (e.g., read, write) to data elements of business objects in any behavior state and the presentation on the screen. In order to create the business objects, views and layouts available at runtime they were assigned to the subject behavior states where necessary.

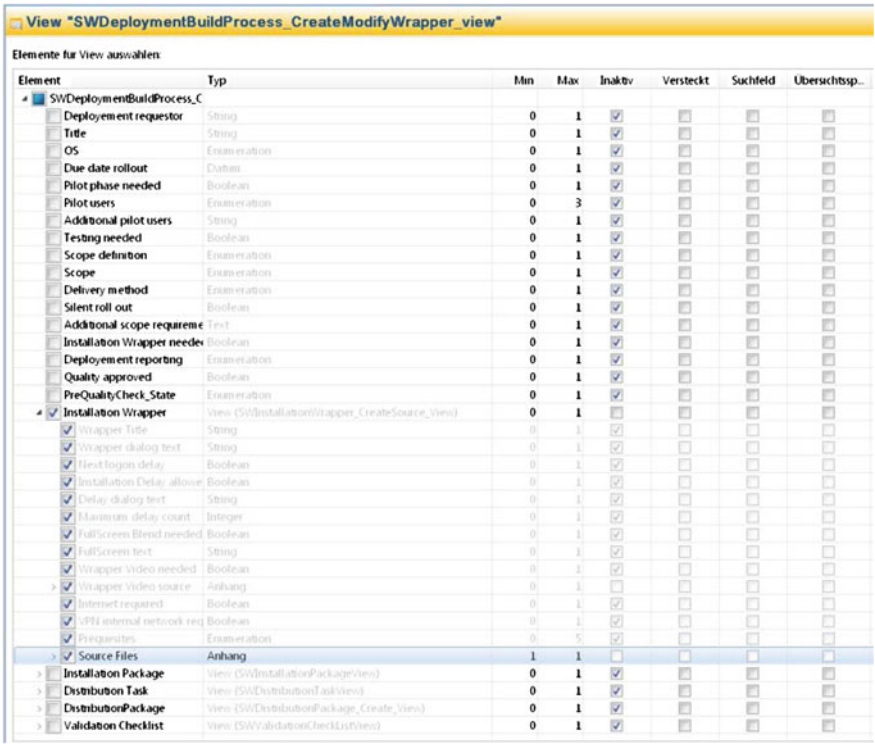

Fig. 4.7 Definition of the 'CreateModifyWrapper' view on the business object 'Software deployment request' (clipped)

4.2.4.2 Validation and Optimization

In line with the major goals in Table 4.2 the focus for validating and optimizing was on increasing the effectiveness in terms of output quality. Improving single steps came second. This primarily led to the modeling of the complete and consistent as-is procedure and to making sure that all activities are being performed. With respect to efficiency the cycle time was of interest. It will be addressed in Sect. 4.2.4.5.

The WK governors considered integrative validation and optimization already accompanying analysis and modeling to be very beneficial. For that reason they developed a respective concept in the initial workshop, jointly with the other participants. It envisages stepwise, bottom-up validation and approval of process artifacts, mainly models, on different levels, respectively involving the responsible governors besides actors and facilitators (see Fig. 4.8). In the course of the project the application of the concept was supported by the Proof and the Flow component of the Metasonic Suite with the created models having been uploaded before.

The first level refers to the subject behavior. Here the Web interface of the Metasonic Proof was used on a single computer to validate the (business) logic of the subject behavior without data and without concrete people being assigned to the

