International Journal
of Applied Information Technology
http://journals.telkomuniversity.ac.id/ijait/

Application for Final Project Collaboration and Management in School of Applied Science Telkom University

Robbi Hendriyanto [a,*], Patrick Adolf Telnoni [b]

[a,b] *School of Applied Science, Telkom University, Indonesia*
robbi@tass.telkomuniversity.ac.id, patrick.telnoni@tass.telkomuniversity.ac.id

ARTICLE INFO

Received 12 October 2017
Revised 04 June 2018
Accepted 07 June 2018
Available online 10 August 2018

Keywords
collaboration, kanban, final project

ABSTRACT

The final project is the last stage of the students to complete the lecture. In accordance with academic rules at Telkom University allow final project work done in groups. This is to improve results more complex than the results of the final project obtained individually. But there are problems faced by students and lecturers who join the final project team, which is how to work on the final project together and monitoring the progress of each student in the team. This is due to differences in location and time in work. On the lecturer, side finds constraints that are how to be able to monitor the progress of each team member in detail. In this paper proposed a solution to overcome the problem is to apply for collaborative information technology support. The collaborative method used the Kanban's method, where each work will be divided into small parts that can be done together. Application for Final Project Collaboration and Management in School of Applied Science Telkom University was developed with Software Development Life Cycle (SDLC) Prototype method using PHP programming scripts and MySQL database. This application will manage the final project, especially in groups, facilitating the final project team to define all parts of the work of each team member who can collaborate with each other and communicate in each job. Lecturers as project supervisors can participate in collaborations and discussions in every work.

* Corresponding author at:
School of Applied Science, Telkom University,
Jl. Telekomunikasi No. 1, Terusan Buah Batu, Bandung, 40257
Indonesia.
E-mail address: robbi@tass.telkomuniversity.ac.id

ORCID ID:
First Author: 0000-0003-3502-6887

https://doi.org/10.25124/ijait.v2i02.955
Paper_reg_number IJAIT000020206 2018

1. Introduction

According to Academic Rule Chapter III Education Implementation System verse 13 of Telkom University, Final Project in Diploma degree is an activity to evaluate Hard Skills of diploma student in Diploma 3 and Diploma 4 degree by creating and developing a product or solving a practical problem in line with his academic background. According to Academic Rule, a final project can be conducted personally or in groups [1]. In fact, communication problem often rises during conducting the final project.

Final project requires constant and detailed supervision. Commonly, the final project involves student and a lecturer. The lecturer act as the supervisor, ensuring that the student will not neglect and giving advice to the student. When a final project is conducted by more than one student, the synchronization, communication and monitoring problem from each student become relevant.

In project development, the team will spend time communicating, coordinating and evaluating, especially with the different location of the project. Collaboration tools have the potential to help overcome the time limitations and location of workers to communicate, coordinate, conduct job evaluation [2]. Lomas define collaboration as an activity involving one or more individual to produce or create something [3].

Collaboration tools are a device to ease a task conducted by one or many people for certain purpose. In this research, collaboration tools to create an application for collaboration management dedicated to the final project, where the final project will be divided into smaller chunks and easily conducted task.

2. Research Methodology

Software Development Life Cycle (SDLC) Prototype to build the model of this prototype application is shown in Figure 1. The prototype built using an idea to create a full system or a subsystem in the experiment phase called prototype [4]. Prototype method has the advantage to provide the system for the user to interact with the system at once, so that user can give feedback at once, thus resulting on the product which perfectly matches user's requirement.

1. Analysis Requirement. Initial stages will be communication between developers with clients to gather the need for software. Conducting interviews with students and lecturers at the School of Applied Science at Telkom University. In addition to observing the final project implementation process carried out during one semester. All needs will be translated into modelling applications such as UML and ER-diagram and mockup interface.
2. Development of Prototype. The next step is to build the final project collaboration application using the web programming language based on the model and interface mockup that has been.
3. User Feedback Testing. The prototype that has been built will be user feedback to see the functionality of the built application is in accordance with the needs of the user or still required an improvement. If it still exists, then it will be repeated at the Development of Prototype stage.
4. Final Project Release Completed applications based on functionality and user needs can be directly used and considered to be a final application.

