Design sprint in learning software development

Jaana Holvikivi

Metropolia UAS, Finland, jaana.holvikivi@gmail.com

Abstract. Agile methods are replacing former, highly systematic project management practices in software development. In this paper, a case of design sprint trial at a university is described, and its benefits and challenges are analysed. The experience is compared to other reported cases and project-based learning practices. Agile methods can be taught in various ways that are compared and evaluated in this paper. Generally, the results of using agile methods as module structure and applying short design sprints have been very promising. The advantages over other methods are connected to hands-on practice, quick implementation and improved student co-operation.

Keywords: agile methods, higher education, collaboration practices, design sprint, software engineering

1 Introduction

The departments of Information Technology and Media Engineering at the Metropolia University of Applied Sciences in Helsinki have applied project-based learning and client-defined assignments in the curriculum over twenty years [1]. The departments have followed a strong tradition of practical projects in engineering education. Nevertheless, the way the projects were introduced and implemented, has evolved significantly over the years. [2,3]

Software developers evidently need to master programming and development technologies. Furthermore, they need to understand the entire software development process, which is a wider and more demanding skill. Software development was initially introduced in the university curriculum as a module called Application development, where students practiced with a school project following a simplified form of software development life-cycle, and implementing project management. However, as school projects were necessarily small compared to industry projects, students were not able to acquire a deep understanding of the complexity of challenges. Mostly they learnt to deal with a small application, creating a working prototype for evaluation. Recently, this approach was replaced by agile development practices, which are increasingly popular, in particular in mobile application and web development industries. The changes that have taken place in software industry are described more closely elsewhere [2,4].

In this paper, an experiment to apply a recent innovation in the software development work is described, and the results of the trial cases are compared with earlier experiences, and experiences from other institutions. The case consists of two implementations of a GV design sprint in two successive years, and its benefits and challenges are analysed. The advantages of a short design sprint over larger learning modules are discussed. Additionally, the experience is compared to other reported cases, which can be found on the Google Ventures website gv.com/sprint [5].

2 Agile method and design sprint

Sprint is a concept in agile software development, referring to a limited interval such as a week or two weeks for developing an application one step further. Agile development aims at getting an early version of a product ready for user evaluation as quickly as possible in order to get constructive feedback [3]. In this way, mistaken assumptions about user needs are caught soon, and proceeding to a wrong direction is avoided. The company Google Ventures has developed a well-documented version of this method called GV Design Sprint that is principally intended to help start-up companies find the right focus for their innovations [5]. Nevertheless, the method is sufficiently general that it can be applied in other contexts as well, such as in colleges and universities to introduce design thinking, user centred development and innovation methods to students.

In this paper, a case of design sprint trial at a university is described, and the experience is compared to other reported cases, which can be found on the GV sprint stories website [6]. Research on this subject in education is still at an emerging stage, even though the method in industry is well-attested and studied. More generally, agile methods have been taught already more than ten years across universities, and a number of positive reports have been published [7-10].

3 Research and methods

The main aim of applying a design sprint to the software development and business skills module at a university of applied sciences was to introduce agile thinking ideas, and on the other hand, let students practice innovation process in a relaxed setting that would foster creativity. All the deliverables of the process were collected, as well as student feedback. This data could be compared to results of earlier modules that were implemented differently. We have a long experience of undergraduate courses in ICT where students have been asked to develop a software product such as a website or a mobile application based on their own product ideas, starting from 1998 [1]. Students have usually been cautious in selecting the idea, because they focused on getting a product prototype implemented, therefore choosing something that is close to their own everyday experience such as personal time-management, finding a restaurant or concert, and alike. The module in this experiment was called Software business start-up, where students were expected to find more innovative ideas for a start-up company that would have chances on the extremely competitive market of mobile applications.

For analysis of the outcomes, we collected student writings, reports, videos and other deliverables, conducted student and teacher interviews, conducted several surveys, and used feedback questionnaires. Data about the modules was also collected through field ethnography and participant observation [11, 12].

