Managing Experience in Business Process Management

First Experiments

Jürgen Dorn[1]

Abstract. Business Process Management (BPM) is a management approach to identify, document, manage and finally to optimize business processes continuously. Today, Information Technology is an enabler to improve such processes. Learning objectives in a university course on BPM should be rather practice oriented in the area of business management, information systems development and team-working, because in enterprises typically people from different departments with different competences have to work together in BPM-projects. We have developed a new master study course that itself follows the paradigm of continuous process improvement and apply case-based reasoning as a mean for improving the projects of participating students. In the course, students develop in teams of five a business process application. Four major steps are problem formation, design and modelling of a process, deployment of the process in a workflow management system and monitoring of the deployed business process. Each team can freely choose an application domain with a business process and key performance indicators, a modelling tool and a workflow engine. Competences of students, decisions during the project and issues occurring in the different phases are documented in a content management system with sophisticated rights management enabling students to learn also from experiences from other teams. Case-based reasoning is applied to share experience between different projects, by indicating similar projects, teams and issues. This approach is now taken in the third year and first experience with knowledge sharing are reported. Such an approach could be also taken by an enterprise planning several business process management projects.

Keywords: Business Process Management, Process Modelling, Workflow Management System, Case-based Reasoning, Continuous Improvement

1 Introduction

Business process management (BPM) is a field in business informatics focusing on improving performance in an organization or over cooperating organizations by managing and optimizing their business processes [Ko09]. It can therefore be seen as

[1] Technische Universität Wien, Institute of Information Systems Engineering, Favoritenstrasse 9-11, A1040 Wien, Austria, juergen.dorn@tuwien.ac.at

a "process optimization process". It is argued that BPM enables organizations to be more efficient, more effective and more flexible than organizations applying only traditional functional-oriented management approach. Business processes impact the cost and revenue generation of an organization.

As a strategic approach, BPM sees processes as important assets of an organization that must be understood, managed, and developed to offer value-added products and services to customers. This approach closely resembles other total quality management or continual improvement process methodologies and BPM proponents also claim that this approach is enabled by information technology [Ri16]. As such, many BPM articles and scholars frequently discuss BPM from one of two viewpoints: management science and/or information technology.

BPM uses various methods to discover, model, analyze, measure, improve, and optimize business processes. A business process coordinates the behavior of people, systems, information, and resources to deliver outcomes in support of a business strategy. Processes can be structured and repeatable (e.g. providing a standardized service to a customer) or unstructured and variable (e.g. development of software). BPM projects may address improvement of organizational processes, but also inter-organizational processes often with a focus on a stronger integration of organizations. BPM is key to align IT investments to business strategy, because if IT is used without restructuring business processes typically no advantages can be achieved and moreover, additional work is introduced.

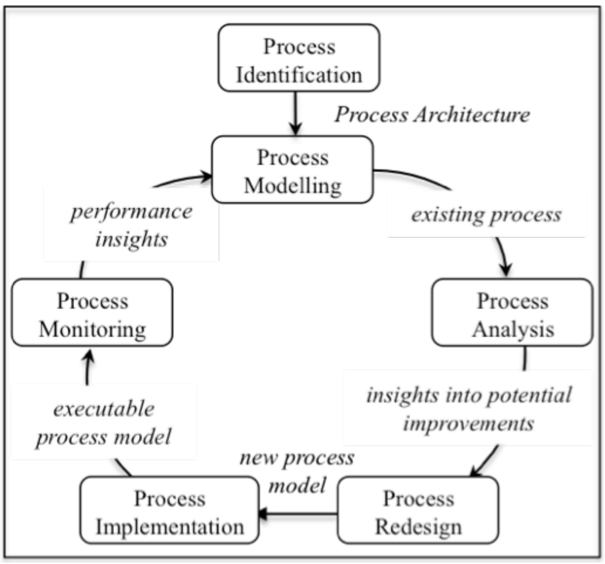

Fig. 1: BPM Lifecycle due to Dumas et al. 2013 [Du13]

BPM projects follow a lifecycle as described in Fig. 1. During process identification, the relevant business processes of a problem domain of an organization are identified. Due to Hammer and Champy [HC93] only such processes should be focused that deliver value to customers. Customers may also be internal in an organization. In process modelling (sometimes also called process discovery), we identify the most important aspects of a process and describe the process as it is performed at the moment. In process analysis, we identify the drawbacks of the current process. Simulation or other techniques are often proposed to identify such drawbacks. In process redesign, the process is improved based on insights from process analysis. In process implementation, a process is deployed and if a Workflow Management System (WfMS) is used, this means the control of flow and data handling for process instances (sometimes also called cases) is managed by the WfMS. The WfMS may also select resources such as members of staff or certain machines for a task instance by applying given rules or strategies (e.g. assigning work to human with best fitting competences). If a business process is deployed, it is advised to monitor the execution of its instances to identify bottlenecks and other issues occurring during execution to support the continuous improvement.