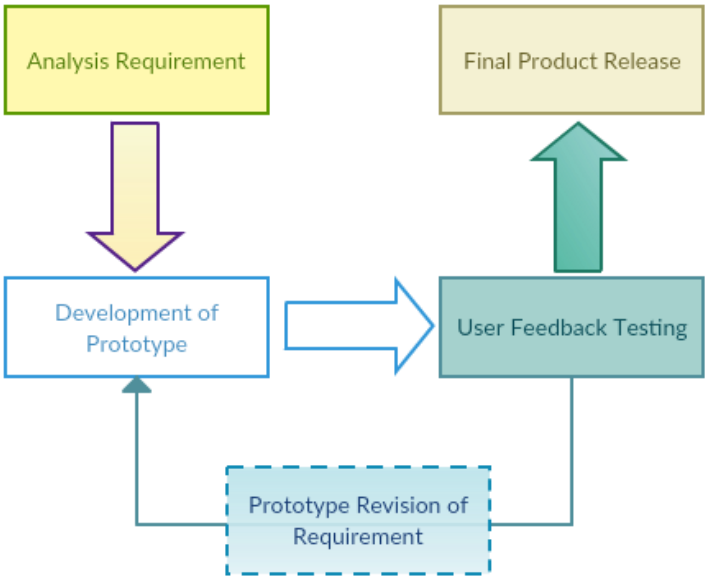

Figure 1 Software Development Life Cycle – Prototype

3. Related Work

Collaborative information systems can assist IT consultants and their client companies in implementing IT project management projects, covering project, task, milestone, roles, project teams, files, discussions, comments, and can also display Gantt charts and project reports [5].

Team collaboration is crucial to the success of a project with multiple developers. When team members are scattered in different places, the individual's awareness of other activities decreases due to communication barriers. This will disrupt team performance and collaborative project work. With a collaborative application can build team awareness by sharing progress and information among team members [6].

Kanban has a positive effect on helping software developers team especially in communicating and collaborating. Kanban helps teams work together to identify and coordinate work. The next phase, able to facilitate team members who already have a good performance progress to be able to collaborate with other team members. Kanban can also help team members collectively identify and bring up missing tasks to keep the development alignment aligned with the rest of the team [7].

4. Project Management

According to the project management institute, a project must have its start point and end point. A project has following characteristic [8].

1. A tangible purpose, which is producing a final product or final work result.
2. A project has cost, timeline, the requirement of quality and resource required to consummate the defined scope.
3. Temporary has its deadline.
4. Does not loop.
5. Type and activities can change during the project are conducted.

A feature of the software project according to Hughes and Cotterell divided into 3 parts [9], as follows.

1. Invisibility, where a progress of a software project cannot be seen immediately.
2. Complexity, software development contains more complexity than another engineering project.
3. Flexibility, a software development project has convenience upon facing change.

Meredith and Mantel stated that project can be described as planning, monitoring, and control to every aspect in a project and motivation in it to achieve the purpose of the project with the determined time, cost, quality, and performance [5]. Project management can be derived as an activity to manage resources which is consist of human and non-human to achieve a certain purpose.

5. Kanban Collaboration Method

Kanban is a framework, which enables team and organization to enhance their understanding of how they work. This framework uses a few mechanisms to simplify coordination di the whole system and to maximize flexibility of the resource in the sub-system [10]. Kanban is suitable for continuous integration, which integrates a relatively smaller part of one project to the main branch at will [11]. Kanban also adopted by widely known Trello (http://www.trello.com) as a tool to help project design and management. Trello uses visual board called Kanban Board. Kanban board filled by the user, defining components and product according to the project.

Kanban emphasizes service provision and catalyzes evolutionary change so Kanban can be defined in the following areas [10].

1. Understanding,
2. Agreement,
3. Respect,
4. Leadership,
5. Flow,
6. Transparency,
7. Balance,
8. Collaboration,
9. Customer focus.

6. Result

The following discussion of the translation framework as a software requirement.

6.1. Analysis Process

During the supervision process, a student whether in person or in a team will discuss, enhance, and build the final project as agreed in their seminar contract. Students, which work on their final project in a team will face a few challenges, one of them is collaboration during the development of their product and the other one is progress monitoring by the lecturer. If those things are not done properly, a few consequences will occur, such as failure to merge and synchronize, causing lost information. A technical problem could occur such as a different version of the database and interface design and overlapping functionality. The lecturer will find difficult to track any progress which has been committed by each individual of the team.

