**Submitted on:** 25.06.2017

**Satellite Meeting - Knowledge Management Section**
New Directions in Knowledge Management

# Collaborative Working and Knowledge Sharing in the Enterprise Wiki: How Teams Develop Concepts Using Sprints

**Sven Strobel**
Competence Centre for Non-Textual Materials, German National Library of Science and Technology (TIB), Hanover, Germany.
sven.strobel@tib.eu

## Abstract:

*Wikis have long been established in the library sector. Yet, librarians tend to only use them when compiling lists of links or files. The Competence Centre for Non-Textual Materials (KNM) at the German National Library of Science and Technology (TIB) rather uses the wiki as a working platform for collaboration, knowledge sharing, and management of tasks and workflows. The KNM team defined an agile sprint workflow and implemented it in the wiki in order to develop requirements specifications for the TIB AV-Portal. The KNM method combines a wiki-based management of workflows with agile methods. This method accelerates complex decision making, structures and standardises the processes, intensifies collaboration, and documents everything in the wiki. The paper presents the execution, roles and artefacts of the KNM sprint workflow. It also shows what the KNM method has in common with Scrum and how it differs from it. The KNM method is not limited to software development. It can be applied to concepts other than requirements specifications, such as research applications, guidelines or manuals.*

**Keywords:** knowledge management, agile project management, Scrum, wiki, collaborative working

## 1. Introduction

Collaborative working implies that two or more people are working together in order to achieve a common goal.[1] Within the working environment, teams are the basic unit for collaboration and conducting tasks that are high in complexity. Sundstrom, De Meuse & Futrell define teams as ‘small groups of interdependent individuals who share responsibility for outcomes for their organisations’.[2] The single activities of the team members have to be coordinated in order to achieve the common goal. The coordination of these activities can be supported by social software such as enterprise wikis. Enterprise wikis have become increasingly popular within organisations since they are powerful and easy to use. They offer functionalities which support the project management and knowledge management of the organisation. At the German National Library of Science and Technology (TIB), we use the wiki software Confluence by Atlassian.

Today, enterprise wikis are widely established within the library sector.[3] Yet, librarians tend to only use them when compiling lists of links or files. The opportunities wikis offer for collaboration, knowledge sharing and management are rarely exploited.[4] In the use case presented here, TIB members use the wiki as a tool for project management and knowledge management. The wiki supports the team in managing the various project phases, i.e. planning, executing, controlling and closing the project.[5] The single activities of the team members can be documented, scheduled, sequenced or parallelised, as well as monitored in the wiki. Furthermore, the wiki helps to acquire, distribute, make use of, preserve, and continuously evaluate and improve the knowledge of the organisation.[6] Using the enterprise wiki as a platform to support project management and knowledge management has the following benefits:

- *'High transparency*: All the relevant information of the project is displayed in the wiki and may be retrieved by the team members at any time
- *Central collaboration platform*: Documents can be centrally collected, linked, and commonly edited and developed by multiple users. They are always up-to-date and develop organically. Older versions can easily be restored, which ensures better protection of the content. Users are notified via email about changes in the document. External project partners may also receive access to certain wiki documents
- *Documentation* & *reusability*: All the relevant information of the project is documented in the wiki, which allows the team members to reconstruct, at a later point in time, why the team has decided in favour of one or the other option. Moreover, the approach, findings or ideas of the project can be reused in other projects'[7]

At TIB, the development team of the Competence Centre for Non-Textual Materials (KNM) uses the enterprise wiki to develop concepts in a collaborative effort.[8] These concepts are requirements specifications for software features of the scientific video portal of TIB (TIB AV-Portal).[9] The KNM team writes the requirements specifications using sprints, which are planned, executed and controlled in the enterprise wiki. Sprints originally come from the agile management framework Scrum.[10] The KNM method was inspired by Scrum. However, we do not work in the Scrum framework but rather with the sprint idea from Scrum. Scrum is a 'framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value'.[11] Scrum is based on the following principles:

- A self-organising and cross-functional team
- A prioritisation of the requirements
- An iterative and incremental procedure to deliver product increments within a fixed time span of 2-4 weeks. The short cyclic intervals in which product increments are created are called 'sprints'
- A high transparency
- A regular inspection of products, processes and methods, and a flexible adaptation to changing requirements[12]

The KNM method described in this paper is not limited to product development. It can be transferred to other areas of project management and knowledge management. More generally, it can be considered as a method for the computer-supported collaborative development of concepts. In fact, many issues of this paper are linked to the multidisciplinary research field of *Computer-Supported Cooperative Work*.[13] The structure of the paper is as follows: section 2 outlines the flow of projects at KNM. It discusses how the KNM development team prioritises the requirements and incorporates them into sprints. Section 3 illustrates the KNM-specific sprint workflow and its roles: sprint manager and sprint participant. Section 4 presents the artefacts of the KNM sprint workflow, i.e. sprint document and sprint overview. Section 5 explains what the KNM method has in common with Scrum and how it differs from it. Section 6 concludes the paper.

