Empowering End-Users to Collaboratively Structure Processes for Knowledge Work

Matheus Hauder[1(✉)], Rick Kazman[2], and Florian Matthes[1]

[1] Software Engineering for Business Information Systems, Technical University Munich, Munich, Germany
{matheus.hauder,matthes}@tum.de

[2] Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA
kazman@sei.cmu.edu

Abstract. Knowledge work is becoming the predominant type of work in many countries and is involved in the most important processes in organizations. Despite its increasing importance business information systems still lack appropriate support for knowledge-intensive processes, since existing workflow management solutions are too rigid and provide no means to deal with unpredictable situations. Future business information systems that attempt to improve this support need to solve the problem of facilitating non-expert users to structure their processes. The recently published Case Management Model and Notation (CMMN) might overwhelm non-expert users. Our research hypotheses is that end-users can be empowered to structure processes for knowledge work. In an evaluation with two student teams working on a software development project our solution improved the information structure, reproducibility, progress visualization, work allocation and guidance without losing the flexibility compared to a leading agile project management tool.

Keywords: Knowledge-intensive processes · CMMN · Adaptive processes · Adaptive case management · Business information systems

1 Introduction

Globalization, digital transformation as well as an ever-increasing use of information technology leads to an automation and replacement of workplaces from less skilled workers in organizations [1]. At the same time the ability to develop innovations and constantly adapt to changing market requirements is crucial for the sustainable success of an organization. As a result today's work environments require highly trained experts that can perform many complex tasks autonomously. These experts are referred to as "knowledge workers" and their processes have a tremendous impact on the success and add the most value in organizations. Davenport describes this development as follows: *"I've come to the conclusion that the most important processes for organizations today involve knowledge work. In the past, these haven't really been the focus of most organizations – improving administrative and operational processes has been easier – but they must be in future"* [2].

© Springer International Publishing Switzerland 2015
W. Abramowicz (Ed.): BIS 2015, LNBIP 208, pp. 207–219, 2015.
DOI: 10.1007/978-3-319-19027-3_17

Workflow management solutions are not suitable to support these knowledge-intensive processes, since they are too rigid and provide no means to deal with unpredictable situations [3]. Due to a large amount of exceptions resulting from the unpredictable nature of knowledge-intensive processes, traditional workflow management models would be too complex to manage and maintain [4]. Support for knowledge-intensive processes requires a balance between structured processes for repetitive steps as well as unstructured processes to facilitate creative aspects that are necessary to solve complex problems [5]. Case management has been initially promoted in 2001 by Van Der Aalst and Berens as a new paradigm to support this flexibility for knowledge workers during a process [6]. In contrast to workflow management which focuses on what *should* be done, case management focuses on what *can* be done to achieve a business goal [7].

The Object Management Group (OMG) issued a request for proposal (RFP) in 2009 to create a standard modeling notation for case management[1]. The main goal of this request is the development of a complement for the Business Process Model and Notation (BPMN) that supports a data-centric approach which is based on business artifacts [6, 7, 9]. In 2014 the result for this request was published as the Case Management Model and Notation (CMMN) [8]. The main difference of CMMN compared to BPMN is that it builds on a declarative instead of an imperative process model [9]. These declarative process models specify what should be done without specifying how it should be done to facilitate the required flexibility for more dynamic processes [10]. After the initial release of BPMN several subsets of the BPMN language were proposed by Zur Muehlen et al. [11]. Similarly, Marin et al. [12] recently proposed that future research needs to identify subsets of the CMMN language that are less complex for end-users who may not have a computer science background, and who do not need to understand the complete CMMN specification.

Our goal is to facilitate non-expert users to structure knowledge-intensive processes. Recent literature frequently emphasizes this challenge as *knowledge worker empowerment*, which is indispensable since these processes cannot be completely predefined by process designers but require on the fly adaptation by end-users while they are being executed [5, 13, 14]. Organizations are increasingly using wikis as shared knowledge repositories that can be used for collaborative gathering of information. We view these wikis as promising tools for the collaborative and self-organizing structuring of knowledge-intensive processes by end-users. Although wikis can be dynamically adapted to new needs, there are limitations of existing solutions and general research challenges for this problem:

- the notations of existing process modeling languages (e.g. CMMN) are not familiar to end-users with limited computer science background
- (lightweight) declarative process constructs like constraints for logical dependencies between tasks with their linkage to data are not supported [5, 13]
- late data modeling that allows end-users to add new data to the information model during the process enactment is not supported [5]
- mechanisms for the collaborative documentation, adaptation and instantiation of work templates for recurring knowledge-intensive tasks are not supported [14].