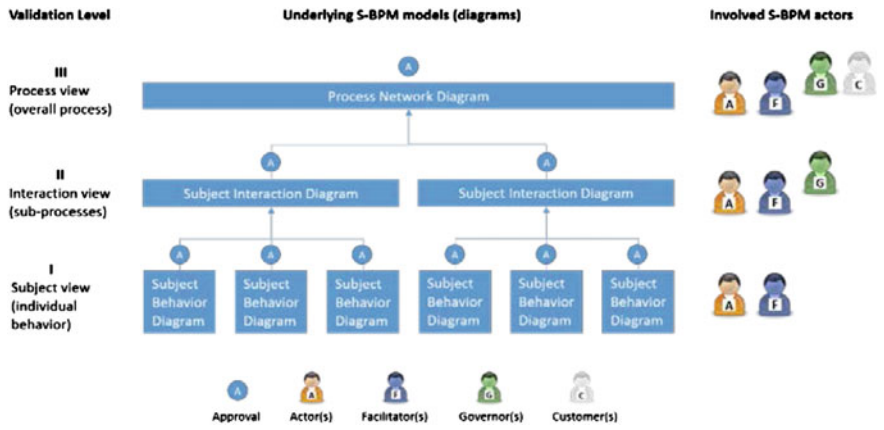

Fig. 4.8 Validation concept

subjects. Supported by the facilitator real actors of the software deployment process as subject representatives could quickly check completeness and the order of steps without the overhead of putting in concrete information. In this way they easily detected faults or missing actions. Corrections and suggested improvements were integrated on the fly and validated again.

After the behavior logic was found to be appropriate, the Metasonic Flow came into play in order to validate the behavior, including business objects and views and layouts. As the Flow component is a workflow engine for running process instances in real-world operations, the facilitator needed to assign people as concrete users to the subjects in the models before (see Sect. 4.2.4.3). After that the actors could log on as individual users to the system.

The facilitator then guided them through the workflow application, which not only controlled the interaction and single behavior steps of all users, but also presented and managed electronic forms based on the specified business objects, views and layouts. This way the users could test the behavior of 'their' subject, this time putting in valid but fictitious data and thus getting the feeling of the real workflow application. Now they could additionally recognize deficiencies with respect to business objects in the process designed so far, like missing or unnecessary data elements or inappropriate settings for change permission and the display of data. Again, modifications could be made and tested on-the-fly until an actor formally approved the correctness of his or her part of the workflow.

On approval of all subject behaviors of a sub-process, the communication as it was modeled in the subject interaction diagram was validated. On this second level the actors and the facilitator again initially used the Metasonic Proof in a server setting with distributed computers in order to iteratively test and improve the interaction in the sub-process from their point of view. The actors could 'play' their subject at their individual workplaces and report on the need for changes to the facilitator by e-mail. After they were satisfied also the team manager 'Client

Services' was involved as governor to identify potential deviations from the process interaction as he expected it. Therefore, the facilitator used the 'Recorder' function of the Proof software to meticulously show him the course of communication during processing instances (see Fig. 4.9).