6.2. Design Process

Unified Modelling Language (UML) is used to illustrate the software, which will be built. UML notation will be used to do prototype design according to the software requirement analysis.

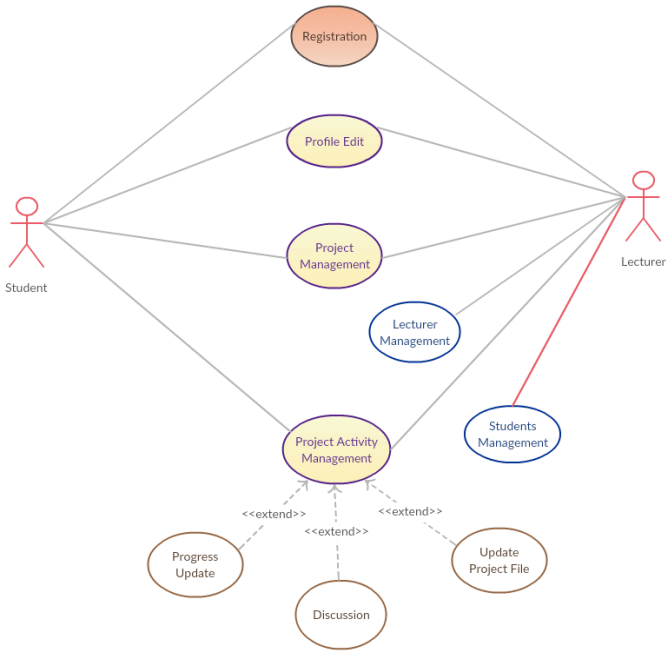

Figure 2 Use case diagram for the prototype

Figure 2 shows use case, which involves two actors, consisting of student and lecturer. Figure 3 shows the table relation used in the prototype design.