[1] http://www.omg.org/cgi-bin/doc?bmi/09-09-23, last accessed on: 2015-01-23.

2 Related Work

Templates facilitate the reuse of content on wiki pages that can be used for the explicit organizational knowledge creation (and sharing) in an organization [16, 17]. In the Darwin Wiki a template consists of types, attributes, and tasks that are collaboratively added by end-users. These templates are used to structure and reuse knowledge-intensive processes and our goal is to make adaptations of these templates as easy as possible for end-users without computer science background. The Semantic Media-Wiki is a prominent project that enables users to structure page content on top of an existing wiki with templates [15]. This tool also includes mechanisms for searching, browsing, and aggregating content through queries. Our approach differs from the Semantic MediaWiki since it supports the creation of *tasks* that can have logical dependencies and assigned attributes to guide knowledge-intensives processes. Furthermore, our solution provides a template mechanism with *roles* that can be used to restrict access rights for attributes and tasks, which is often inevitable in an organizational context.

The Organic Data Science Wiki (ODSW) extends the Semantic MediaWiki with new user interface features to support open science processes [18]. The ODSW focuses on scientific collaborations which revolve around complex science questions that require significant coordination, enticing contributors to remain engaged for extended periods of time, and continuous growth to accommodate new contributors. All user interface features in the ODSW are based on social design principles and observed best practice patterns from successful online communities. These features include tasks that are attached to wiki pages and can be decomposed into smaller subtasks. Every user in the ODSW has his own profile page with all allocated tasks that are structured along the time dimension, e.g., currently active tasks, future tasks and completed tasks. Semantic properties of the Semantic MediaWiki can be used to describe data elements that are attached to tasks. Although there are some commonalities with the usage of social design principles and patterns for successful online communities, the solution presented in this paper provides important functionalities for the structuring of knowledge-intensive processes that are missing in the ODSW. Our approach provides more fine-grained access rights, declarative process model constructs for tasks dependencies and reusable work templates.

Voigt et al. [19] detail application oriented use cases for structured wikis and take the ICKEwiki as an example. In their work the authors mentioned that *"the presentation of (role-based) tasks within the structured wiki seems to be an important feature to adequately display processes"*. The solution proposed with the ICKEwiki contains simple procedural steps. We extend this approach by taking into account dependencies between tasks and more sophisticated roles, e.g., skip and delegate roles. In addition to these aspects, the solution presented in this paper also proposes and implements a subset of the CMMN standard. In previous work [20] we proposed the Hybrid Wiki to allow users to collaboratively structure information. This paper builds on the experiences of the Hybrid Wiki and extends it with new structuring concepts for tasks, complex information structure hierarchies, new user interface concepts, more detailed role concepts for content as well as a declarative process model integrated with a subset of CMMN to support knowledge-intensive processes.

Case management can be distinguished into Production Case Management (PCM) and Adaptive Case Management (ACM) [21]. While cases in PCM are predefined at design-time, ACM proposes an adaptation of cases at run-time through end-users [21]. The main goal of ACM is to support unpredictable processes in which knowledge workers are not controlled but responsible to perform decisions autonomously. Processes with high control flow complexity and high variation are easier to implement in ACM compared to PCM and traditional workflow management. Despite the increasing interest in ACM, as measured by the number of publications on this topic since 2012 (cf. literature review in [13]), there are only few solutions available that address the challenges presented by Hauder et al. [13]. To the best of our knowledge this is the first paper that addresses the challenges of knowledge worker empowerment, data integration as well as knowledge storage and extraction for processes in knowledge work using a wiki approach.

Schönig et al. recently investigated [22] how process models can be learned from cases in knowledge-intensive processes that are described with CMMN. In their work the authors proposed to simplify the formalization of cases by using CMMN process skeletons. These skeletons are further specified by automatically learning from process execution history. As a result process models evolve and become more and more complete with every iteration. This approach could be applied in the Darwin Wiki, since the history of wiki pages could be analyzed to improve the templates. Furthermore, the quality of the learned models could be improved through our solution since our goal is to empower end-users to create more structure that can be learned. In Marin et al. [12], the authors recently investigated the meta-model based method complexity of CMMN. According to the authors future research is needed to identify subsets of the specification that reduce the complexity for end-users.

