Adding Control to Open Innovation Projects Through Agile Practices

Terhi Kilamo[1], Ville Kairamo[2], Petri Räsänen[2], and Jukka P. Saarinen[3]

[1] Tampere University of Technology firstname.lastname@tut.fi
[2] Uusi Tehdas/New Factory firstname.lastname@hermia.fi
[3] Nokia Research Center jukka.p.saarinen@nokia.com

Abstract. Businesses today have to rely on rapid development and release cycles. Thus open innovation has emerged as an increasingly appealing option also for the software business to gain variant ideas and concepts. A local open innovation platform for students, Demola, allows university students to work on real life industrial cases of their own interest. We monitored the daily work routine of a student team and found that practises from agile software development were applied to scope and manage the project activities.

1 Introduction

Many companies rely on innovation on a daily basis to create better products and to improve their internal processes [2] Constant, lightning-fast innovation is becoming an essential element of product development also in the software business. Open innovation helps in identifying the best ideas by combining internal and external ideas [7, 2].

Iterative or agile software development [6] has become more popular over the more traditional processes in the software industry. Agile development practises, mainly the concept of sprinting, has been studied earlier in the context of free software [1]. In this paper we focus on the agile approach in the open innovation setting. Innovation work is similarly characterized by ideas, changes and do as you go attitude. The absence of formal processes and excess documentation is characteristic to it in accordance to the manifesto for agile software development [5] with emphasis on interaction, collaboration and change when necessary.

The paper focuses on open innovation in the context of academia-industry cooperation in the form of a local open innovation platform Demola [8]. One of the aims of the platform is to develop an open innovation environment that is multidisciplinary and agile in the sense that innovations can flow freely and are not restricted to any artificial process or framework that must be obeyed in order to benefit from it. In this paper we discuss how practices of agile software development can be incorporated to compensate such innovation challenges as timely delivery, communication, and quality. We have conducted our study by interviewing the key people behind the environment and by observing an

example development team for identifying their working practices. The main research question was: What kind of development practises are used to work on the projects and how do they compare to agile practises?

The rest of the paper is structured as follows. Section 2 motivates the work by introducing the open innovation platform, Demola. Section 3 discusses the practices agile development in the open innovation context and further highlights them in practise through an example team. The results of the paper are discussed in Section 4 and finally Section 5 concludes the paper with some final remarks.

2 Platform for Open Innovation and Learning

There is a real need for increased opportunities for innovation projects that can lead to new business ideas. Open innovation environments allow businesses to reach beyond the company scope in the search for new concepts and ideas. A local open innovation platform, Demola, provides a governance framework needed with practices and working principles to bring innovation partners together and to ensure ongoing innovation work.

2.1 Demola Organisation

Demola is a modern learning environment for students from different universities. It aims to multidiciplinary and agile development of innovative products and demos. The project ideas come from the industry and public organisations and thus concepts that have practical business importance are developed. The student work is supported by both the industrial and the academia partners, who provide guidance throughout the project. Figure 1 shows the partners in Demola and the flow of communication and support for the project work.

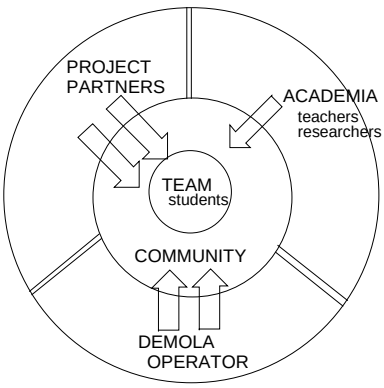

Fig. 1. Demola Partners

Demola offers a governance framework that facilitates team building and supports emerging business ideas. It also incorporates a model for managing immaterial rights that supports startups and respects the authors. On a practical level, Demola provides workspaces that support team work and co-creation.

3 Incorporating Agile Development Practices

Projects come to Demola through industrial project partners and therefore they have an intended outcome, no matter how loosely defined, and a fixed timeframe. The teams are also rather small in size and new members are normally not added after project kickoff. Iterative or agile software development that has gained popularity in the software industry over the more traditional processes lends a way to handle the innovation projects. There are several agile approaches available, such as Scrum [6, 3] or Extreme Programming [6, 4]. We focus here on practices general to the idea of agile and not to any specific approach.

In [6] the authors find that software development can be said to be agile when the releases are small but done often, the customer and the developers work together and in close communication, the development method is straightforward and adapts to the situation making it easier to do rapid changes. These are all identifiable in Demola as the independent teams appear to control their development cycle through applying practices known from agile development. This also in part aids Demola in being a sustainable open innovation community through completed projects.

Rapid Release Cycle Innovative development starts from ideas and concepts. An ideal project timeframe is short in Demola. A typical project is three to four months in duration. Development is done in small increments, the final outcome is loosely specified and the teams have a lot of fluidity in the specification. The current state of the project is demoed regularly to the customer. An agile, demo-driven development approach with frequent demos enables control of the project focus and its intended outcome.