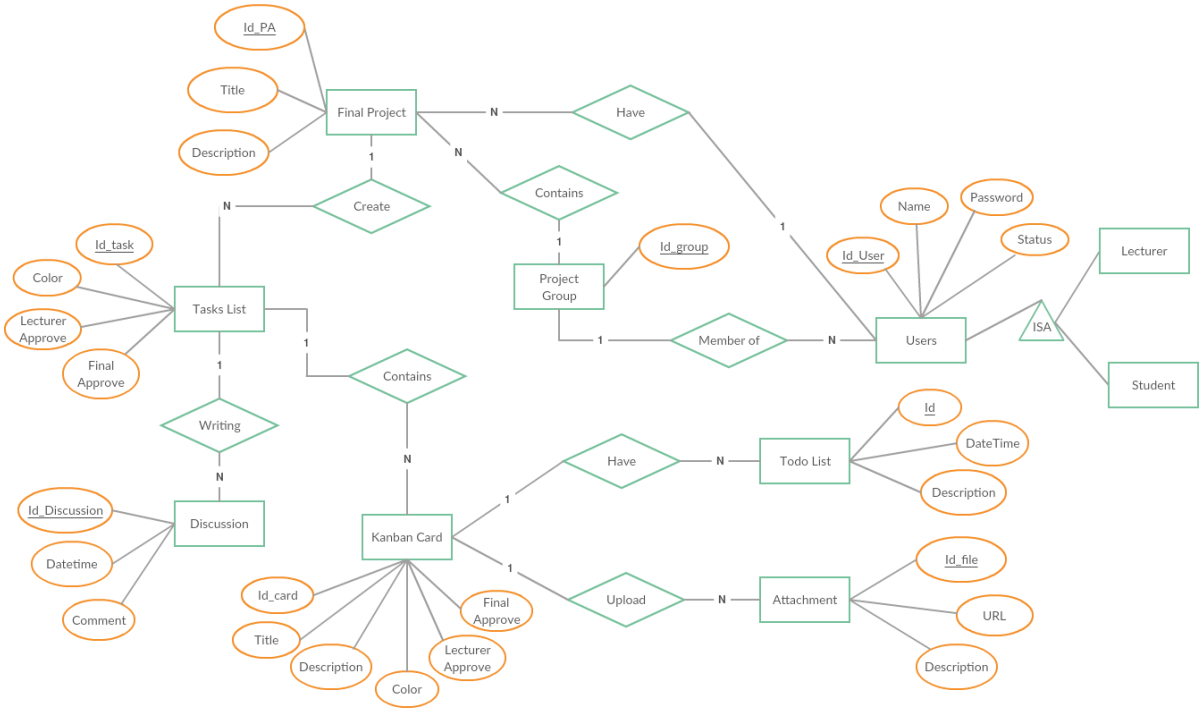

Figure 3 Table relation for the prototype

6.3. Implementation Process

The implementation process will derive from two access controls, one for students and one for a lecturer. Every student and lecturer can log in to access the page. If not registered each student and lecturer can do registration first. Students who have logged in to the content of the project (the title of the project, background, problem formulation, objectives, problem limits, and methodology) and instructions of the lecturers involved.

Some examples of a user interface for registration and login page can see in the following picture.

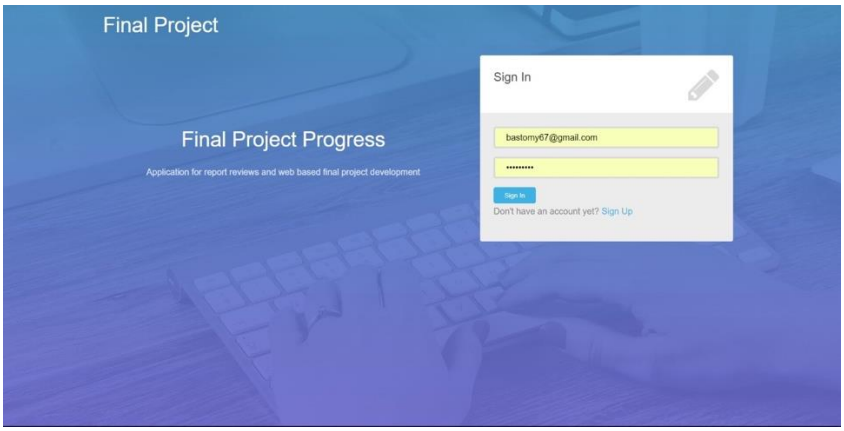

Figure 4 Login page

Figure 4 for the login page where students and lecturers who have access to this page. Figure 5 for the new account registration page.

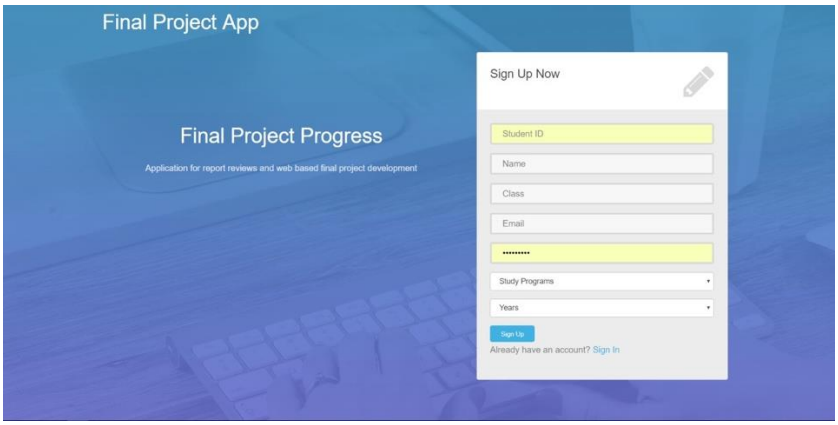

Figure 5 Registration page

Figure 6 shows the view to register the project by entering the data associated with the final project. Any completed final project will require the approval of the lecturer as shown in Figure 7. The teams can add or register its members as shown in Figure 8.