3 The Darwin Wiki

In this section, our solution to the problem of facilitating non-expert users to structure knowledge-intensive processes is presented. Our solution is illustrated with: (1) the design rationale, (2) the structuring concepts, and (3) the implementation, including the meta-model of our solution.

3.1 Design Rationale

Our primary research goal is to make the structuring of knowledge-intensive processes easier, so that non-experts without knowledge about a dedicated process notation can contribute to the modeling process. This is indispensable since one of the most challenging requirements in case management is the empowerment of knowledge workers to facilitate a self-organizing working mode [5, 13, 14]. We achieve this goal by accepting limited modeling capabilities of end-users and use only simple structuring elements as metaphors for processes. Thereby knowledge-intensive processes emerge bottom-up and modeling experts can use these emerging structuring elements (e.g. through mechanisms proposed in [22]) to maintain reusable work templates that become more and more detailed with every iteration of the process.

3.2 Structuring Concepts

In the Darwin approach wiki pages consist of rich-text for unstructured data as well as additional concepts that add more structure. These additional concepts enable the collaborative structuring and instantiation of knowledge-intensive processes for end-users which are referred to as work plans. The main structuring concepts for wiki pages are shown in Fig. 1 and described subsequently:

Tasks. We use tasks attached to wiki pages as an organizational mechanism to coordinate work. Tasks can be decomposed to exhibit goal-oriented hierarchical structures. Tasks for a page are shown in a side window for every wiki page (cf. ⓐ in Fig. 1), which allows users to browse tasks and scroll through the wiki page at the same time. Attributes ⓒ that document results can be assigned to tasks with drag and drop. After a task is selected mandatory attributes for this task are shown on the page.

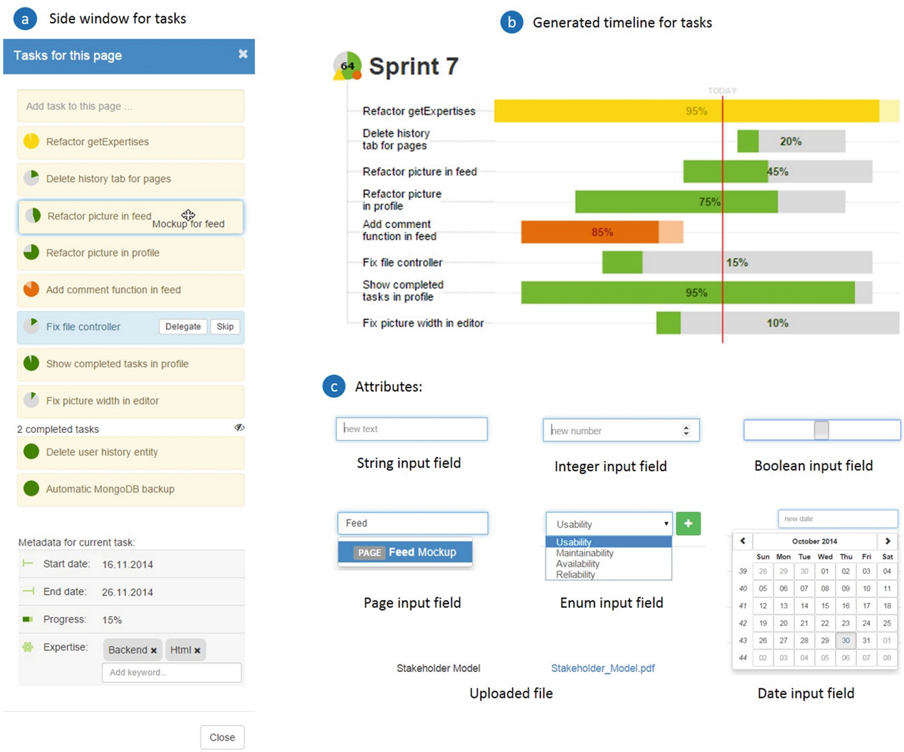

Fig. 1. Concepts for end-users in the Darwin Wiki to structure knowledge-intensive processes

