Tool for scheduling and monitoring teamwork in Project-Based Learning in engineering education

Joan Camps

Universitat Politècnica de Catalunya

joan.camps@upc.edu

Abstract

Project-Based Learning (PBL) is a very relevant pedagogical model for design and for engineering education. Not only encourages the acquisition of knowledge on the subject under study, but also promotes skills such as the ability for cooperation, critical thinking, creative thinking, communication and responsibility. However, it has also some difficulties, e.g., planning can be difficult or students may have different implication degrees. At the evaluation time, teachers generally evaluate the quality of the end product without consider the process of teamwork.

This paper presents a method for planning and monitoring projects based on the agile management framework, Scrum, and on the Concept Knowledge Theory. This method allows, in a simple and automatic way, to obtain all the information needed to monitor and evaluate a project.

1. Introduction

Design is the distinctive activity of engineering, and the pedagogical model that promotes design education is Project-Based Learning (Dym et al, 2005). In Project-Based Learning (PBL) students acquire some knowledge and skills by working, for a certain period of time, on a project as a team. PBL facilitates learning by strengthening the teamwork skills required to entry into a company culture highly oriented to multidisciplinary teams (Mozas-Calvachel and Barba-Colmenero, 2013).

Students develop a product for a specific target group, working through a given problem, and evaluating the process development (Blumenfeld et al, 1991). By using PBL, the students dealt complex issues that require its investigation (Barron, 1998). The students become protagonists of their own learning. In addition, this method offers the opportunity to work relatively independently and it culminates in realistic presentations (Thomas, 2000) or in tangible and observable artifacts that serve as evidence of what they have learned (Rodríguez Montequín et al, 2013).

There is an extensive literature on PBL (Karaman and Celik 2008; Chinnowsky et al, 2006; Doppelt 2003; Frank et al, 2003; Atkinson, 2001; Johnson, 1999). However, PBL can be summarized in four aspects (Filomeno Coelho et al, 2014):

- *theoretical courses, seminars, lectures usually accompany the project itself;*
- *the project work is oriented toward the application of knowledge, but also its acquisition;*
- *a project is usually performed by a small group of students (three to five students);*
- *self-direction and management of time and resources by the students is a key aspect of successful projects.*

The projects face the students into challenges such as design, problem solving, decision making and research activities. Thus, a major argument for applying PBL is that it is one of the most appropriate ways to achieve a competence-based education that integrates knowledge, skills and values (De los Ríos et al, 2010). It promotes skills, such as the ability of cooperation, critical and creative thinking, responsibility, and communication (Moursund, 2003; Chu et al, 2012).

However, this learning framework has also some problems. Milentijevic et al (Milentijevic et al, 2008) pointed that PBL is generally less structured than the traditional approach. Working in a little or non structured environment can introduce significant side effects. E.g., it is difficult to clearly identify the phases of design of a project. In addition, the levels of cooperation and collaboration are difficult to control, making difficult the individual tutoring and evaluation. All of these side effects have been described by several authors (Felder et al, 1993; Karau and Williams, 1993; Kerr and Bruun, 1983; Salomon et Globerson, 1987; Dunlap, 2005; Ertmer et al, 2010).

In this kind of pedagogical model, the teacher's role is that of a consultant, facilitator and evaluator. The professor has to be able to monitor the activities of the students to guide them effectively (Hérold and Ginestie, 2011). However, at the evaluation time, teachers generally evaluate the quality of the final product without considering the teamwork process (Lee and Lim, 2012).

Moreover, another area of research finds that students reach a deeper level of understanding of the academic content if they carry out self-explanations of this content (Chi et al, 1989). Another study (Ainsworth and Loizou, 2003), examines whether there are differences between students who performed the technique of self-explanation from texts and those who use drawings and diagrams. The result of this research shows greater efficiency for those students who learned from diagrams.

2. Tools for monitoring projects in PBL

The aim of this research is to obtain a method that facilitates the scheduling, monitoring and evaluation of projects in the context of PBL, making clear the work process of a team and responding to the difficulties encountered in the previous paragraphs. This method can be compatible with other possible approaches, such as tools for creating groups or techniques of evaluation.

Scheduling, monitoring and evaluation of a project are activities closely linked, as they are concepts related to project management. However, the use of project management standards such as PMBOK guide, does not seem to be the most appropriate approach for a PBL environment. These guides are relatively complex and they were basically designed to handle large projects (Marcelino-Sádaba et al, 2014). Moreover, according to Pant and Baroudi (Pant and Baroudi, 2007), the PMBOK guide prioritizes hard skills (technical) over soft skills (human), which is opposite to the concept of PBL.

The approach that has been adopted to address the project management in a PBL environment is what corresponds to the concept of "agile management". According to Hall (Hall, 2012), this concept is a significant new methodology:

Agile project planning is especially useful for nondeterministic projects, i.e. those where the final configuration of the product or service being developed is not known at the start of the execution stage and only reveals itself as a result of subsequent developments. Examples of nondeterministic projects include research and development, software development, and pharmaceutical drug development.

There is a parallelism between the idea of non determined projects and the projects proposed by the PBL. In the last one, the final configuration of the product is not known, only manifests itself as a result of the activity of the students. Thus, agile planning is especially useful for PBL.