## 2. Flow of Projects

The KNM development team starts by gathering requirements for the development of the TIB AV-Portal in a *list of ideas*. The requirements are normally expressed in user stories such as ‘As a user, I want to add videos to a memory list’. The *list of ideas* can be extended at any time. The requirements may be proposed or inspired by the library management, other departments, customers, users, project partners, or by the KNM team itself. The KNM team consists of a product manager, project manager, Usability expert, IT specialist, database manager, and analyst. The team prioritises the requirements of the *list of ideas* according to five priorities (1 > 2 > 3 > 4 > none).

(Insert figure 1)

The requirements of the *list of ideas* are transferred to the *list of implementations* in order of priority (cf. figure 1). They are transferred until the *list of implementations* can be considered as ‘full’, i.e. until the team members’ capacities for the relevant period of time (e.g. in 2017) are used up. The team processes the requirements of the *list of implementations* from top to bottom by incorporating one requirement after the other into a sprint. The team works on only one sprint at a time. During the sprint, the team develops the requirements specification, which is a detailed description of a software component of the TIB AV-Portal. The requirements specification for ‘Adding Videos to a Memory List’, for example, might describe how the user performs the use case, which user interface elements are needed and where they are placed, which icons and wording are used, which data bases are queried etc. Having finished the sprint, the team hands over the specification to external software developers, who are assigned to implement the software component. The KNM team tests the implementations and organises debugging with the software developers. After debugging, the implementations are deployed in the production system.

## 3. The KNM Sprint Workflow

This section provides a detailed description of the KNM sprint workflow. The KNM development team does not work in the Scrum framework but rather uses the sprint idea from Scrum. At KNM, we developed our own sprint workflow containing roles and artefacts that are not defined by Scrum.[14] We define a sprint as a short working cycle (4 weeks), in which a concept is collaboratively and mainly virtually developed and approved. The total set of concepts is incrementally developed by the repeated execution of sprints. In our use case, the concepts are requirements specifications for software features of the TIB AV-Portal.

The KNM sprint workflow involves a sprint manager and normally several sprint participants. The sprint manager's task is to plan, steer and control the sprint workflow. Furthermore, the sprint manager is 'editor-in-chief' of the requirements specifications. The sprint participants' job, on the other hand, is to complete the tasks assigned to them by the sprint manager. The wiki supports the sprint team in coordinating, consensus finding and cooperating. One member (normally the project manager) of the KNM development team takes up the role of the sprint manager; the other members are sprint participants. In addition, the team members have other roles to fulfil in the project, such as analyst, IT specialist or Usability expert. In terms of the TIB AV-Portal, the team primarily specifies the requirements and implements only few of them. The majority of the requirements are implemented by external software developers.

KNM sprints are devised for small (2-5 people), cross-functional and self-organising teams. The principle 'as many team members as necessary, as little as possible' shall apply. On the one hand, the team has to be large enough to combine all necessary competences. On the other hand, communication and coordination efforts rise as the number of team members increases. Goll & Hommel state that higher communication and administrative efforts can significantly diminish the flexibility and quick adaptation to changes, which are considered as major benefits of an agile working method.[15] Figure 2 illustrates the KNM sprint workflow.

(Insert figure 2)

1. For each requirements specification, the sprint manager prepares a sprint document in the wiki. He assigns scheduled tasks to the sprint participants. Normally, the first task of all sprint participants is to write down the requirements for the particular software component as exactly and comprehensively as possible using the sprint document
2. The sprint participants collaboratively write the requirements specification for the software component. They add comments and responses, additional proposals and modifications. Moreover, they complete the tasks assigned to them by the sprint manager. One task might be, for example, to visualise a new software feature. When they have finished their tasks, the sprint participants check off the corresponding action items in the wiki
3. The sprint manager structures the specification: he rewrites and unifies parts of the text, removes redundancies, incorporates new issues after team meetings, divides the document into several sections, notes down decisions and decision making in info boxes etc. He ensures that a coherent requirements specification emerges from the many individual supplements of the sprint participants. He also checks that the scheduled tasks are completed on time. Finally, he identifies pending decisions and plans a team meeting to clarify them
4. At the team meeting, the team members make final decisions: How can a specific problem be solved? Which alternative should be implemented? etc. Furthermore, the team members identify new tasks and discuss them
5. The sprint manager restructures the specification by incorporating the decisions into the sprint document. He assigns new tasks to the sprint participants and schedules them. (The components 2. – 5. are normally repeated several times)
6. The sprint participants approve the final version of the specification. After that, the specification can be handed over to the external software developers to be implemented