Furthermore, tasks help to track the progress of activities through meta-data for start and end date, progress, and the required expertise for every task. Based on this meta-data a timeline similar to a Gantt chart is automatically generated on every page showing the current progress of all enabled tasks (and subtasks) for this page ⓑ Tasks can have attribute values assigned that reference wiki pages. These referenced wiki pages can again have tasks assigned to enable hierarchical task structures. In case a task containing subtasks is selected in the side window, the corresponding timeline with the subtasks is shown on the wiki page. The progress of tasks is either automatically computed based on the progress of the subtasks and the completion of mandatory attributes or can be manually edited by the users. Subtasks with start and end dates that exceed the dates of the parent task are marked as yellow. Overdue tasks that are not completed before the end date are marked as red. End-users can easily create new tasks on the fly at run-time in the side window by defining a name for the task and (optionally) adding attributes using the drag and drop functionality (cf. ❶ in Fig. 2). Similar to case handling, tasks can also be skipped or delegated to another responsible person if necessary [3].

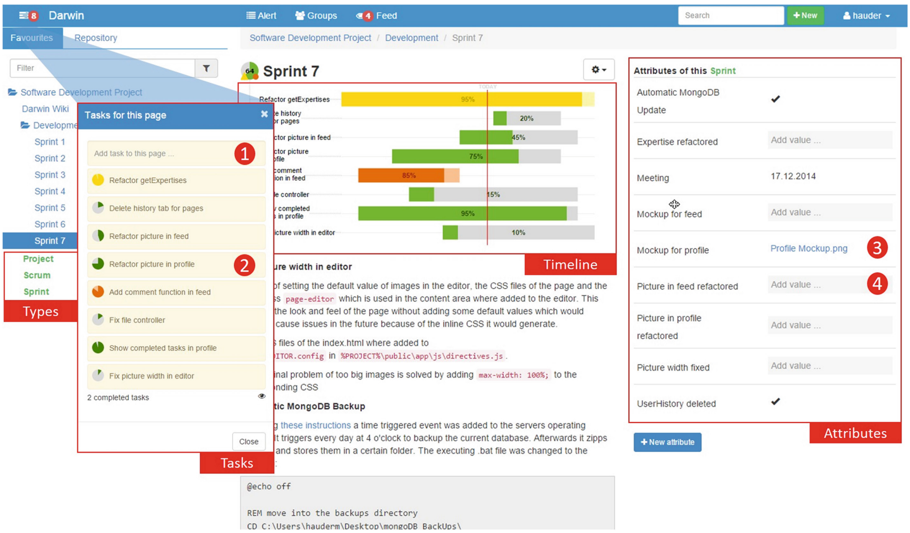

Fig. 2. Structuring concepts with the task progress visualization embedded in a wiki page

Attributes. Attributes can be assigned to tasks as mandatory work results, i.e. to complete the task *refactor picture in profile* ❷ a file for the attribute *mockup for profile* ❸ has to be uploaded and the attribute *picture in feed refactored* ❹ checked. In combination with the tasks it is possible to specify who is responsible for completing which attributes, and by when. Attributes have a name string and a grey box for the value. Values can have a simple type, e.g. file, string, date, boolean, enumeration, or a complex type with a link to another wiki page. Values with links to other wiki pages can be used to model hierarchical information structures, e.g. it can be specified that the linked wiki pages need to have certain types. Similar to our previous work, attributes and tasks from other users can be suggested to increase consistent usage of terms [20]. During the creation of an attribute end-users specify access rights to define who is allowed to edit and read attribute values. This flexible information structuring mechanism is used to integrate data in the knowledge-intensive processes which is indispensable for knowledge work (cf. data integration challenge and late data modeling requirement identified in the literature reviews in [5, 13]).

Types. End-users can assign a type for every wiki page to describe what kind of content is presented on the page. An example of a type is shown at the top of the attribute box in Fig. 2 as *sprint*. The main purpose of the type is to dynamically determine what attributes and tasks are predefined or have been added by other users. Predefined attributes and tasks of a type are immediately added when a new wiki page with this specific type is created. In addition, end-users can collaboratively extend these pages with new tasks and attributes at run-time. Predefined tasks can be created based on the collaboratively defined extensions by end-users or in advance by modeling experts in an interactive and web-based CMMN editor. This editor is only visible for modeling experts and can be accessed by clicking on the name of a particular type, e.g. *sprint* in the file explorer or above the attribute list in Fig. 2.