Agile processes meet the challenges of an unpredictable world, relying on the people and their creativity (Dyba, 2000; Nerur et al, 2005). Thus, this kind of management is based on leadership and cooperation; it is also based on a tacit knowledge management and informal communication within an organic structure (flexible and participative encouraging cooperative social action). By contrast, traditional methods respond to a style based on the control, and they are focused on explicit knowledge management, in a formal communication, and in a bureaucratic structure (Dyba and Dingsøyr, 2008). According to Sutherland (Sutherland, 2012), these traditional methods involve a great number of drawbacks, such as being rigid processes, resistant to the change, and in which the resulting products can not express the creativity, skills and passion of their creators. At the same time, they require, for example, that all the good ideas occur at the beginning of the cycle, when in fact, good ideas can appear at any time.

The agile development have several versions, such as: Crystal methodologies (Cockburn, 2004), Dynamic software development method (Stapleton, 2003), Feature-driven development (Palmer et al, 2002), Lean Software Development (Poppendieck and Poppendieck, 2003), Extreme programming (Beck, 2000) (Beck, 2004), and Scrum (Schwaber et al, 2001).

2.1 Overview of Scrum

Scrum is the most generalized agile development method. It is focused on project management for those situations where it is difficult to plan, and it is focused on empirical control mechanisms, where feedback loops constitute the core element. The product is developed by a self-organized team, by increments, called sprints, starting with a planning and finishing with a presentation (Schwaber et al, 2001).

Summarizing the process (figure 1), the first thing to do is to define the project requirements, functionalities or parts (usually named stories), and to assign a time required to solve each one. Once the stories are defined, they must be arranged in order of importance and preference; this is the product backlog (the set of requirements that define the product). Then, the team makes a meeting, which is called sprint planning, to plan and to prioritize the stories. The sprints are time periods of a fixed length (between one and four weeks), where the team must do the assigned stories in the sprint blacklog. Finally, the sprint review and sprint retrospective have to be done. The sprint review is the event in which the results of the sprint are showed. The sprint retrospective is the meeting made at the end of each sprint in order to evaluate it.

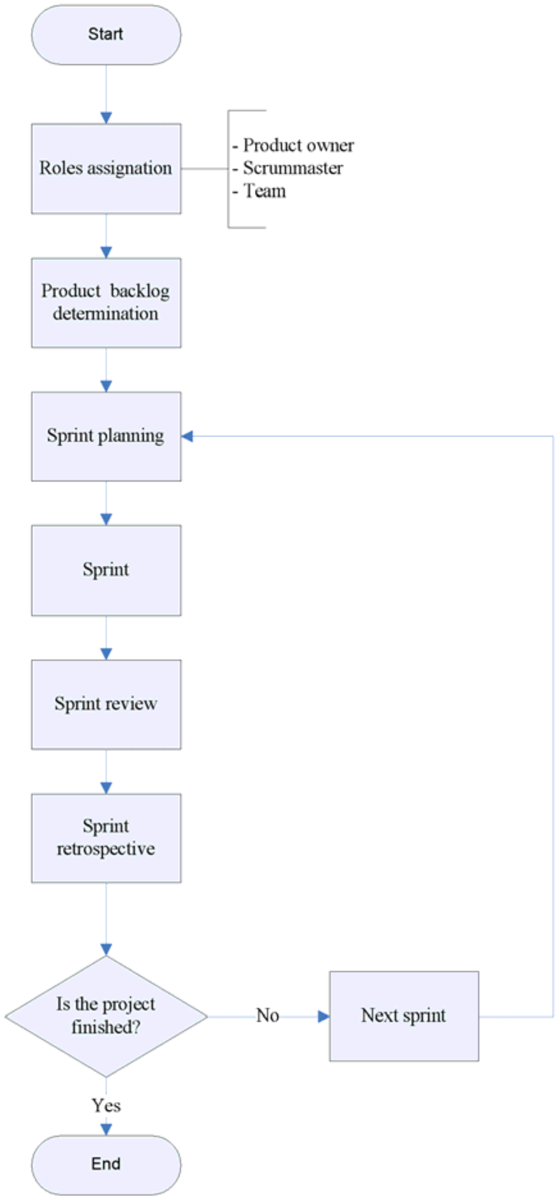


Figure 1. Scrum scheme

Schematically, the general mechanism is as follows:

- Establish a self-organized team.
- Define the functionalities (stories) of the product.
- Split development into smaller parts with a very specific objective that must be accomplished in a short period of time (sprint). This objective is the resolution of the defined functionalities of the product.

The product progresses in a series of sprints lasting several days or weeks.

To implement this mechanism there are three main roles: the product owner, the scrummaster and the team (Benefield et al, 2009). The product owner represents the customer, the person who commissioned the project. The scrummater is the team member responsible of the compliance of the rules of Scrum. Essentially, it is the facilitator of the work of the team. The team is a small number of people who develops the project.

There are also four types of meetings: sprint planning, sprint review, sprint retrospective and daily scrum. The last type is a daily meeting in which each team member answers three questions: What did you do yesterday? What are you going to do today? What problems prevent you from reaching your goals?