The sprint yields a requirements specification, which the team has collaboratively evaluated, developed and approved. The key benefit of a sprint is that it helps the team to facilitate and accelerate complex decision making. The sprint workflow is mainly performed in the wiki. The team meeting is the only part of the workflow which is not virtually processed. This non-virtual component, however, is indispensable. The wiki significantly supports the team to collaboratively write and structure the specification, plan and monitor the tasks, and identify pending decisions. However, the decisions are not made in the wiki but at a team meeting. It would be too complicated and cumbersome to make complex decisions in the wiki. Votes, the quick exchange of pros and cons, and the dynamic reaction of the team members can be handled much better at a meeting. Nevertheless, decisions can be initiated in wiki discussions. The efficiency of the workflow is based on the very fact that it combines benefits of the virtual and non-virtual components.

## 4. The Artefacts of the KNM Sprint Workflow

The central artefact of the KNM sprint workflow is the sprint document, which is created by the sprint manager and collaboratively worked on by the entire team. All the relevant information for the specification is represented in this wiki document. The team works on only one document for each specification in order to minimise the administrative expense. The sprint document normally contains several sections:

- A section for issues the team has to pursue
- A section including documented decisions and decision making
- A section for issues the external software developers have to consider

The sections can be framed in different colours to be more easily distinguished. The following table summarises the elements of the sprint document.

(Insert table 1)

Some elements of the sprint document are passed to an overview page using a wiki function; these include *title, status, start & end, sprint manager* and *sprint participants* (cf. figure 3). The overview page is the other artefact of the sprint workflow. It helps the sprint manager to control the sprint workflow. Using the overview page, he recognises which sprint documents are currently being processed (status: IN PROGRESS). In these cases, he checks that the tasks assigned to the sprint participants are completed on time. Sprints having the status FINISHED show that the corresponding requirements specifications can be handed over to the software developers. Sprints having the status STOPPED point out that the team has to solve a major problem before being able to continue with the sprint.

(Insert figure 3)

## 5. Similarities & Differences between KNM Method and Scrum

What has the KNM method in common with Scrum (cf. Scrum principles in section 1)?

- A self-organising and cross-functional team develops the requirements specifications
- Requirements are prioritised
- Specifications are developed in an iterative and incremental manner using sprints

- All the relevant information of the project is documented in the wiki and can be retrieved by all team members at any time, ensuring a high transparency
- Specifications, processes and methods are regularly inspected and may be adapted

What distinguishes the KNM method from Scrum?

- The KNM development team has a manager (sprint manager), who coordinates the sprint workflow
- The KNM method does not employ fixed time boxes for the sprints. The processing time needed for the collaborative development of a specific requirements specification is estimated by the team before the sprint starts. It may be adapted (shortened or lengthened) during the course of the sprint
- Given the fact that Scrum is normally used to develop software components, the Scrum literature frequently refers to creating ‘product increments’. The KNM method, however, delivers ‘requirements specifications’
- Scrum entails specific roles (e.g. scrum master, product owner), events (e.g. daily scrum), artefacts and rules, which are not included in the KNM concept. For that reason, the method described here is not Scrum but only inspired by it
- The KNM method deals with the organisation of collaborative working in the enterprise wiki and with the interplay between virtual (wiki) and non-virtual components (team meeting). This issue is compatible with Scrum but is not a special topic of it

## 6. Conclusion

The KNM development team uses the enterprise wiki to collaboratively develop the requirements specifications for the software features of the TIB AV-Portal. The specifications are worked out on the basis of standardised sprints. The advantage of the standardisation is that every sprint participant knows what she or he has to do in each section of the sprint workflow. Despite the high standardisation, the sprint concept is open to adaptations. It is not only the specification, which is developed during the course of the sprint, but also the processes and methods are consistently refined and adapted. For example, we reduced the number of sprint documents the team works on during the sprint from three to one. By concentrating on only one sprint document (and hence on only one requirements specification), we were able to work out the total set of specifications more efficiently.