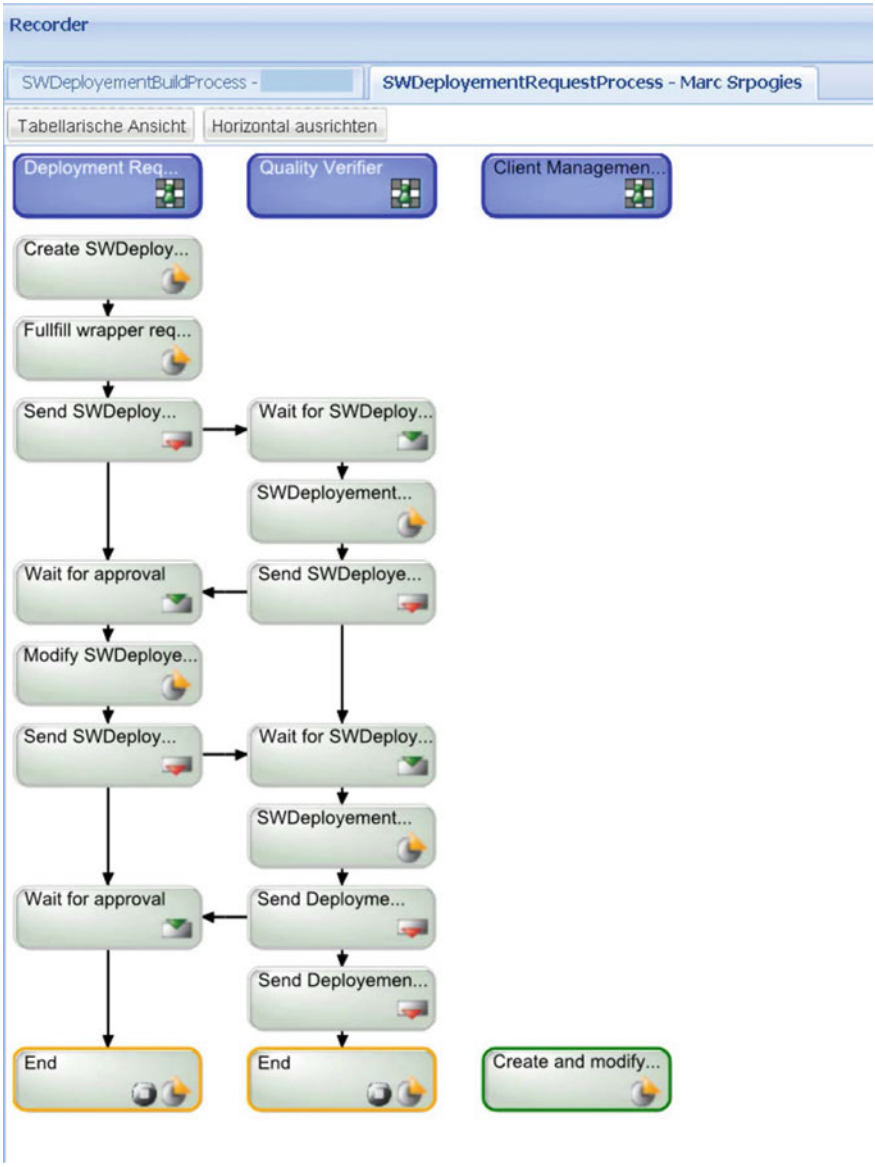

Fig. 4.9 Recorder log of sub-process 'Software deployment request'

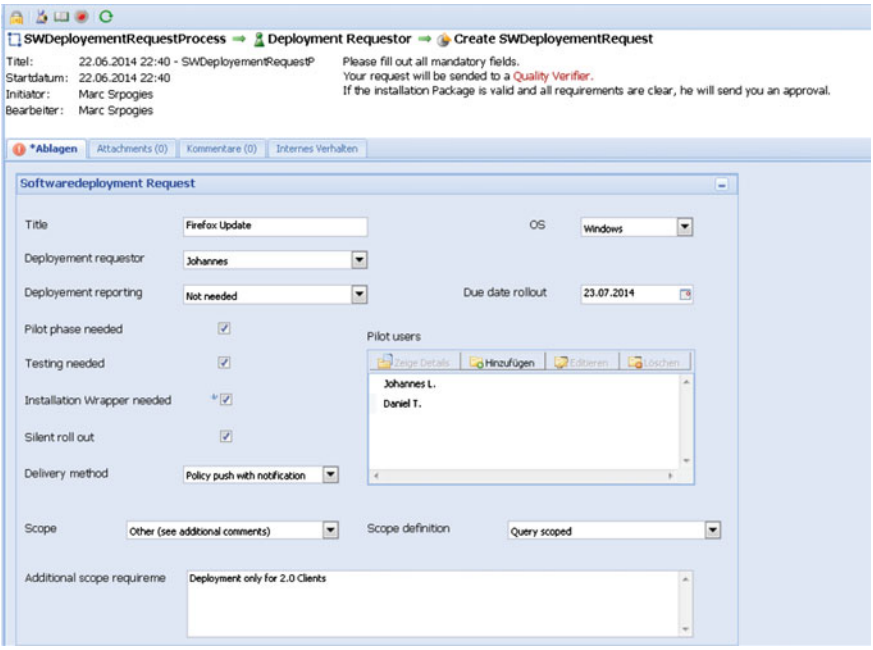

Fig. 4.10 Initiating a 'Software deployment request'