The product backlog is the list of the stories that must be solved during the entire project. It is a document (a table) that contains descriptions of all the generic requirements or functionalities that the product must have. It is an open document and it can be changed. It contains estimations of the required effort (Kniberg, 2007). This work effort is expressed in points.

The sprint backlog is the list of the stories of the product backlog that will be resolved in a sprint. So, it is also a table that contains effort estimations associated to each story and it is also expressed in points of work.

The burndown chart is the chart that shows the evolution of the sprint. The horizontal axis marks the time scale, usually in days, and in the vertical axis there are the points of work. Every day, the work points that have been done must be marked.

The Scrum is a framework thought to be applied to software development. The idea behind this paper is that the Scrum can also be applied in any field. The key to use Scrum on any undetermined project is only the ability to translate it in terms of stories, i.e. parts of the project that may have a certain identity. The method is flexible enough to define a story according to the needs of the project team, and it can be redefined or reconsidered between sprints.

Scrum provides structured information presented as tables, such as the product backlog and the sprint backlog, but also as graphics, as the burndown chart. At the same time, it imposes a daily activity of updating data, the daily scrum. In the case of its application on an activity of PBL, this data can be used by participants (students and teachers) to monitor its evolution. However, there is some information that Scrum does not offer and which could be interesting for the evaluation of a project:

- Firstly, it may be interesting to note two parallel processes. On one hand, dynamic mapping between the specifications and the design solutions (the graphical representation of the process). On the other hand, the generation of unknown objects, whose existence can be guaranteed by the knowledge that can be discovered during the process.
- Secondly, the participation of each member of the team must during the process be known.

The first point can be addressed from design theory. The second point will be solved using a file hosting system.

2.2 Overview of C-K Theory and C-K diagrams

According to Thompson and Paredis (Thompson and Paredis, 2010), a design theory is a model of the act of design that allows the interpretation of the actions of design from a theoretical point of view. The act of design is a particular action of information processing performed by a designer. Thus, this is the perspective that must be considered in order to describe the evolution of the content in a PBL.

During the last decades, several theories have been proposed. Those listed below are the most representative (Hatchuel, et al, 2011): General Design Theory (Takeda, et al, 1990), Axiomatic Design (Suh, 1990; Suh, 1999a; Suh, 1999b), Infused Design (Shai and Reich, 2004a; Shai and Reich, 2004b; Shai, et al, 2009), Coupled Design Process (Reich, 1995; Braha and Reich, 2003) and Concept-Knowledge theory (Hatchuel and Weil, 2003; Hatchuel and Weil, 2009).

The Concept-Knowledge Theory (C-K Theory) is the most recent theory, is the most general one and it allows the graphical representation of a design process.

This theory is proposed by Hatchuel and Weil (Hatchuel and Weil, 2003; Hatchuel and Weil, 2007; Hatchuel and Weil, 2009; Le Masson et al, 2010), and it is based on the distinction and interaction between two spaces (figure 1).

In summarized form:

- The Knowledge space is the knowledge available for a designer (or group of designers) at a given time. It is composed by propositions whose logical status is known by the designer.
- The Concept space contains undecidable propositions, propositions whose logical status is unknown and cannot be determined respect to a given knowledge space. These propositions, called concepts, cannot be stated as either true or false by the designer.

- A designer can develop the initial concept by adding new properties in C using his available knowledge. This development is called partitioning.

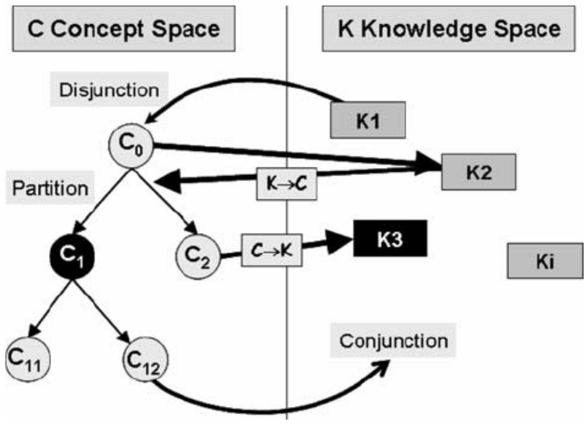


Figure 1. C-K diagram (Hatchuel et al, 2011)

- When a concept space is elaborated, a designer might use his or her knowledge either for a further partition of the concepts, or for attempting a validation of a given concept.
- The result of a K-validation is positive when the designer knows that the concept is true. The result is negative when the knowledge available for the designer allows him to state that such an object cannot be built. In both cases, the conception ends when the concept has been validated (or rejected) (Kazakçi, 2013). The process can also continue creating new concepts.

According to Agogué (Agogué et al, 2014), C-K Theory can graphically show a design activity by C-K diagrams. These diagrams should include both spaces, show its gradual expansion and the use of the four operators.

A C-K diagram represents a snapshot of a design situation, i.e., the state, at a time t, of the activated knowledge and of the several attributes expressed in the process. It allows the representation of all the different explored ways, at the same time, by using different partitions. These diagrams allow to represent the design process and they serve as a support for its discussion. They are also useful for the designer, as noted by the various attributes that define the object, and therefore they can explain the choices and alternatives taken. Finally, these schemes allow to the design teams to explore and to coordinate their activities more easily.