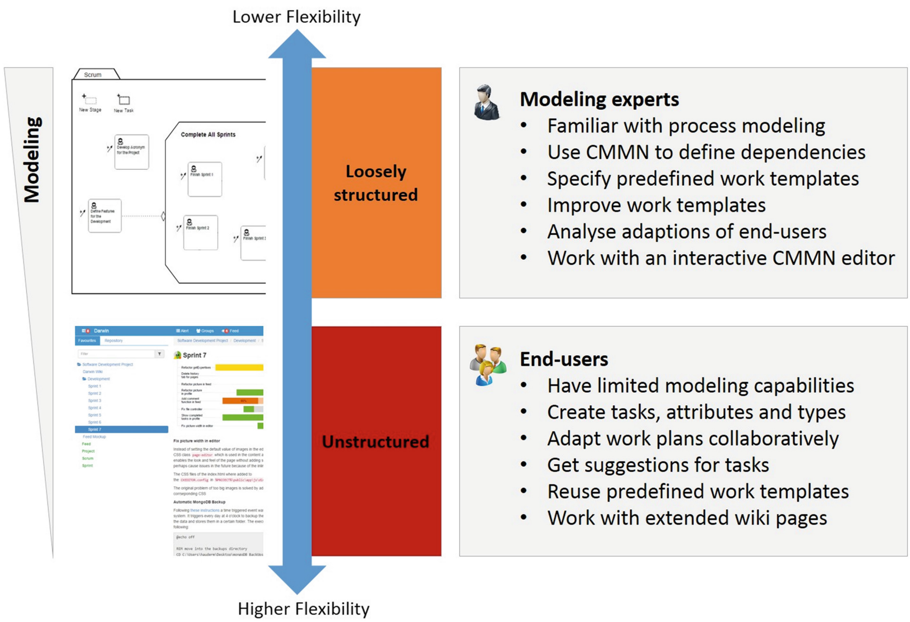

Fig. 3. Modeling experts use an interactive CMMN editor to define dependencies for tasks

Within this CMMN editor dependencies between tasks as well as completely new tasks and stages can be created by a modeling expert (cf. Fig. 3). Every task in this CMMN editor corresponds to a task that is shown in the side window for tasks on wiki pages. CMMN allows an experienced modeling expert to define the behavior of a type with stages and define logical dependencies, i.e. the task *define features for the development* has to be completed before all tasks in the stage *complete all sprints* are enabled. Tasks created by end-users are shown with simple usage statistics and the modeling experts can make them predefined for future work plans of this type. The main advantage of our approach is that end-users are able to structure and adapt a knowledge-intensive process without having to learn or understand CMMN. The Darwin Wiki uses only Stages, HumanTasks, CaseTasks, and Sentries from the CMMN 1.0 specification. The relationships between these types with the previously explained tasks and attributes are presented with our meta-model next.

3.3 Implementation

The Darwin Wiki builds on the experiences made with Hybrid Wikis that were developed and used as productive system in our group since 2009 [20]. Darwin extends Hybrid Wikis with structuring concepts for tasks, complex information structure hierarchies, user interface concepts to visualize the progress of tasks, more detailed role concepts for attributes as well as a declarative process model. The underlying meta-model of the Darwin Wiki is shown in Fig. 4 and explained next.