A second criterion for giving his approval to the sub-process was the quality of the deliverables (e.g., an installation package). To check it, actors and facilitator provided the governor with the respective output. After all sub-processes had been validated and approved as described, the overall software deployment process was tested on the third validation level. The previous steps had led to a process design without logic or content-related weaknesses. So at this stage not only the business unit manager 'Managed IT Services' and the managing director as governors, but also a real customer as a service consumer were involved. The customer got access to the Metasonic Flow and placed a variety of realistic orders in the system (see Fig. 4.10), which were then processed supported by the workflow application built so far without any programming. This gave the service consumer the chance to check whether the process matched his expectations with respect to the course of communication with the service provider and the desired output. The latter was not only evaluated by isolated inspecting of the quality of the software package, the roll-out and the accompanying documentation. In parallel all deployments processed were completed using the new workflow following the previous as-is procedure, so that the results could be compared.

As all parties were satisfied with new process and its results the overall process was approved.

4.2.4.3 Organizational Embedding

In S-BPM organizational embedding means relating process models to the actual organization, in particular assigning concrete human actors to the abstract subjects, allowing the workflow engine at runtime to involve the people as specified by the organizational design. For the software deployment process this only took minutes applying the User Manager component of the Metasonic Suite. The relevant information was already obtained during analysis and modeling. The facilitator just needed to represent it by defining the few users and, via groups and roles, finally assigning them to the subjects modeled in the SIDs in Build.

4.2.4.4 Implementation and Embedding in IT

Thanks to the nature of the S-BPM notation and the Metasonic Suite, bringing the designed process to execution as an IT-based workflow only required uploading the models to the Model Manager component. It became available for interpretation by Flow to process real-world instances. IT implementation of electronic forms included in the workflow at runtime had already been accomplished by modeling them in Build as business objects with data structures, views, layouts and specifications for their use in behavior states.

The PL also used the Metasonic Build functionality to define so-called dynamic process rules in order to automate state transitions at runtime based on business object content. This helped automatically route to the right process branch without user intervention. Such a rule, e.g., was used to control the behavior of the Client Management Engineer depending on the value of the data element ‘silent roll-out’ (true or false) in the business object ‘deployment request’. In the ‘True’ case the workflow engine would perform the state transition to the respective activity thread and otherwise follow the transition to the alternative path.

The integration of other IT applications was realized by so-called refinements in the respective states, e.g., invoking MS Excel and opening a spreadsheet in the behavior specification of the subject ‘Client Management Engineer’ in the subprocess ‘Software Deployment Build’. Integrating LAN Desk via refinements, where appropriate, was taken under consideration, but postponed to a follow-up project.

4.2.4.5 Monitoring

Monitoring aspects were considered in the project in a twofold manner. The first was providing real-time information about the current status of process instances at runtime. This transparency could be realized using the ‘Recorder’ function already described in the validation and optimization section (see Fig. 4.9) and also available in the Metasonic Flow.

With respect to the goal of reducing cycle time this PPI needs to be measured and controlled. For that reason the target value of two weeks (=10 days × 24 × 60 = 14,400 min) was specified in Build as a maximum on (sub) process level (see Fig. 4.11).