The C-K diagrams are particularly important to visualize the processes carried out by students. In PBL, it can be a tool that forces students to make self-explanation of their processes, thereby reinforcing their learning.

2.3 File hosting service

In Scrum, a project is conducted by a team. Whether the members of this team share the workspace, as if they work from different locations, the method should allow to share all the documents (files) for anyone at anytime. For this reason, it is necessary to have a file hosting service.

In the case of this study, the file hosting service used is Dropbox. The system allows to store and to synchronize files online and to share them with others. There is a feature that allows to know the history of these files. This tool is a record of all the events that have been done in the system. So, if Dropbox is used as file hosting service relating to a project, it allows to obtain useful data for monitoring the process. Thus, it provides relevant information to any file: name, extension, date and time of the change, and the person who has changed it.

This information can be displayed using a table in which the number and the type of actions made in DropBox, for each member and for the whole team, is indicated. The possible actions are: add, modify, delete, rename, move, edit and restore files. A table showing the grouping of activities “add” and “modify” files, compared to “delete” files can also be included. The first two actions (add and modify) are related to the generation of the content in the project, and the delete action is related to the elimination of content. The remaining actions are not involved, just functional. The difference between these kinds of actions indicates the number of actions that are effective at work. At the same time, this can be graphically represented. For example, it is easy to show the dates in which a team does some actions, or even the evolution of the number of files that have been added or changed over time, compared with the number of deleted files.

By using the recorded activities in Dropbox, many other data related to different aspects of the work could be shown. For example, the following variables can be displayed:

- Date corresponding to each action type recorded (group and individual).
- Hour for each action type recorded (group and individual).
- Types of files (text, spreadsheet, drawing file) for each action (group and individual).
- Date for each file type (group and individual).
- Hour for each file type (group and individual).

The data presented in this paper is considered sufficient to show the feasibility of the method for monitoring projects.

3 Method for scheduling and monitoring PBL projects

The proposed method is based on the extensive existing literature on Scrum and C-K Theory. Broadly, the proposal is to use the Scrum framework to manage a project and at the same time, to describe, by C-K diagrams, the state of the configuration of the object being designed.

As the project evolves, any development, in accordance with the tools of the method, must be written down. As a result, the method gives a set of information (the product backlog, the sprint backlog, the burdown chart, and the event log file hosting system) that is a snapshot of the state of a project at any time. Moreover, there is a C-K diagram that evolves every day according to the different solutions proposed to solve a design. It means that there is the tool that allows monitoring and controlling the project.

This method facilitates the acquisition of information as it is inherent at its operation. If Scrum is used, then the product backlog is needed and the daily scrum has to be done, as well as other aspects that form it. If the file hosting system is used, it automatically generates a log. So, monitoring occurs almost automatically, facilitated by the evolution of the process. The product owner, the scrummaster, and also the team can access to updated information corresponding to the parameters of the project. This study proposes a way to present this information using "monitoring cards".

The monitoring cards contain information that allow to view the status of a project at a given time. Thus, the tables and graphs obtained in the proposed process of the method (the product backlog, the sprint backlog, the burdown chart), the tables corresponding to the activity generated in Dropbox, and the C-K diagrams, constitute the information necessary to monitor the project. All this information can be concentrated displayed in a monitoring card. This card can be updated daily and it shows a quick and simplified overview of the state of development of the project.

The aim pursued with this methodology is neither the result that may be obtained by its application, nor the quality of the decisions that have been taken, but the potential for monitoring offered by the procedure itself. The success of a team is determined by multiple interactions between complex factors that are difficult to evaluate (Rodríguez Montequín et al, 2013).

3.1 Method description

The method (figure 3) begins with a non determined project. It success when there is the necessity to develop a design project. The other essential element is the definition of the team responsible of the development. It is recommended that this team is composed of a set of 3 to 5 students.

Then, there is the identification and assignation of the roles, and also the verification that all the participants have a definite function. The professor can be the product owner. Following, the first task of the scrumaster is to prepare the file hosting system.

At this point, the working group (the product owner, scrummaster and the team) determine the product backlog and make the first sprint planning. Afterwards, the team is ready to start the first sprint. Conducting the sprint involves the execution of the design activity itself. This means that it is in the sprint where the application of C-K Theory is done. In the product backlog, the functional requirements that the project has to solve have been exposed, and it is the starting point of the design process. The process will start by defining a concept that will be constituted by a set of initially limited attributes that are propositions in K space. These attributes arise from the functional requirements set out in the product backlog. The stories that make up the sprint evolve according to the theory and according the way how the team is able to apply it.