Over the recent years, many approaches were undertaken to facilitate the development of business process-aware information systems. The ultimate objective is to generate software and information systems automatically from designed models [Do09]. Moreover, if business analysts see a possibility to improve a process, it should be facilitated that the change is initiated in the business process model and then transformed in an automated fashion into the executed workflow [Do09].

Current state of the art is modelling of business processes with BPMN 2.0 [OM10]. A number of free as well as commercial tools exist that allow the modelling of business processes graphically. These graphical models can be exported into BPMN-compliant XML format. This export can be used to automate the creation of a WfMS for a designed process. The process management blog[2] of A. Naef references 109 different process management systems automating processes in different domains. A WfMS may automate business processes in different domains. Typically, such automated processes are strongly structured, with clearly defined decision points. Process tasks may be automated themselves, but manual tasks may be integrated also by letting users entering the status of an activity. Thus, the WfMS does support the integration, but does not necessarily automate the whole process. Projects (e.g. software development projects) are typically less structured than workflows. Nevertheless, also such projects contain tasks and dependencies between tasks that could be controlled by an information system. Here, the term adaptive case management system is often used. Thus, the large number of existing tools reflect the great diversity of business process types.

---

[2] http://prozessmanagement-blog.ch/post/47680997258/die-grosse-bpm-anbieter-liste

BPM advises to improve processes continuously. For improving a process, a process manager has to define optimization goals such as higher quality, faster or cheaper execution, greater flexibility or other criteria. For such an improvement, critical aspects of individual process invocations have to be stored. Key Performance/Process Indicators (KPIs) are a generally accepted concept for operationalization of such objectives. However, these are not defined as elements in BPMN 2.0 and other process modeling approaches. In [Fr12] an approach to extend BPMN 2.0 with KPIs is presented, but not yet implemented in available tools.

In this paper, case-based reasoning is proposed to store and to learn from past experience and to share knowledge between different teams. Each process instance (the different attributes of a team's project) is stored as a case and for decision points in such a process, case-based reasoning supports teams. Our research objective is to identify, which knowledge has to be stored and in which phases a reuse of knowledge can be supported by a case-based approach. This approach is evaluated in the course on workflow management where the course itself is modelled as a process / workflow and participating students have to make certain decisions such as selection of KPIs or selecting tools. The idea is that students participating in the course can improve their solutions by copying ideas from other actual teams and from projects from the past. To measure success in our course and to obtain a manageable KPI for our (learning) business process, we introduce user points in different categories for different tasks in the course which are managed partially by the course management system. Students also estimate their required work for each step and their result will be finally evaluated qualitatively by their peers.

In the following section, we give a short introduction into case-based reasoning and in a case-based reasoning module that we have developed with the Drupal Content Management System. In the third section, we describe the organization of our course and the course management system used to store project information and to give feedback to participants. In the fourth section, we show how cased-based reasoning is applied in course management. Finally, we conclude and propose some planned extensions for the future.

2 Case-based Reasoning

Case-based reasoning is a problem-solving approach where experience of individual problems and their solutions is stored as a case. If a new problem shall be solved, old similar cases are retrieved from a case base to reuse stored experience [Ko92]. If no similar case can be found, deep reasoning in the application domain can solve the problem or a user solves the problem without support and enters the solution as a case. Typically, cases are records of attributes, and different cases have different values for these attributes. These attributes describe the given problem (e.g. the planned project with estimated values), the problem solution (e.g. the project with the realized values), the context (e.g. the characteristics of the stakeholders) and an evaluation how successful the project was.