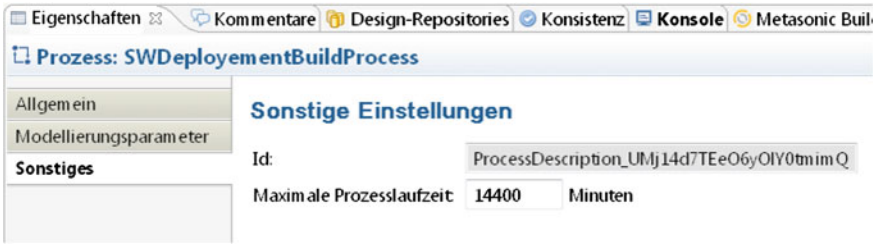

Fig. 4.11 Setting of maximum cycle time

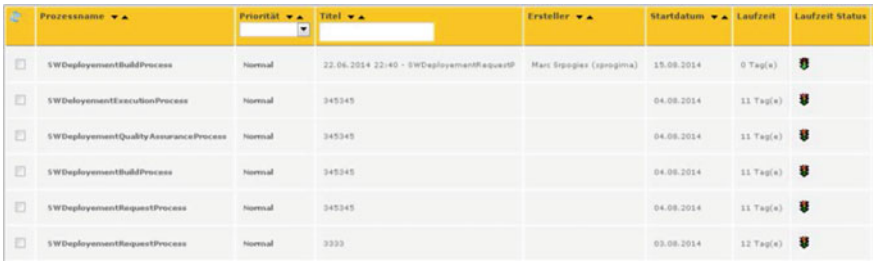

Fig. 4.12 Monitoring running process instances

At runtime the Instance manager component of the Metasonic Suite allows monitoring the running instances by displaying the elapsed processing time and traffic lights indicating the status with regard to the given maximum of cycle time (see Fig. 4.12).

As the Metasonic Flow process engine logs all sorts of events during execution (e.g., timestamps for state transitions), many valuable pieces of information are available for middle and long term analysis and reporting. Limited time in the project at hand prevented the stakeholders from getting deeper into that. Defining sense-making PPIs and further exploiting the capabilities Metasonic Suite offers for monitoring and reporting are candidates for future steps.

4.3 Results

Results of the work described in the previous sections can be distinguished in achievements and findings in the domain of software deployment, and in experiences related to S-BPM.

4.3.1 Goal Achievement in the Software Deployment Domain

Table 4.5 summarizes the achievements of the project in terms of improving the software deployment process, referring to the goals and metrics in Table 4.2.

Table 4.5 Goals, metrics and achievements

Major goals	Metrics	Achievements
Improved output quality through standardized process	>85 % of all client computers (deployment targets) are reached	Approved standardized process design implemented as IT-supported workflow automats decisions and can guarantee completeness of process steps
Improved output quality through enforcing performance of all steps, particularly in quality assurance	>98 % successful deployments (on reached target systems)	
Increased transparency	Stakeholders can access instance status information at any time and in real time	Subject-oriented process models and the 'Recorder' function allow one-stop info about status of instances being processed by distributed contributors
Reduced cycle time	<2 weeks (for standard deployments)	Cycle time is modeled as a constraint, can be monitored and thus be managed
Minor goals	Metrics	Achievements
Automated and detailed documentation of instances (logging)	Availability of detailed event logs	Given by process design and log file capabilities of the workflow engine
Improve response time in problem handling	Meet defined time limits for (emergency) changes	See above

Some more findings not directly related to the aspects in the table were:

- Intensive stakeholder discussion about process goals helped to identify process quality factors like client reachability, installation success and cycle time, which had not been completely understood before.
- During analysis, modeling and validation, stakeholders gained deep insight into how specifications (decisions) in the deployment order have impact on the steps and the course of a deployment and thus also influence cycle time. For instance, the customers can decide whether they want their package being tested only in a laboratory setting or during a pilot phase. Choosing the first option apparently leads to a different procedure and different consequences compared to the second one. The stakeholders explicitly understood that the customer thus takes a decision like "time before quality" or the other way round.
- Based on these insights the participants could clearly structure the process in several parts, in future allowing intermediate evaluation of (sub) process results (quality gates) and measurement of elapsed time in order to intervene early in case of deviations from to-be settings.

The proof of concept was given throughout the extensive validation and optimization sessions. The positive impact on the quantitative metrics still needs to be evaluated in daily operation after going live.

4.3.2 Experience with S-BPM Methodology and Software

The student started working on the project early in April and finished by the end of June 2014. He spent half of his working capacity on the project, which means the effort from his side was 40 man-days. Table 4.6 summarizes the experience gathered in the course of the project.