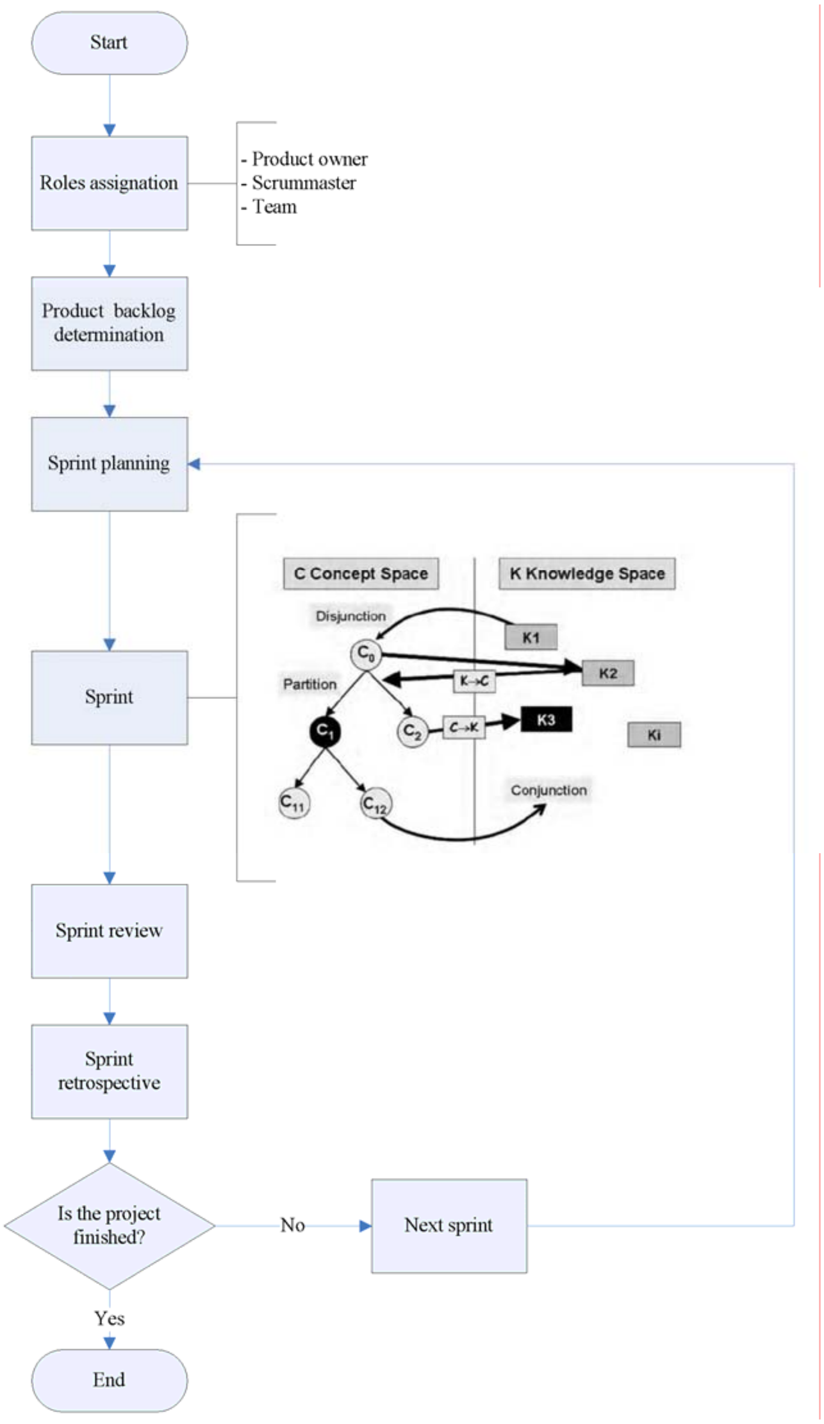


Figure 3. Method scheme

The progress of the project is done by successive stages called sprints. Every sprint corresponds to a new phase of the design process and therefore it is the framework for the use of C-K Theory. The process continues as follows: the presentation of the first sprint results (sprint review) and the first sprint evaluation (sprint retrospective). From the point of view of the design process, the sprint review and sprint retrospective are only functional tasks and they are not involved in the configuration of the object.

Once the results are showed and the first sprint is evaluated, the start of the second sprint should be done. That means to proceed in the same way: to make the second sprint planning, to conduct the second sprint (application of C-K Theory), to present the results of the second sprint (sprint review) and to evaluate the second sprint (sprint retrospective).

The process can continue with a third sprint, if the project has not been finalized. It is developed in the same way as above. The process continues until as many as sprints are necessary for obtaining an end result of the project. The result should be a concept with enough attributes such as to demonstrate its viability in K space.

Every day, the team and the scrummaster must make a planning meeting, the daily Scrum. In this meeting each team member must answer three questions: What did you do yesterday? What are you planning to do today? And, have you any problem that has prevented you to achieve your goal? At this meeting, the tools for monitoring the project must be updated. First, solved points must be written down in the burndown chart. Secondly, the records in the file hosting service must be compared with the answers of the team members. At the same time, the actions done in the system must be written down. Finally, the C-K diagram must be reviewed and updated.

All these indicators for monitoring and control are added and updated at the monitoring card of the project.