The sprint document develops ‘organically’ through the incremental editing process. This process has a high transparency since every change is documented in the sprint document and can be reviewed by all team members. Hence, they are able to continuously monitor and influence the specification. The sprint manager coordinates and controls all activities of the workflow using the wiki. The sprint participants, in turn, are supposed to control the completion of their own tasks and to consider dependencies to other participants’ tasks.

All the relevant information for the specification should be documented in only one sprint document in order to keep the administrative burden as small as possible. Once the requirements specifications have been completed, they are archived in the wiki. They can be reused if an existing feature is to be modified or a new feature is to be implemented.

I would like to encourage people to use this method to collaboratively develop concepts other than requirements specifications, such as research applications, articles, manuals, guidelines, instructions, change logs, software code, mockups, UI prototypes, screenplays or taxonomies, just to name a few.

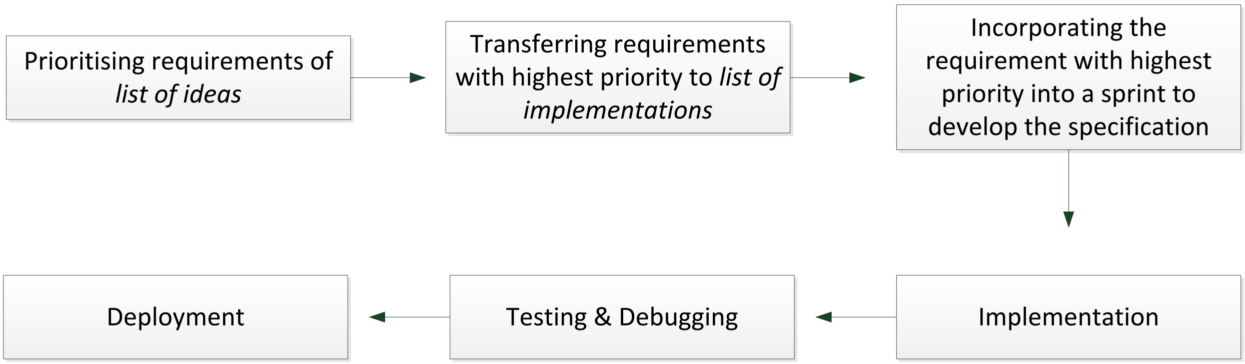

Prioritising requirements of *list of ideas*
Transferring requirements with highest priority to *list of implementations*
Incorporating the requirement with highest priority into a sprint to develop the specification
Implementation
Testing & Debugging
Deployment