Table 4.6 Experience with S-BPM

What worked well? (positive aspects)	What needs to be considered? (Trade-offs, issues)
Analysis and Modeling	
Middle-out approach worked well	
Top-down structuring in process networks reduced complexity	Increasing modeling effort because of many external subjects
Independent bottom-up behavior modeling 'picks up' the individual actors and lets the process emerge	Missing end-to-end view (compared to flowchart) caused some irritation on management (governor) level
Active modeling by stakeholders increases their attention and concentration and accelerates elicitation of process information	Although the Metasonic Build user interface was perceived quite intuitive S-BPM modeling without substantial training turned out to be not as easy as expected
Direct modeling in the Metasonic Build is more efficient than using jBook forms initially	
Existing flowchart with swim lanes allowed behavior modeling in advance what significantly saved time of the actors	Flowchart was not very detailed
Apprenticing by the PL also helped preparing and refining behavior models, business objects and added to actors' time savings	
Interviews and small dedicated modeling workshops were very efficient (compared to workshops with many participants as experienced in other projects)	
Intensive stakeholder inclusion eliciting a lot of implicit process knowledge like communication patterns and information exchanged which were documented so far	

(continued)

Table 4.6 (continued)

What worked well? (positive aspects)	What needs to be considered? (Trade-offs, issues)
Stepwise validation concept	
Misunderstandings and logical errors were early and quickly identified and resolved both on individual behavior and on interaction level	Validation sessions are time-consuming and collide with daily operation. The facilitator needs to carefully coordinate them for balancing time savings through clearing faults with respect to the work capacity invested in validation
Stepwise procedure saved time of governors as they were only involved on an advanced maturity level (after approval of all actors)	
Time to (overall) approval was felt to be pretty short	Individual behavior validation can be performed with subject representatives of each subject at their workplace. The facilitator comes with a portable computer running a single instance of the Metasonic Proof. At the latest when the Metasonic Flow is used to test the process with real users and business objects a server installation is necessary to do it in a distributed environment. Otherwise the stakeholders need to leave their workplaces and meet in a single location which costs them additional time
Contentwise intensive but resourcewise moderate participation of all stakeholders until their approval fostered high acceptance of the resulting process design	
Involving a service consumer as customer can help increasing customer satisfaction	
Business objects	
Definition of business objects in general is easy	
Validation steps help quickly defining and verifying business objects	
Views and layouts allow sophisticated specification of behavior at runtime without programming	Defining high numbers of views and layouts is rather time-consuming
Organizational embedding	
Easy and quick assignment of concrete users with the Metasonic User Manager	
Changes of user data and roles do not require deployment to be effective in the runtime environment	
Implementation and embedding in IT	
Deployment of models and business objects is easy and does not require expert know how	
Refinements offer good opportunities to integrate software applications like LanDesk Client Management	For autodidacts like the student the available version of the Metasonic Suite documentation was not sufficient in the area of particular functionalities, such as refinements, process performance indicators, reporting

4.4 Conclusion and Outlook

The project results presented in the previous section indicate that the S-BPM methodology supported by suitable software tools actually can unfold many of the benefits claimed by its proponents.

The experience gained in this to a certain extent typical application setting provides valuable findings, even though the developed solution has not gone live yet. Whether and when this will happen is a matter of management decision, not only in terms of the overall future of BPM in the company, but also with regard to the underlying methodology and tool environment.

During the project a single cycle of organizational development was walked through completely, however without putting the result to operation. After going live, continuous organizational development could start, following and occasionally adjusting the presented pattern. As mentioned above, pushing forward PPI-based monitoring and seamless integration of LanDesk Client Management could be among the activities to further develop the designed business process as well as the S-BPM process.

References

Fleischmann A, Schmidt W, Stary C, Obermeier S, Börger E (2012) Subject-oriented business process management. Springer, Berlin

Fleischmann A, Raß S, Singer R (2013) S-BPM illustrated. Springer, Berlin (Open Access)

Sprogies M (2014) Prototypische Implementierung eines nach der subjektorientierten Methode entwickelten Geschäftsprozesses bei einem IT-Dienstleistungsunternehmen. Bachelor Thesis, Technische Hochschule Ingolstadt (in German)