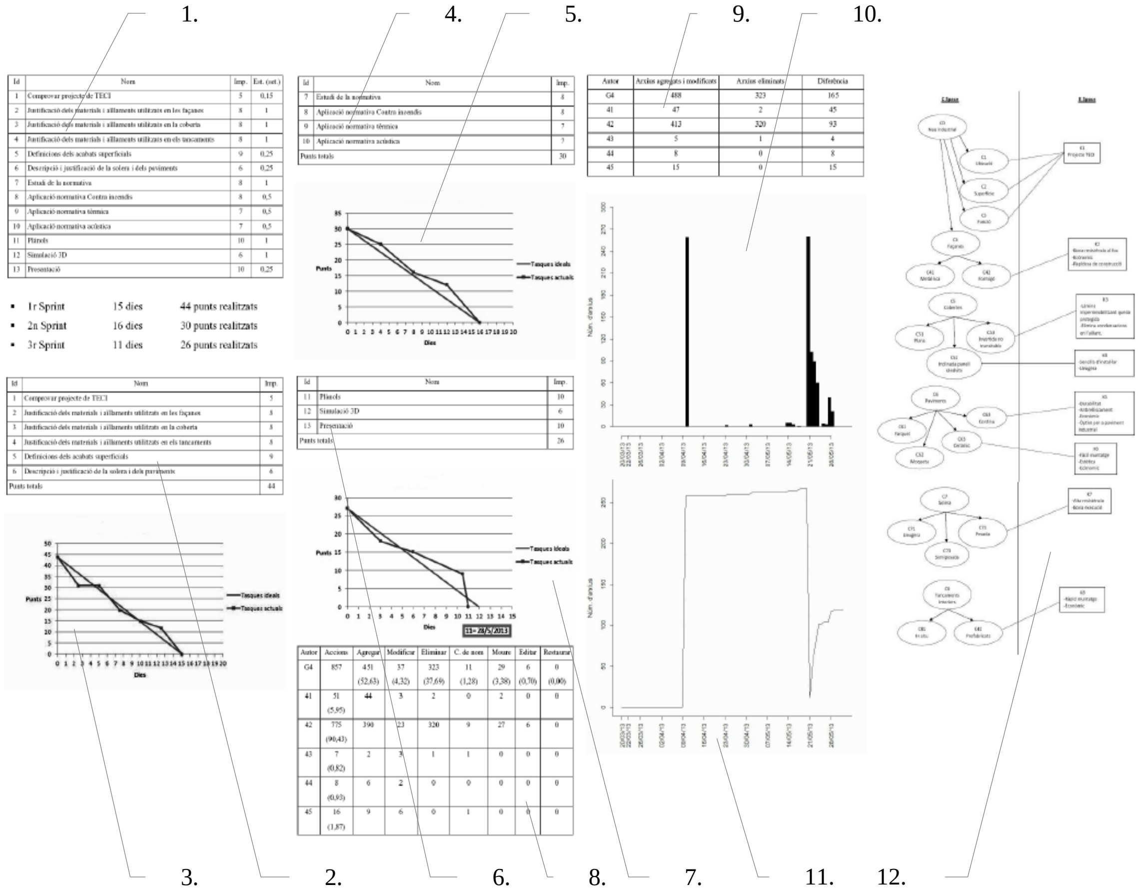
3.2 Monitoring card model

The design of the proposed card corresponds to an elongated paper sheet of sufficient size to allow reading (figure 4). At the top of the file, the date on which the project has begun, the date of the last update of the card, the name of the product owner, the scrummaster and all members team, the project name and other information that may be considered of importance (name of developer, description, contact numbers, ...) must be written down.

The rest of the card is divided into four columns. The elements of monitoring are put, from left to right and from top to bottom, in the following order:

- Product Backlog.
- Sprint dates.
- Sprint Backlogs.

- Burdown charts.
- Activity in Dropbox tables and graphs.
- C-K Diagrams.



1. Product backlog	5. Second sprint burndown chart	9. Content files table
2. First sprint backlog	6. Third sprint backlog	10. Type of actions graphic
3. First sprint burndown chart	7. Third sprint burndown chart	11. Content files graphic
4.Second sprint backlog	8. Type of actions table	12. C-K Diagram

Figure 4. Monitoring card

4. Conclusions and future works

In this paper, two research areas have been joined: project management and engineering design. These areas, although related, do not usually show up together. A method to monitor projects has been proposed. The monitoring elements are generated from the working method itself. This method facilitates the set of data necessary to monitor and control a project. A monitoring card has been designed, which allows a quickly and easily view of the status of a project. This information can be daily updated.

The next step is to implement the method in a PBL engineering course. The aim of this experiment will be to prove the validity of the method. Some questions could be: is the method really simple? Is it really agile? Can students learn it in a short period of time?

Finally, the monitoring method can be extended in order to monitor other variables that have not been controlled in this study. For example, the evolution of the costs or an indicator of the speed at which the different tasks have been done.