To retrieve similar cases, either an explicit or implicit function to measure similarity of cases has to be defined. Usually, for each attribute a similarity is defined, where the similarity is expressed by a value between 0 and 1.0. If the domain of an attribute is an Integer, two equal Integers result in a similarity of 1. If the values are equally distributed in the domain, a linear function can be used to describe the similarity. However, if we want to measure the similarity of effort estimated for a project a logarithmic function would represent similarity better. For example, if we have projects with small effort (e.g. 3 person months) and other large projects with an effort of hundreds of person years a linear function would not be appropriate. Very often qualitative estimations of attributes are used. For example, we may evaluate the competences of the project leader or the team members with terms such as beginner, elementary, intermediate, advanced and proficient. In this case, similarity must be represented explicitly. For most attribute types, the similarity function is inverse but sometimes it may be also meaningful to define the inverse function differently. Domain concepts and terms are another type of attribute. For example, we classify the applied project management process by a hierarchical representation of concepts where Scrum is modelled as a subclass of Agile Project Management. Using concepts lead to a similarity function based on ontological similarity [Le08]. If two concepts in the ontology are closer to each other in a hierarchy, the similarity is higher than for concepts that are far away.

For the similarity of two cases, the different attributes do not have necessarily the same importance. If we store the name of the project as an attribute, this attribute will not be important for measuring the similarity of cases. Thus, the weight for attributes is defined to enable a weighted similarity function for cases.

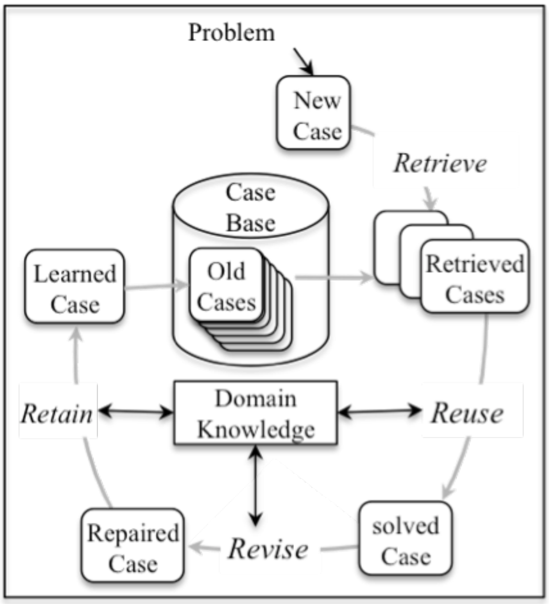

Fig. 2: CBS Cycle due to Aamodt and Plaza [AP94]

Case-based reasoning is defined as a cycle of steps supporting a continuous learning in an application. The retrieval of a number of cases (old projects) similar to an actual problem (new project) is the first step. In a second step, if the new problem shall be solved, potential adaptations of the retrieved old cases are made and a best case is selected. For example, if the estimated effort is larger than in a similar old case, we have a new case as a mixture between old and new case. Due to the larger effort, we automatically compute higher costs, if such a functional dependence is defined in the domain knowledge. This repair of a case is the third step in case-based reasoning cycle. A repair may also occur during the application of a case, i.e. in our domain during the project or in the closing phase of the project. Domain-dependent reasoning can support a repair, but also user interception is possible. The fourth step is the storage of the newly adapted and/or repaired case. This is the most important step for learning in a case-based reasoning system. In contrast to inductive learning approaches no generalization of examples is searched for, but the concrete experience of the new case is stored avoiding the so-called inductive bias. Figure 2 shows the case-based reasoning approach and is taken from the overview paper [AP94].

We have used the Content Management System Drupal 7[4] to implement our course management system. For a project on supporting knowledge sharing in project management we have developed a Case-based Reasoning module as an extension for Drupal. With the usage of Drupal, we have a basic functionality to store structured content in a database, a management of users with different privileges for user groups and to log users' activities. We use the Drupal approach of hook methods to extend standard behavior of modules. We define a *project* as a new kind of content node. The *case* module implements dedicated *case fields* defining certain hook functions which can be used to extend the functionality. For project management, we extend the functionality for similarity, adaptability, repair and learning of cases.

A project is defined as a content node with attributes realized by fields. The existing *field* module implements a certain behavior that enables the easy attachment of fields (attributes) such as integers, reals, strings or lists to a content node. The Drupal community has implemented further types of fields. For example, there exist modules implementing date fields, fields to reference a taxonomy, fields to reference another entity (a content node or a user) or to store a collection of values. We can also define an attribute that links to an external knowledge source. Since a project is a content node, several basic attributes such as a name, creation date and author are already defined. Additional generic attributes for the project structure are an extended field structure containing elements to store the weight of an attribute, a link to a similarity function and adaptation and repair methods (all implemented as hooks that can be overwritten by more specific project/case structures). As a default, all attributes get the same weight and a default similarity defined for standard fields.