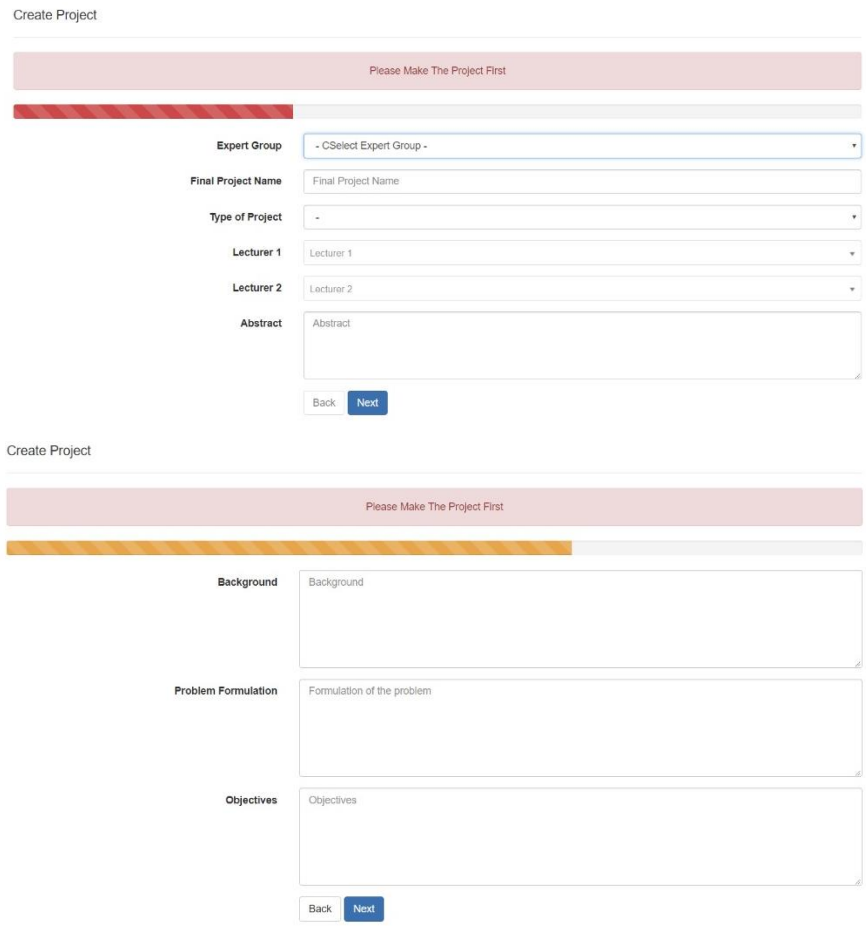

Figure 6 Input form for the project

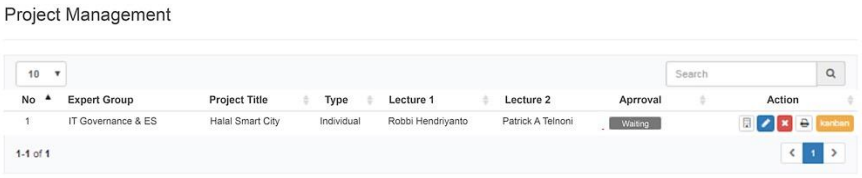

Figure 7 Project registration approval

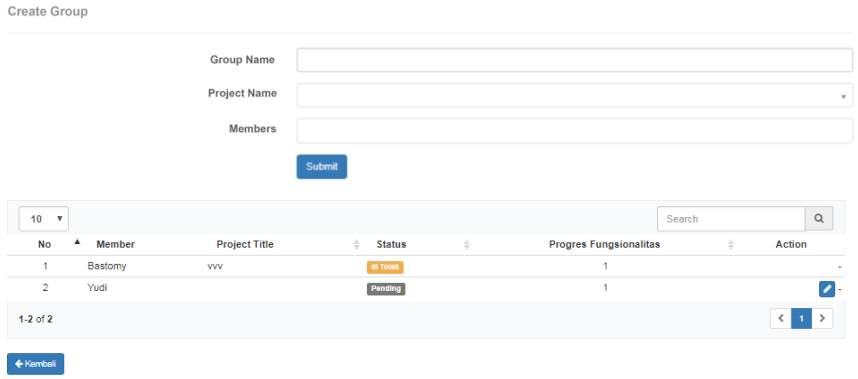

Figure 8 Create a project group

Each project has a collaboration page with Kanban implementation methods. Each frame represents the work done by each team member. Each structure is composed of a list that is a subset of the work. And each record consists of cards

that are a small part of the job. The card has a dialogue to input data about the position, deadline, attachment files, and comments. Figure 9 shows the frame and the card in an example, while Figure 10 shows the details of the card dialogue.

All the progress of the Final Project will be divided into several lists and Kanban cards. Each Kanban card and list if it is completed then the student will request approval from the lecturer. Each student who joined the team can contribute to the list and the card provided in accordance with the obligations of each individual.