References

1. Agogué, M., Cassotti, M. 2013. "Understanding fixation effects in creativity: a design-theory approach". In *Workshop Design Theory Design Society Special Interest Group*, Paris.
2. Agogué, M., Hooge, S., Arnoux, F., and Brown, I. 2014. *An Introduction to Innovative Design Elements and Applications of C-K Theory*. Paris: Press des Mines, Transvalor.
3. Ainsworth, S., and Loizou, A.T. 2003. "The effects of self-explaining when learning with text or diagrams". *Cognitive Science* 27: 669-681.
4. Atkinson, J. 2001. *Developing teams through project-based learning*. Hampshire: Gower Publishing.
5. Barron, B. 1998. "Doing with understanding: Lessons from research on problem- and project-based learning". *Journal of the Learning Sciences* 7(3, 4): 271-311.
6. Beck, K. 2000. *Extreme Programming Explained: Embrace Change*. Boston (MA): Addison
7. Beck, K. 2004. *Extreme Programming Explained: Embrace Change*. 2nd ed. Boston (MA): Addison

1. Benefield, G., Deemer, P., Larman, C., and Vodde, B. 2009. *The Scrum Primer*. Scrum Training Institute. http://scrumfoundation.com/library

8. Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., and Palincsar, A. 1991. "Motivating project-based learning: Sustaining the doing, supporting the learning". *Educational Psychologist* 26: 369-398.
9. Braha, D., and Reich, Y. 2003. "Topologial structures for modelling engineering design processes". *Research in Engineering Design* 14(4): 185-199.
10. Cockburn, A. 2004. *Crystal Clear: A Human-Powered Methodology for Small Teams*. Boston (MA): Addison-Wesley.
11. Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P. and Glaser, R 1989 "Self-explanations: How students study and use examples in learning to solve problems." *Cognitive Science* 13: 145-182.
12. Chinnowsky, P., Brown, H., Szajnman, A., and Realph, A. 2006. "Developing knowledge landscapes through project-based learning". *Journal of Professional Issues in Engineering Education and Practice* 132(2): 118-125.

13. Chu, S.K.W., Tavares, N.J., Chu, D., Ho, S.Y., Chow, K., Siu, F.L.C., and Wong, M. 2012. *Developing upper primary students' 21st century skills: inquiry learning through collaborative teaching and Web 2.0 technology*. Hong Kong: Centre for Information Technology in Education, Faculty of Education, University of Hong Kong. http://hdl.handle.net/10722/161055.
14. De los Ríos, I., Cazorla, A., Díaz-Puentea, J. M. and Yagüea, J. L. 2010. "Project–based learning in engineering higher education: two decades of teaching competences in real environments" *Procedia-Social and Behavioral Sciences* 2(2): 1368-1378
15. Doppelt, Y. 2003. "Implementation and assessment of project-based learning in a flexible environment". *International Journal of Technology and Design Education* 13(3): 255-272.
16. Dunlap, J.C. 2005. "Workload reduction in online courses: Getting some shuteye". *Performance and Improvement* 44(5): 18-25.
17. Dyba, T., and Dingsøyr, T. 2008. "Empirical studies of agile software development: A systematic review". *Information and Software Technology* 50: 833-859.
18. Dyba, T. 2000. "Improvisation in small software organizations". *IEEE Software* 17(5): 82-87.
19. Dym, C.L., Agogino, A.L., Eris, O., Frey, D.D., and Leifer, L.J. 2005. "Engineering Design Thinking, Teaching, and Learning". *Journal of Engineering Education* 94(1): 103-120.
20. Ertmer, P.A., Richardson, J.C., Belland, B., Camin, D., Connolly, P., Coulthard, G., Lei, K., and Mong, C. 2010. *Impact and Perceived Value of Peer Feedback in Online Learning Environments*. http://www.edci.purdue.edu/ertmer/docs/AECT05_Proc.pdf .
21. Felder, R.M., Forrest, K.D., Baker-Ward, L., Dietz, E.J., and Mohr, P.H. 1993. "A longitudinal study of engineering student performance and Retention. I. Success and failure in the introductory course". *Journal of Engineering Education* 82(1): 15-21.
22. Filomeno Coelho, R., Tysmans, T., and Verwimp, E. 2014. "Form finding & structural optimization, a project-based course for graduate students in civil and architectural engineering". *Structural and Multidisciplinary Optimization* 49: 1037-1046.
23. Frank, M., Lavy, I., and Elata, D. 2003. "Implementing the project-based learning approach in an academic engineering course". *International Journal of Technology and Design Education* 13(3): 273-288.
24. Hall, N. 2012. "Project management: Recent developments and research opportunities". *Journal of Systems Science and Systems Engineering* 21(2): 129-143.
25. Hatchuel, A., and Weil B. 2003. "A new approach of innovative design: an introduction to C–K design theory". In *Proceedings of ICED 03, the 14th International Conference on Engineering Design*, edited by A. Folkeson, K. Gralen, M. Norell, and U. Sellgren. Design Society.
26. Hatchuel, A., and Weil B. 2007. "Design as forcing: deepening the foundations of C-K Theory". In *Proceedings of ICED 2007, the 16th International Conference on Engineering Design*, edited by J.-C. Bocquet. Design Society.