The typical attributes of a project are defined in a project module. Defined attributes of a project content node are for example "project-type", "customer-type", "size" and many more. Thus, we have defined standard attributes for projects, but we assume that every organization using our system will define their own project attributes. For a new implementation, concepts and terms or our implementation could be reused, extended or replaced by a different set of concepts and terms.

Retrieving projects is supported by the "views"-module. A list of all stored cases as well as a list of cases filtered by certain attributes is supported by the existing functionality of Drupal. Additional functionality to search for similar cases was implemented for general cases based on a weighted aggregation of similarity of the fields of the cases.

A project content node is some kind of template similar to a project handbook template in which a user can enter relevant knowledge about a project. The user interface for Drupal fields allows several pre-settings to make navigation and entering of data easy. A content node has a unique identifier assigned by Drupal. Additionally,

---

[3] https://www.drupal.org

[5] https://wfm.ec.tuwien.ac.at

a user may enter a name for the node. Further the author as well as the creation time is automatically stored for a project since it is derived from the content node. We distinguish six parts of a project:

- header,
- result (product) with information about the expected outcome,
- environment with stakeholder information,
- process with information how the project is executed,
- people with information about project leader and staff and
- used technology.

The header contains information about project type, start time and budget. In each of these parts, there may exist planned as well as realized values. The task structure and assigned personnel to tasks and certain documents as risk analysis are part of the process description. We assume that each human resource assigned to a task is also a user of the system with certain credentials. For each user, competencies may be defined.

All members of a project as well as stakeholders should be able to enter relevant knowledge into a case. Often staff is busy with development and then staff does not document project progress and occurring problems. Also, the reflection after finishing a project is often not realized due to different reasons. Our proposal is to make participation transparent in this knowledge-sharing process. We propose to assign points for different kinds of participation. Depending on these types and amount of points, different kind of digital badges may be assigned to users. A user can become an expert in describing problems and their solutions or an advanced project planner because he has entered several risk analyses into the system. Assigning points to users is supported by the community module “userpoints” and for digital badges again a community module exists.

The system is described in more detail in [Do16]. Also in our course, we will store experience about projects, however, they are more specialized and some attributes relevant for general projects are not necessary here (e.g. all projects have the same size and type).

3 The Course

The course “Workflow Modeling and Business Process Management” is a mandatory course in the master study Business Informatics and a course under the module Information Systems in the master study Software Engineering of our university. The course was redesigned three years ago and one of the design criteria was to have more freedom in deciding on potential applications and tools. Many participants already work for different companies and have thus different interests and experience in process management. Thus, in the lecture no introduction into certain tools is given and only principal concepts are taught. The course is supported by a course management system called WFM, a Drupal-based Content Management System to enable knowledge sharing[1]. We expect students to have sufficient competences in Java programming. A requirement is also process modelling which is taught in the Business Informatics bachelor study, however, process modelling is partially repeated because not all participants have these required competences.

60 (75) students have enrolled in the first two cycles in the course. Before building teams, the participants have answered two questionnaires related to the topics of the course and made a self-evaluation in four competences: English language, Java programming, process modelling and business management. We used the classification defined in the Common Reference Levels of Language Proficiency for English [CE11] as well as for programming which was developed as an online test by Raphael Poss[1]. Thus, for each competence, a level beginner (A1), elementary (A2), intermediate (B1), upper intermediate (B2), advanced (C1) or proficiency (2) is assigned. Based on the results 50 (65) students remained and these were divided into ten (13) teams á five students. About 80% of students have finished the bachelor study Business Informatics, resulting in sufficient competences. However, typically students from the Software Engineering bachelor study are more competent in software development than students from business informatics.

Every team has to select a certain application domain with a core business process and has to model, to implement and to evaluate this process. For evaluation, three key performance indicators (KPIs) have to be defined. For example, one team has selected the delivery of bought products by a drone and one of the KPIs is cycle time of the workflow which is the time from ordering a transport to finished delivery of the product. A second KPI are the costs computed by aggregating costs for each activity (i.e. Activity-Based Costing [KC97]) where the costs of the drone flight is a product of required time for the flight with the costs of a drone per time unit. A third KPI is the customer satisfaction evaluated after delivery.