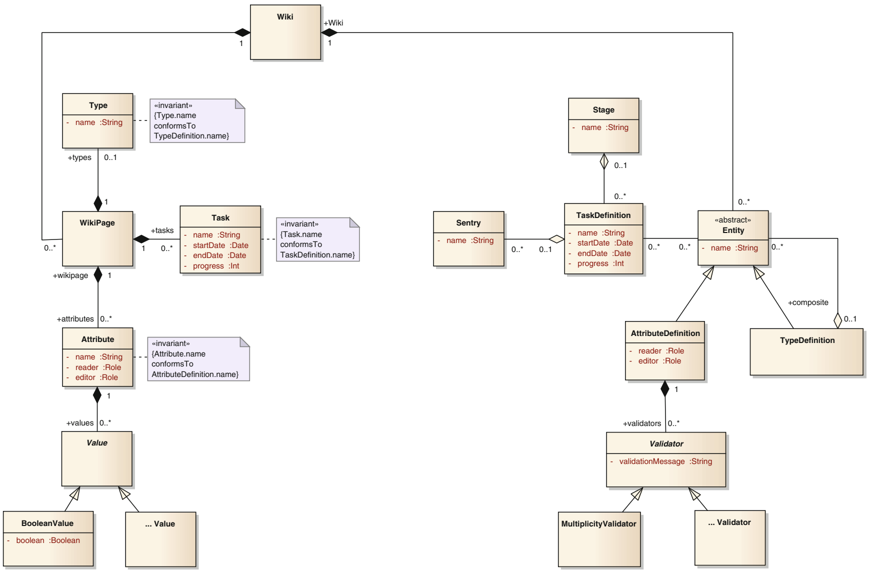

Fig. 4. Simplified conceptual meta-model of the Darwin Wiki showing the core elements in the prototype to support the collaborative structuring of processes for knowledge work

Wiki pages are extended with `Tasks`, `Attributes`, and a `Type` to structure knowledge-intensive processes in work plans. Additional concepts for `Task-Definition`, `TypeDefinition`, and `AttributeDefinition` are used to specify integrity constraints for predefined work templates. Work plans are structured by end-users, while work templates are defined by modeling experts. A type definition is loosely coupled to a type to enable the creation of new types by end-users on the fly at run-time. The same applies to attribute definition and attribute as well as task definition and task. Tasks are mapped either to HumanTasks or CaseTasks in CMMN depending on the assigned `Entities`. `Stages` are concepts that serve as containers for TaskDefinitions and track the behavior (or "episodes" [8, p. 28]) of a wiki page at run-time. The `TypeDefinition` is a composite that facilitates the modeling of complex hierarchical information and task structures. Tasks that reference other pages are automatically finished after all subtasks on these referenced pages are completed. `Sentries` specify logical entry and exit criteria between tasks and stages, e.g. a task is only enabled after certain other tasks are completed.

4 Findings

Figure 5 shows the main result of our evaluation with the work template that emerged for the development process within a project with four students in a practical course at the Technical University Munich over a time span of three months. This practical course is for master's level students in the computer science faculty and the goal for these students is the development of a software application. The project started on 22nd October 2014 and will continue until the end of February 2015. The students are following a simple Scrum process and have regular sprint meetings every week in which tasks are assigned. One team is using the Darwin Wiki to support the development process, while another team with four students is working with the project management tool Trello[2]. The students started with an empty wiki and they collaboratively structured this knowledge-intensive process gradually with a work template in just 14 days. The advisor of the course served as modeling expert. None of these students had previous experience in business process modeling.

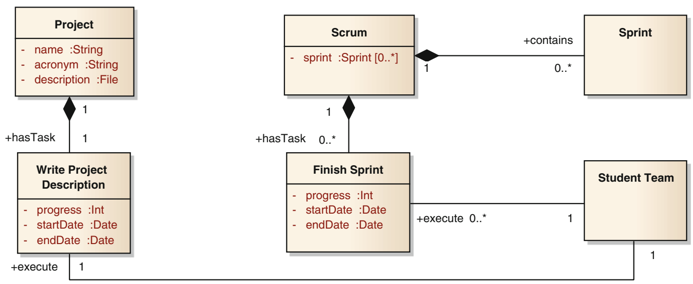

Fig. 5. Emerged work template for the software development project with four students

[2] https://trello.com/, last accessed on: 2014-12-04.

Each `project` is described on a wiki page and contains a predefined task `write project description`. The `sprints` are structured in a composite type definition called `scrum`, which contains all sprints with a predefined task `finish sprint` for every sprint that is assigned to the `student team`. The finish sprint tasks are automatically completed as soon as all subtasks on the assigned sprint are completed. Within the wiki pages for a sprint the required deliverables are documented as attributes and associated with tasks, e.g. mockups, development documents, checklist items consisting of boolean attribute values. Students resolve and create new tasks on their own if necessary to enable a self-organizing working mode during the sprints, which would be very difficult to achieve with a rigid and predefined workflow. Within seven weeks the students completed seven sprints, in which they created 7.3 tasks on average for every sprint. There were 0.8 attributes associated with each task on average. Examples for tasks that have been created are 'Refactor getExpertises', 'Add comment function in feed', and 'Show completed tasks in profile' (cf. Fig. 2). Attributes that were created by the students are e.g. 'Mapping of features to design principles' (as file), 'Mockup for feed' (as file), and '(Project) acronym' (as string).