27. Hatchuel, A., and Weil B. 2009. “C-K design theory: an advanced formulation”. *Research in Engineering Design* 19(4): 181-192.
28. Hatchuel, A., Le Masson, P., Reich, Y., and Weil B. 2011. “A systematic approach of design theories Using generativeness and robustness”. In *Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 7: Human Behaviour in Design*, edited by S.J. Culley, B.J. Hicks,T.C. McAloone, T.J. Howard and P. Badke-Schaub. Design Society.
29. Hérold, J.-F., and Ginestié, J. 2011. “Help with solving technological problems in project activities”. *International Journal of Technology and Design Education* 21(1): 55-70.
30. Johnson, P.A. 1999. “Project-based, cooperative learning in the engineering classroom”. *Journal of Professional Issues in Engineering Education and Practice* 125 (1): 8-11.
31. Karaman, S., and Celik, S. 2008. “An exploratory study on the perspectives of prospective computer teachers following project-based learning”. *International Journal of Technology and Design Education* 18(2): 203-215.
32. Karau, S. J., and Williams, K. D. 1993. “Social loafing: A meta-analytic review and theoretical integration”. *Journal of Personality and Social Psychology* 65(4): 681-706.
33. Kazakçı, A. 2013. “On the imaginative constructivist nature of design: a theoretical approach”. *Research in Engineering Design* 24 (2): 97-103.
34. Kerr, N. L., and Bruun, S. E. 1983. “Dispensability of member effort and group motivation losses: Free rider effects”. *Journal of Personality and Social Psychology* 44: 78-94.
35. Kniberg, H. 2007. *Scrum & XP from the trenches: How we do Scrum*. Toronto: C4Media Inc.
36. Lee, H.J., and Lim, C. 2012. “Peer Evaluation in Blended Team Project-Based Learning: What Do Students Find Important?”. *Educational Technology & Society* 15(4): 214-224.
37. Le Masson, P., Weil, B., and Hatchuel, A. 2010. *Strategic Management of Design and Innovation*. Cambridge: Cambridge University Press.
38. Marcelino-Sádaba, S., Pérez-Ezcurdia, A., Echeverría Lazcano, A.M. and Villanueva, P. 2014. “Project risk management methodology for small firms”. *International Journal of Project Management* 32: 327-340.
39. Milentijevic, I., Ciric, V., and Vojinovic, O. 2008. “Version control in project-based learning”. *Computers & Education* 50: 1331-1338.
40. Moursund, D. G. 2003. *Project-based learning using information technology*. Eugene (OR): International Society for Technology in Education.
41. Mozas-Calvache1, A.T.and Barba-Colmenero, F. 2013. “System for Evaluating Groups When Applying Project-Based Learning to Surveying Engineering Education”. *Journal of professional issues in engineering education & practice* 139: 317-324.
42. Nerur, S., Mahapatra, R., Mangalaraj, G. 2005. “Challenges of migrating to agile methodologies”. *Communications of the ACM* (May) (2005): 72-78.
43. Palmer S.R., and Felsing, J.M. 2002. *A Practical Guide to Feature-driven Development*. Upper Saddle River (NJ): Prentice Hall.

44. Pant, I. and Baroudi, B. 2008. "Project management education: The human skills imperative". *International Journal of Project Management* 26: 124-128.

45. Poppendieck, M., and Poppendieck, T. 2003. *Lean Software Development: An Agile Toolkit for Software Development Managers*. Boston (MA): Addison-Wesley.

46. Reich, Y. 1995. "A Critical Review of General Design Theory". *Research in Engineering Design* 7: 1-18.

47. Rodríguez Montequín, V., Mesa Fernández, J. M., Villanueva Balsera, J. and García Nieto, A. 2013. "Using MBTI for the success assessment of engineering teams in project-based learning". International Journal of Technology and Design Education 23: 1127-1146.

48. Salomon, G., and Globerson, T. 1987. "When teams do not function they way they ought to". *International Journal of Educational Research* 13: 89-100.

49. Schwaber K., and Beedle, M. 2001. *Agile Software Development with Scrum*. Upper Saddle River (NJ): Prentice Hall.

50. Shai, O. and Reich, Y. 2004a. "Infused Design: I Theory". *Research in Engineering Design* 15(2): 93-107.

51. Shai, O. and Reich, Y. 2004b. "Infused Design: II Practice". *Research in Engineering Design* 15(2): 108-121.

52. Shai, O., Reich, Y. and Rubin, D. 2009. "Creative Conceptual Design: Extending the Scope by Infused Design". *Computer-Aided Design* 41(3): 117-135.

53. Stapleton, J. 2003. *DSDM: Business Focused Development*. 2nd ed. Boston: Addison-Wesley.

54. Suh, N.P. 1990. *Principles of Design*. New York: Oxford University Press.

55. Suh, N.P. 1999a. "Applications of Axiomatic Design". In *Proceedings of the 1999 CIRP International Design Seminar*, edited by H. Kals and F. van Houten, 1-46.

56. Suh, N.P. 1999b. "A Theory of Complexity, Periodicity and the Design Axioms". *Research in Engineering Design* 11: 116-131.

57. Sutherland, J. 2012. *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework*. Cambridge: Scrum, Foundation. http://scrumfoundation.com/library.

58. Takeda, H., Veerkamp, P., Tomiyama, T., and Yoshikawa, H. 1999. "Modeling Design Processes". *AI Magazine* 20(4): 37-48.

59. Thomas, J. 2000. *A review of research on project-based learning*. The Autodesk Foundation. http://www.bie.org/research/study/review_of_project_based_learning_2000.

60. Thompson S., and Paredis, C. 2010. "An Introduction to Rational Design Theory". In *ASME Proceedings 22nd International Conference on Design Theory and Methodology; Special Conference on Mechanical Vibration and Noise*, 59-72.