The concept of these applications was presented orally and evaluated by the supervisor. For each project, a list of aspects to focus further on were presented to the teams as feedback in WFM. The teams then selected a tool for modelling their processes and modelled the process. Eight different tools were selected. The modelled process was uploaded as graphic file in pdf format and in BPMN format. A first round

[5] https://wfm.ec.tuwien.ac.at

[6] http://science.raphael.poss.name/programming-levels/test

of feedback on the process model was published through the WFM. The following shows the selectable issues from the system for the assessor.

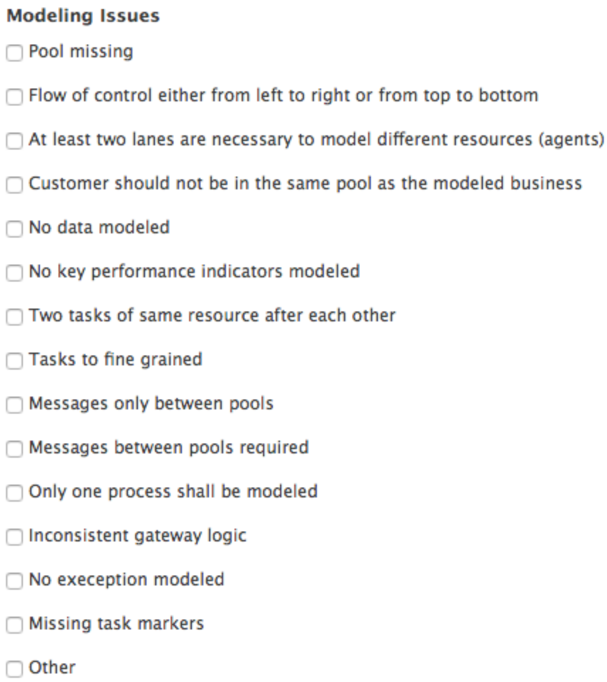

Fig. 3: Modeling Issues

One constraint on the process model is to have at least five tasks. These tasks must be performed by different resources (humans or information systems), because we want to stress the integration aspect. After completion of the process model, each member of a team implements a task written in a programming language that was also selectable. Most tasks were implemented in Java or some scripting language. Often these tasks were simulated activities. For example, the team that designed the drone delivery, has not really controlled a drone, but has taken the coordinates of actual stand point of the drone, the location of the shop from where the product has to be fetched and the location of the customer and calculated the time for flying to the different locations. One of the tasks is a user interface supporting the communication with the process customer which is accessible in the Internet.

The next step is the integration of tasks into the workflow with different techniques and protocols, the deployment of the whole system on a public, private or university server. Since workflow management is about integration, a requirement for the projects is, that each task has to be deployed on a different (virtual) server. Then, the workflow application is tested by the team. Afterwards, an evaluation by all other teams/participants resulted in 45 (65) test users for every workflow. Based on these users, the KPIs were measured in each project and a final presentation was the place for discussing success as well as pitfalls and drawbacks. Lessons learned are stored as a final aspect in WFM.

During the different phases of the projects individual members as well as whole teams can obtain points for achievements either by automated reasoning in the system or by manual evaluation through the supervisor. Three categories of points can be obtained: knowledge, project and participation points. Knowledge points are achieved through questionnaires and submissions related to theoretical knowledge. Project points are achieved for tasks prescribed in the course and participation points are achieved for additional participation in the WFM, for example, in the forum or other discussions and also for accessing documents. The quality of the solutions is evaluated by peers and supervisor. The grade for the whole course was based on the last presentation and the achieved points.

4 Case-based Management of BPM Projects

We use the Drupal module described in section 2 and create a content node "Workflow Project" with dedicated fields that are enabled for case-based reasoning. Thus, for every student project, we create an instance of this content node and store individual values for each instance. An attribute is typically a decision made in a project (selection of a tool) or an outcome of certain tasks (an issue occurred in the description of the workflow). Furthermore, certain basic information such as competences and knowledge of team members is stored. In principle, issues are negative outcomes that should be avoided in projects and the implicit reasoning of the case-based reasoning approach should make decisions transparent that lead to negative outcomes so that teams can avoid these negative outcomes. If a team member enters certain data into such a case/workflow project, she can search for similar projects to see what other teams have decided. If that has led to negative outcomes in the other project, she may change its decision.