Although statistical conclusions and hypothesis testing is not possible due to the limited number of participants, we have observed that non-expert users are able to easily structure a knowledge-intensive process. In addition, we conducted group interviews using open-ended and closed-ended questions to compare the Trello and the Darwin Wiki teams. We used group interviews to leverage the interactions of the group members to stimulate their experiences and filter out extreme answers. The results of the closed-ended questions are illustrated in Fig. 6 as spider diagrams and we used the open-ended questions to further elaborate these results next. The results indicate that the proposed solution is able to improve process support without losing the flexibility which is required for this knowledge-intensive process.

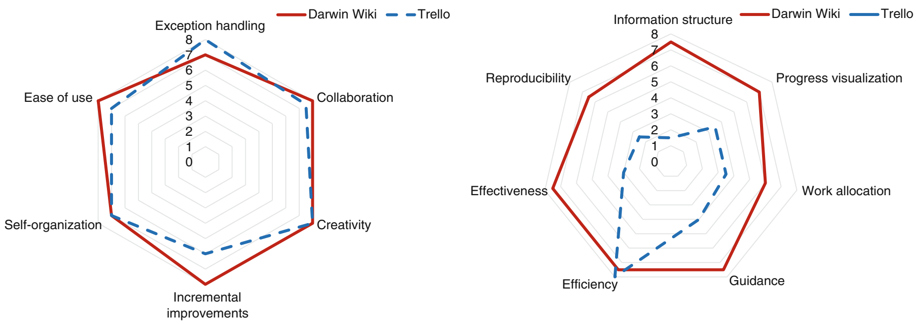

Fig. 6. Comparison of flexibility (left) and process support (right) for Darwin Wiki and Trello

We measured the flexibility of the tools with six variables. In general both teams confirmed that Trello as well as the Darwin Wiki provide very good means to adapt their processes. Trello has its strength in particular in the handling of exceptions with simple cards that are organized in subsequent columns e.g. to do, doing, and done. New cards can be easily added by end-users and moved between the columns when the project progresses. The Darwin Wiki is slightly more suitable for incremental improvements through the reuse of previously created structures e.g. sprints. Large differences between both solutions could be observed for process support, which was measured with seven variables. The Darwin Wiki allows the structuring of information to be organized e.g. presentations, mockups or sprints. Due to limitations of the Trello tool the team was using a separate web storage solution to share files and only copied the links to the files in the tool. Visualization of task progress is only possible on a coarse-grained level in Trello. In the Darwin Wiki expertise tags and the progress of the tasks were used to allocate work to students that have certain skills, while Trello only supports colors that can be added on cards to indicate priorities. Guidance during the process is worse in Trello since it is not possible to define complex hierarchies and dependencies between cards, whereas this was achieved through subtasks and different task start and end dates in Darwin. The efficiency was rated similarly for both tools and the effectiveness was rated as better in Darwin due to the attributes that were used to specify mandatory results for tasks. Reproducibility was rated in favor of Darwin because of the reuse of sprints every week.

5 Conclusion

We presented a solution that empowers non-expert users to structure knowledge-intensive processes. These process structures emerge bottom-up through contributions of end-users at run-time. Our implementation Darwin Wiki achieves this by employing lightweight structuring concepts such as attributes, tasks and types as metaphors that require no understanding of process modeling notations for end-users.
Evaluation: Based on a practical application with students conducting a software development project with Scrum, it was shown that such non-expert users were readily able to structure knowledge-intensive processes. A qualitative comparison to a widely used agile project management solution that can be applied for these kinds of processes (Trello) was conducted. As compared with Trello, the Darwin Wiki dramatically improves the organization of information, reproducibility, and guidance within the software development process without limiting users' flexibility.
Limitations: Due to the small number of students in this practical setting the results for this evaluation cannot be generalized. Knowledge-intensive processes with substantially different dimensions might be more difficult to structure for end-users or have other requirements that we didn't experience in our evaluation. Furthermore, end-users with more experience and different backgrounds have to be investigated.
Future Work: Guidance and recommendation techniques for tasks, constraints and attributes that are automatically suggested on wiki pages have to be improved. These recommendations can be based on activities performed by other users in related contexts. Automated learning of process models could be investigated to improve the work templates with every iteration.