The GV design sprint was chosen because it is well-documented, it has a large user base in industry, and it applies the most important ideas of agile development and design thinking. It can be introduced quickly, and its progress is intuitive and systematic. It does not require much earlier knowledge of any of the methods, even though some previous exposure to design thinking and user centred methods is helpful. Currently, most information technology students are aware of agile methods, and as they know that they are important in the industry, they are eager to learn them. The method has a book as well as a website that contains a schedule for the sprint, a list of supplies and equipment that is needed, plus a set of short videos where the five days of the sprint are introduced. Students like the videos in particular, because it is more and more common that they practice new skills by watching videos [5].

We have implemented the design sprint twice in the beginning of the Software business start-up module that lasted full time altogether eight weeks for third-year bachelor students in an international group. The module offered some new technical knowledge, mainly the MEAN method for implementing web applications (Mongo database, Javascript and NodeJS servers), as well as business skills such as accounting, financial statements and start-up business basics. The idea of the module was to practice business skills by creating a mock start-up company and developing a product prototype. During this module, the first week was spent in starting to get familiar to the technology, and beginning to set up the tools for development. The second week was devoted to the design sprint. 45 students were divided into 7 teams by the instructor. The teams were formed in a way that they had a diverse composition of female and male students, and from several nationalities, such as Finnish, Russian, Vietnamese, Nepalese, French and German students. The reason for the team diversity was to expose students to as many different kinds of fellow students as possible in order for them to set up innovative business project teams for the rest of the course. The product ideas for the sprint were intended to be for practice only, but in some cases, students decided to continue with the same idea for their business case.

The instructions for the GV design sprint were adapted to the educational setting, and the modified schedule for four days was given as a guideline to the student teams. Two large classrooms were booked for the whole period of time, and they were supplied with whiteboards, flap-paper boards, and a huge amount of post-it stickers. Students used their own laptops for documenting their progress in a cloud-document that consisted their log. All documentation was shared among all participants and instructors. Students also used their own smartphones to take photos and videos of their prototypes and user testing sessions.

The sprint schedule instructions looked as follows:

On **Monday**, teacher explains the sprint and gives guidelines. You are placed in a team with 5 other students. You'll kick off your sprint by sharing knowledge, understanding the problem, and choosing a target for the week's efforts. Write your checklist on a whiteboard. When you're done with a task, check off the item. One person acts as facilitator who looks after the process. First, you'll try to understand the challenge. Next, you'll ask the experts to share what they know. This is done mainly by searching the net. Then you map the challenge. Finally, you'll pick a target by voting on ideas: an ambitious but manageable piece of the problem that you can solve in one week. The decider has the final say. You are collectively responsible for the team deliverables.

On **Tuesday**, you get to focus on solutions. The day starts with inspiration: a review of existing ideas to remix and improve. After that, each person will sketch, following a four-step process that emphasizes critical thinking over artistry. You'll also begin planning customer tests.

Wednesday: You have now decided which ideas have the best chance of achieving your long-term goal. Then, you'll take the winning scenes from your sketches and weave them into a storyboard: a step-by-step plan for your prototype. Then, you'll adopt a "fake it" philosophy to turn that storyboard into a prototype that can be tested with users.

Thursday: Prototype! Do a trial run. Run through your prototype. Look for mistakes. Finish up the prototype. Write interview script. Conduct the interviews with at least three users and report your findings. Look for patterns. At the end of the day, read the board in silence and write down patterns. Make a list of all the patterns people noticed. Wrap up. Review your long-term goal and your sprint questions. Compare with the patterns you saw in the interviews. Decide how to follow-up after the sprint. Write it down.

4 Results and discussion

As mentioned before, this process was implemented twice in successive years. Because the student groups were international, they were quite heterogeneous. Nevertheless, the results of the experiment were predominantly positive. First of all, all teams completed their sprint successfully on time, and delivering all that was asked, namely a rudimentary prototype, a log and user testing results. No particular conflicts arose within teams, but a couple of students dropped out as they were also working outside school at the same time. Many teams were happy enough with the experience to continue the module with the same team members during the business development process, even though they were allowed to reorganize after the initial sprint. Moreover, all teams continued to apply methods that they had learnt during the sprint, using daily or twice-weekly scrum meetings, developing their ideas on post-it stickers, and discussing in open spirit their innovation development.