The values allowed in most fields are based on a domain theory consisting of hierarchies described by term taxonomies in Drupal. Thus, we have defined taxonomies for competences, application domains, key performance indicators, modelling tools, task types and workflow engines, and issues. Especially issues are defined by a complex hierarchy that can grow over new applications. The important aspect is, that if a new issue is defined by a user, it must be positioned at the right place in the hierarchy. In our application, this can only be done by a supervisor and not by a student.

Next, we describe the most relevant attributes defined for a Workflow Project. These attributes are separated into the four phases of the project: *formation*, *modelling*, *deployment* and *evaluation*. Afterwards, we show what can be recommended from old projects for new projects.

4.1 Storing experiences as cases

For the *formation* phase, we store the competences of team members, their field and semester of their study and the selected application domain. Thus, it may be a result of reasoning in this phase that for certain application types the competences of team members are not sufficient. For the application domain, we distinguish on the top level between workflow, projects and inter-organizational processes. Below the workflows are separated into public administration, simple order processes, healthcare, scientific processes and cloud-based processes. The classification partially relates to the process model developed later. For a project or an inter-organizational process different features will be in the process model. The KPIs are also stored as attributes because these are dependent on the application domain and can be again seen as feature of the BPMN model.

For the *modelling* phase, we store which tool was used. Modelling tools promise compliance to BPMN 2.0, but certain extensions have influence on the mapping from model to execution. Therefore, we store which elements and partially how many elements were used. For example, we recommend to use task markers to signify which kind of tasks was designed. If not available, this is an issue. If only script tasks are used, the process seems to be very simple. If more than five tasks are modelled, the process may be too complex. In principle, a process may be complex, but if the competences of the team are not so good, this may lead to a failure. Further, issues that were criticized by the supervisor are stored and can be investigated by other groups.

For the *deployment* phase, we store the used workflow engine which may have great impact on the success of the project. There exist special process management systems dedicated to certain application domains (e.g. Apache Taverna for scientific workflows or Jira for IT project management). It may also be used for other application domains, but then a failure will be more likely. Each task in the workflow has certain characteristics that may have also impact on the success. Thus, the assignment of team members should match their competences. The programming language and complexity are stored here. Characteristics about the user interfaces are stored, too.

Finally, for the *evaluation* phase, attributes describing the final workflow and execution parameters are stored. The used public web space, the occurring problems and the measurements for the KPIs are further attributes. A final attribute is a classification of the success into *innovative* (best evaluation), *successful*, *sufficient*, *borderline* and *failed*.

4.2 Learning

Case-based reasoning is applied iteratively for the four phases of our project. In each of the four phases (problem *formation*, process *modelling*, process *deployment* and evaluation), data is entered in a case and similar cases are then retrieved that may help to support the new project.

For the first phase, the case-based reasoning approach assumes that a team is formed and it enters partial data into a new case (a template for information about their workflow project). Based on the entered data similar projects can be found. This search requires that we have defined similarity for the different attributes and have set the weights for the attributes. At the moment, all case sensitive attributes have the same weight. Thus, if only two attributes are set for a new project, both attributes are considered half for the calculation of the similarity.

Based on old cases, the system shows which KPIs were used for similar applications and whether the measurement was successful in the old projects. Thus, the new project team may decide on their KPIs based on old experience.

Based on the application domain we have certain expectations of the process model as described before. Thus, if a team enters a model that deviates considerable from such expectations, the system should make this transparent. Existing modelling tools have some impact on the ease of mapping to the execution. Therefore, depending on the software engineering competences of a team and the selected application domain, certain tools should be preferable.

In the second phase, a team designs a process and by reasoning about similar designs, existing experience is retrieved.

In the third phase, the completion of the BPMN model with execution parameters, the selected workflow engine, the used programming language and attributes of the hosting servers have a great impact on the successful implementation. Again, the competences of the team are a parameter to decide, whether certain project types may be realized successful.

In the fourth phase, the manner how test users are enabled in a deployed workflow depends heavenly on the used platform and workflow engine. The engine is also responsible for the monitoring. Some engines can monitor some KPIs more easily than other types of KPIs.

In general, the possibilities what can be learned is open, because if we identify new attributes that relate to the success of projects, these will be defined for our cases.