Figure 1. Flow of Projects at KNM

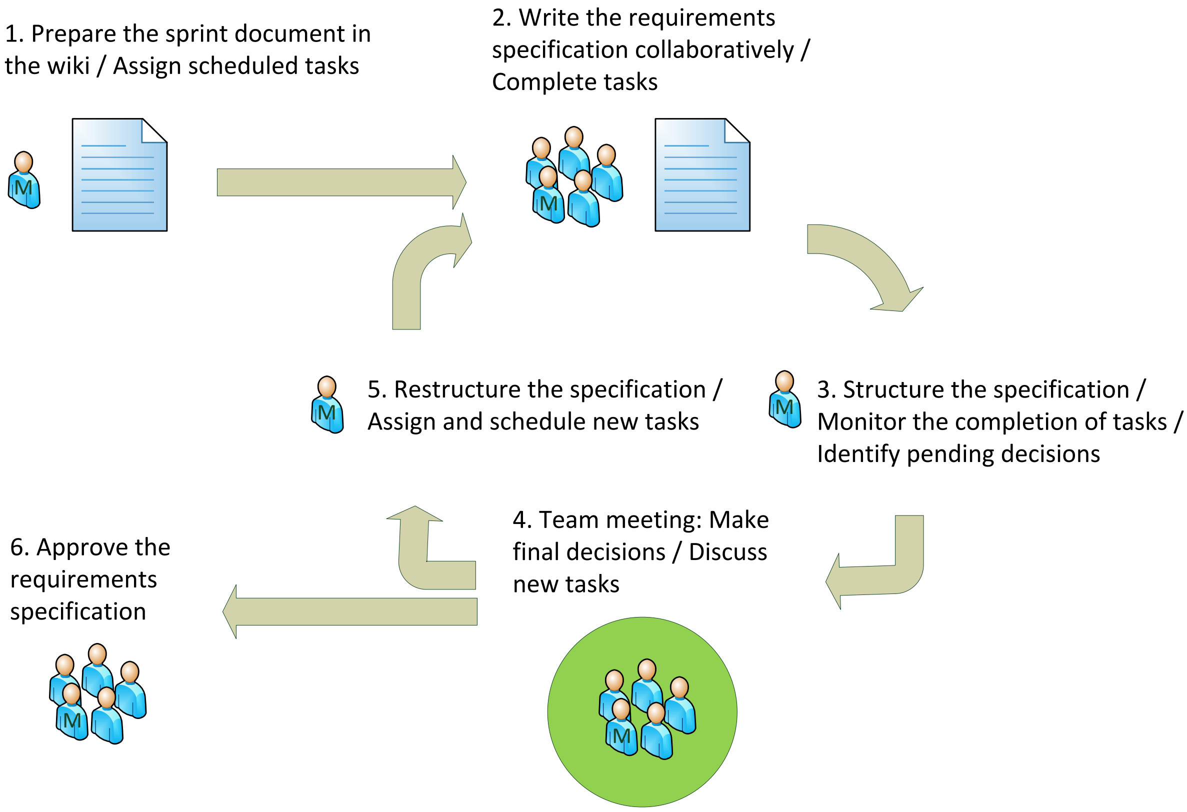
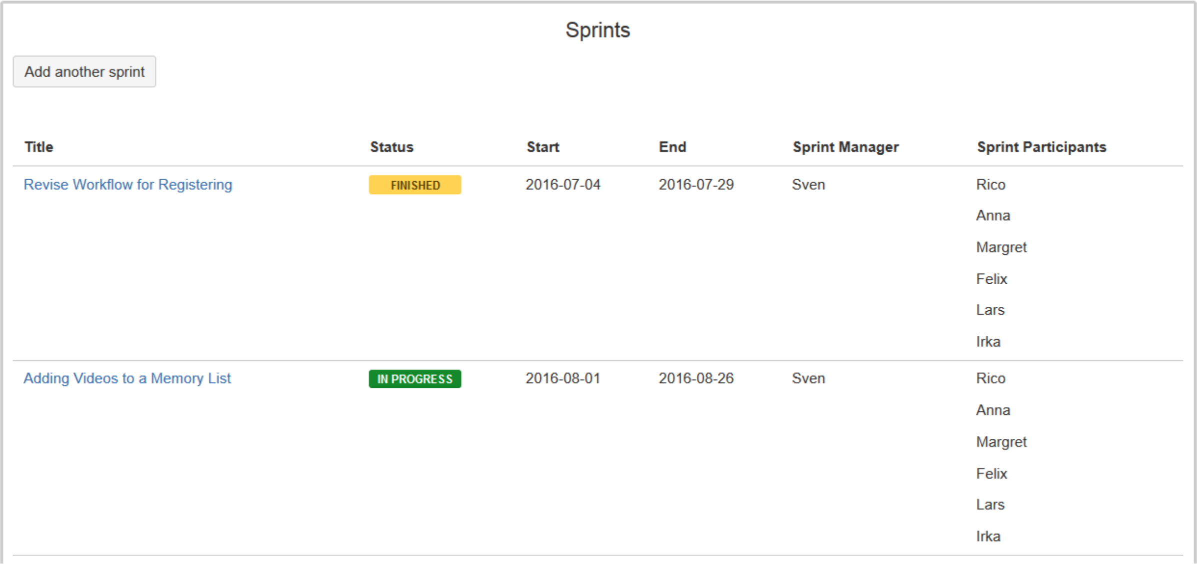

1. Prepare the sprint document in the wiki / Assign scheduled tasks
M
2. Write the requirements specification collaboratively / Complete tasks
M
3. Structure the specification / Monitor the completion of tasks / Identify pending decisions
M
4. Team meeting: Make final decisions / Discuss new tasks
M
5. Restructure the specification / Assign and schedule new tasks
M
6. Approve the requirements specification
M

Figure 2. KNM Sprint Workflow

| Element | Explanation |
|---|---|
| Title | Sprint documents have a title, e.g. ‘Adding Videos to a Memory List’ |
| Status | IN PROGRESS – the sprint is running, i.e. the team is working on the sprint document<br>FINISHED – the sprint participants have approved the specification<br>STOPPED – the sprint was stopped |
| Start & End | Sprints have a tight schedule (4 weeks) |
| Sprint Manager | Naming the sprint manager |
| Sprint Participants | List of sprint participants |
| Section for the Specification | In this section, the sprint participants collaboratively write the specification |
| Tasks | *Who* does *what* by *when* |