Figure 9 Create a project group

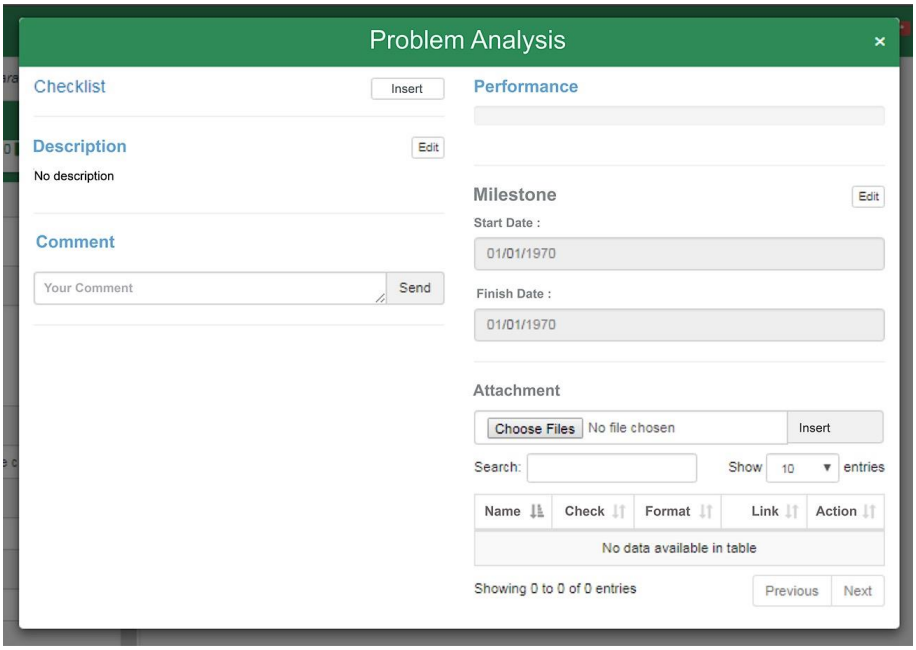

Figure 10 Detail card dialogue

7. Conclusion

This research aims to provide a solution to overcome the limitations of the final project work in groups i.e. location and time. On the lecturers' side, it is difficult to monitor the workmanship of each team member on the work done. Therefore, it is proposed that a collaborative application for final project development based on the Kanban's method can provide collaboration facilities to the final project group of students. In the resulting application can connect student work in groups even if done separately and not at the same time. All work done by the students will be validated by the lecturers, in order to maintain the performance of the final project work as it has been targeted. By using the Kanban method all parts of the final project work done together will be a more visible individual contribution and simultaneously supervised by the lecturer.

Bibliography

[1] Telkom University, "Sistem Penyelenggaraan Pendidikan Telkom University," 2016. [Online]. Available: http://baa.telkomuniversity.ac.id/iii-sistem-penyelenggaraan-pendidikan/. [Accessed November 2017].

[2] I. Gorton, I. Hawryszkiewycz and K. Ragoonaden, "Collaborative tools and processes to support software engineering shift work," *BT Technology Journal*, vol. 15, pp. 189-198, 1997.

[3] C. Lomas, M. Burke and C. L. Page, Collaboration Tools, Educause, 2008.

[4] P. Isaias and T. Issa, "Information System Development Life Cycle Models," in *High Level Models and Methodologies for Information Systems*, New York, Springer , 2015, p. 33.

[5] J. R. Meredith and S. J. Mantel. Jr, Project Management A Managerial Approach, John Wiley & Sons, Inc. , 2009.

[6] L. Hattori and M. Lanza, "Syde: a tool for collaborative software development," in *€ • Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE*, 2010.

[7] N. Oza, F. Fagerholm and J. Münch, "How does Kanban impact communication and collaboration in software engineering teams?," in *International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2013.

[8] Project Management Institute, A Guide To The Project Management Body Of Knowledge, 4th Edition ed., Pennsylvania: Project Management Institute, Inc, 2008.

[9] B. Hughes and M. Cotterell, Software Project Management (Second Edition), Cambridge: McGraw-Hill International, 1999.

[10] A. Reddy, The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban, Pearson Education, 2015.

[11] E. Brechner, Agile Project Management with Kanban, Microsoft Press, 2015.