5 Conclusions and Outlook

Process improvement was the core focus in our course on Workflow Modelling and Process Management. This improvement was addressed on three layers: the workflows developed by student teams have addressed improvement by defining KPIs and measuring these in the evaluation phase for 45 (60) users. The next step would be to search for improvements that lead to better values for the KPIs. This was, however, not part of the project. On a second layer, we try to improve the development process of Workflow Management Systems by knowledge sharing and the Case-based Reasoning approach. Here, first improvements were possible, but the full advantage can only be achieved over several years. Especially, the further development of supporting tools and methods must be incorporated so that we can react on changes in the state of the art. On the third layer, we investigate our course as business process that should be improved year for year. The following objectives (KPIs) are addressed:

- increasing relevant learning outcomes,
- increasing efficient learning for participants,
- decreasing effort in the management of the course.

For the first objective, motivation for more knowledge exchange and usage of new software and technologies is addressed. We already honor during evaluation innovative approaches by giving extra points, but this should be made more transparent. For the second objective, better knowledge sharing and usage of old experience has to be extended. By measuring time and effort, we have also identified bottlenecks in the project work that have resulted in stronger deadlines for the second year's projects. The second objective can also be supported by recognizing further dependencies (e.g. evaluating further competences of teams and team members). The facilitation of the course management can be supported by different means such as:

- automated extraction of features from BPMN file (e.g. number of tasks or pools)
- automated recognition of issues in models and applications,
- simulation of process models,
- improvement of definition of privileges of user and teams,
- classification of applications with learning of attributes,
- automated assignment of points and
- automated extraction of new concepts for our domain model.

An important step in project management generally, is a lessons-learned phase at the end of the project. We have this also implemented within our last meeting with the teams. However, here it is difficult to distinguish issues related to the project and issues related to the course management. Both shall be documented whereby the first have to be stored in cases and the second in our course redesign for the next course.

The presented course is taught this term the third time in the described way. Certainly, further cycles are required to gain the full advantages of the course design. A further improvement could be also achieved if other teachers would follow this approach to gain more experience with such a course.

References

[Ko09] Ko, Ryan K. L. "A Computer Scientist's Introductory Guide to Business Process Management (BPM)" ACM Crossroads Summer 2009/ Vol. 15, No. 4.

[Ri16] Richardson, C. "Future Look: BPM Programs Shift Gears To Accelerate Digital Transformation Vision: The Business Process Management Playbook", Forrester Research, 2016.

[Du13] Dumas, M., La Rosa, M., Mendling, J. and Reijers, H. A. "Fundamentals of Business Process Management" Springer, 2013

[HC93] Hammer, M. and Champy, J. A. "Reengineering the Corporation: A Manifesto for Business Revolution" Harper Business Books, NY, 1993.

[Do09] Dorn, J., Grün, Ch., Werthner, H. and Zapletal, M. "From Business to Software: A B2B Survey", Information Systems and e-Business Management, 7, pp. 123-142, 2009.

[OM10] Object Management Group "Business Process Model and Notation (BPMN) Version 2.0", http://www.omg.org/spec/BPMN/2.0, 2010.

[Fr12] Friedenstab, J.P. Janiesch, Ch. Matzner, M. and Müller, O. "Extending BPMN for Business Activity Monitoring" Proc. of the 45th Hawaii International Conference on System Sciences, pp. 4158-4167, 2012.

[Ko92] Kolodner, J. "An Introduction to Case-based Reasoning" Artificial Intelligence Review 6, pp. 3–34, 1992.

[Le08] Lee, W., Shah, N., Sundlas, K. and Musen, M. "Comparison of Ontology-based Semantic-Similarity Measures" AMIA Annual Symposium Proceedings, pp. 384–388, 2008.

[AP94] Aamodt, A. and Plaza, E. "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" AI Communications, Vol 7, Nr. 1., 1994.

[Do16] Dorn, J. "Sharing Project Experience through Case-based Reasoning" in Proceedings of Int. Conference on Knowledge Management, 2016.

[CE11] Council of Europe "Common European Framework of Reference for Languages: Learning, Teaching, Assessment", http://www.coe.int/ t/dg4/ linguistic/Cadre1_en.asp, 2011.

[KC97] Kaplan, R.S. and Cooper, R., “Cost and Effect: Using Integrated Cost Systems to Drive Profitability and Performance”, Harvard Business School Press, 1997