Acknowledgement. This research has been sponsored in part by the German Federal Ministry of Education and Research (BMBF) with grant number TUM: 01IS12057.

References

1. Brynjolfsson, E., McAfee, A.: Race against the Machine: How the Digital Revolution is Accelerating Innovation, Driving Productivity, and Irreversibly Transforming Employment and the Economy. Digital Frontier Press, Lexington (2011)
2. Davenport, T.H.: Thinking for a living: how to get better performances and results from knowledge workers. Harvard Business Press, Boston (2005)
3. Van der Aalst, W., Stoffele, M., Wamelink, J.: Case handling in construction. Autom. Constr. **12**(3), 303–320 (2003)
4. Strong, D.M., Miller, S.M.: Exceptions and exception handling in computerized information processes. ACM Trans. Inf. Syst. (TOIS) **13**(2), 206–233 (1995)
5. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. 1–29 (2013)
6. Van Der Aalst, W.M.P., Berens, P.J.S.: Beyond workflow management: product-driven case handling. In: Proceedings of the 2001 International ACM SIGGROUP, pp. 42–51. ACM Press, New York (2001)
7. Reijers, H.A., Rigter, J., Van Der Aalst, W.M.P.: The case handling case. Int. J. Coop. Inf. Syst. **12**(3), 365–391 (2003)
8. OMG. Case Management Model and Notation (CMMN), version 1.0, May 2014. Document formal/2014-05-05
9. Marin, M., Hull, R., Vaculín, R.: Data centric BPM and the emerging case management standard: a short survey. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 24–30. Springer, Heidelberg (2013)
10. Pesic, M., van der Aalst, W.M.: A declarative approach for flexible business processes management. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 169–180. Springer, Heidelberg (2006)
11. Muehlen, Mz, Recker, J.: How much language is enough? theoretical and practical use of the business process modeling notation. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 465–479. Springer, Heidelberg (2008)
12. Marin, M.A., Lotriet, H., Van Der Poll, J.A.: Measuring method complexity of the case management modeling and notation (CMMN). In: Proceedings of the Southern African Institute for Computer Scientist and Information Technologists Annual Conference 2014 on SAICSIT 2014 Empowered by Technology. ACM (2014)
13. Hauder, M., Pigat, S., Matthes, F.: Research challenges in adaptive case management: a literature review. In: 3rd International Workshop on Adaptive Case Management and other non-workflow approaches to BPM (AdaptiveCM), Ulm, Germany (2014)
14. Mundbrod, N., Reichert, M.: Process-aware task management support for knowledge-intensive business processes: findings, challenges, requirements. In: 3rd International Workshop on Adaptive Case Management and other non-workflow approaches to BPM (AdaptiveCM), Ulm, Germany (2014)
15. Bry, F., Schaffert, S., Vrandečić, D., Weiand, K.: Semantic Wikis: approaches, applications, and perspectives. In: Eiter, T., Krennwallner, T. (eds.) Reasoning Web 2012. LNCS, vol. 7487, pp. 329–369. Springer, Heidelberg (2012)
16. Haake, A., Lukosch, S., Schümmer, T.: Wiki-templates: adding structure support to wikis on demand. In WikiSym 2005, pp. 41–51. ACM Press, New York (2005)
17. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, New York (1995)

18. Michel, F., Gil, Y., Ratnakar, V., Hauder, M.: A task-centered interface for on-line collaboration in science. In: Proceedings of the 20th International ACM Conference on Intelligent User Interfaces (IUI), Atlanta, USA (2015)
19. Voigt, S., Fuchs-Kittowski, F., Gohr, A.: Structured Wikis: application oriented use cases. In: Proceedings of the International Symposium on Open Collaboration. ACM (2014)
20. Matthes, F., Neubert, C., Steinhoff, A.: Hybrid Wikis: empowering users to collaboratively structure information. In: ICSOFT (1) (2011)
21. Swenson, K.D.: Designing for an innovative learning organization. In: 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC). IEEE (2013)
22. Schönig, S., Zeising, M., Jablonski, S.: Supporting collaborative work by learning process models and patterns from cases. In: 2013 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). IEEE (2013)