Close Communication The teams commonly meet with eachother and the customer on a regular basis during the lifecycle of their Demola project. It is typical that teams keep in touch regularly, mostly daily, to sync their work progress via chats, online phone applications or meetings. Even though there are no product releases during the life cycle of the project the customer gets the current version of the product in these meetings. Changes can be made to the requirements and project outcome based on the teams work. While the requirements management is flexible with requirements changed and added as the project evolves, the project runs for a predetermined time. Similar fixed time projects are known from agile software development and give the project customer control over the end product. They can add, remove and prioritize the requirements as they go thus controlling the outcome of the project.

Self-Managing Teams The teams themselves can be seen through agile practises, where development is built around small development teams or pairs. One Demola project team forms such a unit and has freedom in choosing and adapting the working methods and arrangements as they see fit. There is likely to be a wide variation of practices here as the teams and projects vary. What is

common to them is the Demola workplace that provides premises and tools to enable independent, collaborative work of the teams as they best see fit.

3.1 Sample Case: Team Practices

We monitored the work of one team through the course of their development project to see how our observations on Demola apply to the daily work routine. The team was selected as it had a suitable kickoff date and project schedule and both the team and their customer project partner had no objections on us observing their collaboration. The development practices relied on iterative development with one week intervals.

There were five members in the sample case team. The educational background of the participants varied with one of the team members having completed their master's degree. One had a bachelor of science degree while three were still working on completing their undergraduate studies. The cultural background was diverse with members of four different nationalities and from two different continents. However, all participants in the project were software engineering majors even though innovation projects would benefit from a wider view with participants majoring in usability, human sciences or graphics design to name a few.

3.2 Team Collaboration

The team collaboration was informal but had certain structuring elements in it. The overall format of communication and syncing followed agile practices. There were more quiet and more talkative people in the team but no sense on dictatorship emerged. The eldest team member could be seen as a team leader and was also voted into that position by the team. They found a named leader necessary to keep the work in sync and for managing the work load.

At the beginning of the project the team decided to keep in touch daily to sync what everyone has done. As the development started in earnest the team abandonded such a strict, approach and adapted to a more flexible once a week sync. A daily sync would have followed agile methods better, but the teams self direction abandoned the approach.

3.3 Team and Customer Interaction

The team met the customer project partners weekly in a meeting at the project partners offices. A demo was prepared to show the progress that week. The length was roughly one hour, never over two. The meetings were informal but followed a certain structure that resembled a process known from agile methods. What progress the team had done during the week was discussed over a demo and what needed to get done in the future was agreed upon based on that. The possible problems, or impediments, that stood in the way of the team were covered together with if the customer could help the team in solving them was

also covered. A checklist of the project's status was maintained not only to keep track of the project but also to map the changing and emerging requirements. Both the customer and the team were able to make changes to the requirements but the customer had the final say. The customer acted as a product owner in agile.

The team members and a person responsible for the project on the customer's side were always present at the meetings. In addition, people from the customer company interested in the project attended when necessary. These outside experts were also brought in to aid in a development issue or give insight on technical topics.

4 Discussion

Demola is at heart a community. Additionally its day to day practises lean towards agile methods for managing the project as community driven development approach alone does not provide sufficient tools for timeboxing or requirements management. This brings a natural addition to the innovation work without endangering the community level principles.

The participants have the final responsibility of the work and project outcome. The teams keep in close communication not only with eachother but also with the project partner. Furthermore, frequent demos add flexibility to the requirements. Based on the overall Demola approach and the work of the sample case team, the agile approach appears as a viable way for the teams to keep the project on track and to adjust it to the needs of the project partner during the project. Ability to meet the project requirements and create innovative products and demos within Demola is an important factor in Demola's sustainability as Demola is dependent on industrial partner's project ideas.

The biggest limitation of our research on Demola so far is that our observation is limited to one example team. There is a risk that we get an overly idealistic and onesided view of the teams based on just one project. We believe the results are applicable to other teams as well since the agile practices were identified from the Demola community as well before monitoring a project team. We intend to enforce the work through a wider study of the Demola projects. Identifying further how variance in team member's background and multidiciplinary teams effects the project work is also of interest.

5 Conclusions

We set out to study an open innovation approach and how the daily workflow of the project development in a single team shares similarities with agile software development to control the innovative work flow. Our findings suggest challenges of community work can be addressed by the adoption of such new practises as time-boxing, face to face meetings, and demo-driven development.

References

1. Capiluppi A. and Adams P.J. Bridging the gap between agile and free software. *International Journal of Open Source Software and Processes*, 1(1):58–71, 2009.
2. Chesbrough H. *Open Innovation: Researching a New Paradigm*, chapter Open Innovation: A New Paradigm for Understanding Industrial Innovation. Oxford University Press, 2006.
3. Takeuchi H. and Nonaka I. The New New Product Development Game. *Harvard Business Review*, pages 137–146, January-February 1986.
4. Beck K. Embracing Change With Extreme Programming. *Computer*, 32(10):70–77, October 1999.
5. Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R.C., Mellor S., Schwaber K., Sutherland J., and Thomas D. Manifesto for Agile Software Development. Available at: `http://agilemanifesto.org/`, March 2002. Last visited March 2011.
6. Abrahamsson P., Salo O., Ronkainen J., and Warsta J. Agile Software Development Methods Review and Analysis. VTT Publications 478, 2002.
7. Davis S. How to Make Open Innovation Work in Your Company. *Visions Magazine*, December 2006.
8. Demola Innovation Platform. `http://www.demola.fi`. Last visited March 2011.