Group processes and team work in student groups face several challenges that have been reported previously [13]. However, this particular method addresses some problems implicitly. The dominance of vocal students is mitigated by the demand to give turns in discussion to all team members, as well as the requirement for voting for proposed ideas. That also reduces criticism, when ideas go through a selective process and can be kept for future reference. The results were not marked, therefore students could concentrate on the process and quality of original ideas without worrying about technical issues in the prototype.

Importantly, the process had a strict schedule where results of each day were reported immediately, which prevented procrastination. Students tend to work hard only when the deadline approaches, therefore it helps to give them short deadlines and to split the process into small increments, which exactly is the agile idea. The process also has clearly defined roles that keep all participants busy. Because of the very practical hands-on methods for idea creation, organization and development, the target remains visible. The groups were all week in the same room where the learnt to know each other, and had physical communication. Body language helped in understanding others and their feelings better, in particular, as the different nationalities have different ways of expressing themselves. The physical proximity created a feeling of belonging together, which sometimes is missing if students work much online.

The cases that are described at the sprint stories website are from business studies faculties [5]. Therefore, a direct comparison may not be warranted. However, the reported cases show similar positive effects in the Reykjavik University and also Laurea University of Applied Sciences in Finland.

5 Conclusion

Based on earlier studies and literature reporting teaching of agile courses, the promising outcomes of this study were not surprising. I would suggest this approach to complement project-based curricula at universities, as it neatly adds to methodological selection. Moreover, it could be used also as an introduction to teamwork and project-based learning in more traditional educational settings. The cases that were presented here are from universities with advanced technologies but there is nothing that actually requires advanced technologies, as only paper, a room and pens are needed. Essentially the process enlarges thinking of the participants and opens up ideas for innovation, improving their teamwork skills.

References

1. Holvikivi, J.: Culture and cognition in information technology education, SimLab publications, Dissertation series 5, Espoo, Finland (2009)
2. Holvikivi, J., Hjort, P.: Agile development in software engineering instruction. in A Tatnall & M Webb (eds) Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing. IFIP Advances in Information and Communication Technology. Springer. (2018).
3. Holvikivi J., Lakkala M., Muukkonen, H.: Introducing collaborative practices to undergraduate studies. In: Brinda T., Mavengere N., Haukijärvi I., Lewin C., Passey D. (eds) Stakeholders and Information Technology in Education. IFIP Advances in Information and Communication Technology, vol 493. Springer, Cham (2017)
4. Encyclopedia of Software Engineering - Wiley Online Library (2002)
5. Google Ventures site: http://gv.com/sprint
6. Sprint case presentations: https://sprintstories.com/

7. Kropp, M., Meier, A., Perellano, G.: Experience Report of Teaching Agile Collaboration and Values: Agile Software Development in Large Student Teams, IEEE 29th International Conference on Software Engineering Education and Training (2016)
8. Read, A., Derrick, D.C., Ligon, G.S.: Developing Entrepreneurial Skills in IT Courses: The Role of Agile Software Development Practices in Producing Successful Student Initiated Products, 47th Hawaii International Conference on System Science, pp. 201-209 (2014)
9. Mahnic, V.. A Capstone Course on Agile Software Development Using Scrum, IEEE Trans on Education, Vol 55, 1 (2012)
10. Paasivaara, M., Blincoe, K., Lassenius, C., Damian, D., Sheoran, J., Harrison, F., Chhabra, P., Yussuf, A., Isotalo, V.: Learning Global Agile Software Engineering Using Same-Site and Cross-Site Teams. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. 285-294 (2015)
11. Spradley, J.P.: Participant Observation. USA: Thomson Learning. (1980)
12. Yin, R.K.: Case study research. Design and methods (5th ed.). USA: Sage Publications (2014)
13. Vesikivi, P., Lakkala, M., Holvikivi, J., Muukkonen, H.: Team teaching implementation in engineering education: teacher perceptions and experiences. European Journal of Engineering Education. Vol (44) Issue 4, pp 519-534 (2019)