Table 1. Elements of the Sprint Document



Sprints

| Title | Status | Start | End | Sprint Manager | Sprint Participants |
|---|---|---|---|---|---|
| Revise Workflow for Registering | FINISHED | 2016-07-04 | 2016-07-29 | Sven | Rico<br>Anna<br>Margret<br>Felix<br>Lars<br>Irka |
| Adding Videos to a Memory List | IN PROGRESS | 2016-08-01 | 2016-08-26 | Sven | Rico<br>Anna<br>Margret<br>Felix<br>Lars<br>Irka |

Figure 3. Sprint Overview

[1] Ignacio J. Martinez-Moyano, "Exploring the Dynamics of Collaboration in Interorganizational Settings," in *Creating a Culture of Collaboration,* ed. Sandy Schuman (San Francisco: Jossey-Bass, 2006): 69-85.

[2] Eric Sundstrom, Keneth De Meuse, & David Futrell, "Work Teams: Applications and Effectiveness," *American Psychologist* 45 (1990): 120-133.

[3] Jeremy Frumkin, "The Wiki and the Digital Library," *OCLC Systems & Services. International Digital Library Perspectives* 21 (2005): 18-22; Yong-Mi Kim & June Abbas, "Adoption of Library 2.0 Functionalities by Academic Libraries and Users: A Knowledge Management Perspective," *The Journal of Academic Librarianship* 36 (2010): 211-218.

[4] Sarah Ann Long, "Exploring the Wiki World: The New Face of Collaboration," *New Library World* 107 (2006): 157-159.

[5] Anja Ebersbach, Markus Glaser & Richard Heigl, *Wiki: Web Collaboration* (Berlin & Heidelberg: Springer, 2006).

[6] Ayelt Komus & Franziska Wauch, *Wikimanagement: Was Unternehmen von Social Software und Web 2.0 lernen können* (München: Oldenbourg, 2008): 167.

[7] Sven Strobel, "Workflow Management of Sprints and Software Tests: Coordination, Consensus, and Cooperation in the Enterprise Wiki of the German National Library of Science and Technology," *Information and Knowledge Management* 6 (2016): 58-64.

[8] www.tib.eu/en/research-development/non-textual-materials

[9] av.tib.eu

[10] Roman Pichler, *Agile Product Management with Scrum: Creating Products that Costumers Love* (London: Addison-Wesley, 2010); Pete Deemer, Gabrielle Benefield, Craig Larman & Bas Vodde, *The Scrum Primer: A Lightweight Guide to the Theory and Practice of Scrum* (Version 2.0, InfoQ.com, 2012); Ken Schwaber & Jeff Sutherland, *The Scrum Guide™, the Definitive Guide to Scrum: The Rules of the Game* (scrum.org, 2013).

[11] Schwaber & Sutherland, "The Scrum Guide™, the Definitive Guide to Scrum," 3.

[12] Schwaber & Sutherland, "The Scrum Guide™, the Definitive Guide to Scrum," 3-8; Joachim Goll & Daniel Hommel, *Mit Scrum zum gewünschten System* (Wiesbaden: Springer, 2015): 84.

[13] John M. Bowers & Stephen Benford, *Studies in Computer-Supported Cooperative Work: Theory, Practice, and Design* (Amsterdam & New York: Elsevier, 1991); Kjeld Schmidt, *Cooperative Work and Coordinative Practices: Contributions to the Conceptual Foundations of Computer-Supported Cooperative Work (CSCW)* (London: Springer, 2011); Sean P. Goggins, Isa Jahnke & Volker Wulf, *Computer-Supported Collaborative Learning at the Workplace* (New York: Springer, 2013).

[14] Sven Strobel, "Einsatz von Sprints in der Produktentwicklung der Technischen Informationsbibliothek," *Bub* 67 (2015): 713-715; Sven Strobel, "Developing the Web Portals of the German National Library of Science and Technology: Tools and Workflows Used," *Qualitative and Quantitative Methods in Libraries (QQML)* 5 (2016): 79-88; Sven Strobel, "Wiki-unterstütztes Prozessmanagement von agilen Workflows an der Technischen Informationsbibliothek," *Blog 'Agile Verwaltung'* (https://agile-verwaltung.org/2017/02/23/wiki-unterstuetztes-prozessmanagement-von-agilen-workflows-an-der-technischen-informationsbibliothek, 2017).

[15] Goll & Hommel, "Mit Scrum zum gewünschten System," 103.