Lean Software Startup – an Experience Report from an Entrepreneurial Software Business Course

Antero Järvi, Ville Taajamaa, Sami Hyrynsalmi

University of Turku, Department of Information Technology, Finland
{antero.jarvi, ville.taajamaa, sthyry}@utu.fi

Abstract. This paper offers blueprints for and reports upon three years experience from teaching the university course "Lean Software Startup" for information technology and economics students. The course aims to give a learning experience on ideation/innovation and subsequent product and business development using the lean startup method. The course educates the students in software business, entrepreneurship, teamwork and the lean startup method. The paper describes the pedagogical design and practical implementation of the course in sufficient detail to serve as an example of how entrepreneurship and business issues can be integrated into a software engineering curriculum. The course is evaluated through learning diaries and a questionnaire, as well as the primary teacher's learnings in the three course instances. We also examine the course in the context of CDIO and show its connection points to this broader engineering education framework. Finally we discuss the challenges and opportunities of engaging students with different backgrounds in a hands-on entrepreneurial software business course.

Keywords: software entrepreneurship, education, software business, lean startup, CDIO

1 Introduction

Due to the global changes in business landscape, software entrepreneurship is currently a popular and an important topic to teach to students. Recent development in the industry has created the lean startup method that aims to speed up startup evolution and eliminate waste during the process. While the lean startup movement started in the software entrepreneurship domain, its principles are currently spreading to other, more tangible domains. Established companies, such as F-Secure and Tieto in Finland, have founded small startup-like teams inside the corporation to develop products and services for volatile market segments.

There is, however little evidence on how lean startup method works as a teaching tool. Thus, the research objective of this study is to evaluate the usefulness of the lean startup method in incorporating entrepreneurial, business and transferable working life skills into a software engineering project course.

In this paper, we describe the course "Lean Software Startup" that has been taught yearly in Department of Information Technology at the University of Turku, Finland

since 2011. In addition to technical students, the course has participants from the business faculty which serves the interdisciplinary goal of the course. The paper presents the used pedagogical strategy and discusses and shares the experiences gained teaching the course during the previous three years. The course design and implementation are discussed in detail so the same principles and structure can be adapted by others.

The rest of the paper is structured as follows. The following section provides a brief introduction to the related concepts as well as motivation for the course and briefly describes related work. Sections 3 and 4 present the design of the course and its evaluation, respectively. The final section concludes the study with discussing challenges and proposing further ideas for development.

2 Background and Motivation

2.1 Customer Development and Principles of Lean Start-up

Blank [1] presented a model that helps startups to build and improve their success by acquiring a better understanding of their customers. The model consists of the four steps as presented in Fig. 1. The first step aims to identify the customer segments and how they value the problem that the product or service proposal tries to solve (so-called Problem/Solution fit). The second step attempts to prove that there is a market for the product or service proposal that positively response to the problem (so-called Product/Market fit). The third step focuses on scaling the market by creating and driving customer demand. The fourth step aims to transform the startup firm from a learning and discovery organization to a business execution machine. During this course, only the first two steps are addressed.

In industry, the lean start-up method by Eric Ries [2, 3] is an extremely popular tool for technology start-ups to manage the creation of the new company. The initial model [2] was built on the top of three principles. These are: 1) the use of free and open-source software or cheap software development platforms; 2) the use of Agile software development methodologies; and 3) the use of Blank's [1] Customer Development method. The fourth basic principle to the model, the use of cheap and effective analysis tools, was added in [4]. To summarize, these principles aim to cheaply develop a 'minimum viable product' (MVP) [5] that can be used to empirically test customers' real needs.

The lean start-up model has since been redefined by Ries in [3]. The basic philosophy, however, remained the same. In [3], the Build-Measure-Learn -loop (Fig. 2) is raised to a central position in the lean start-up model. The loop guides start-ups to turn

Fig. 1. Customer Development model (adopted, [1])

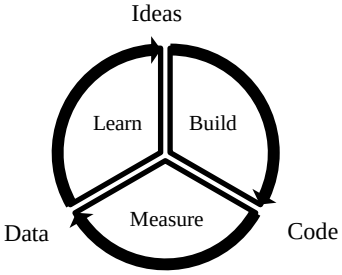

Fig. 2. The Lean Build-Measure-Learn –loop [3]

ideas into products, measure the customers' response and learn from this data. This process is a fundamental part of the lean start-up method and the process is repeated over and over again. Furthermore, the same learning process is applied not only on the product but also to test the assumptions underlying the business model.

2.2 Related work

Course, curriculum and teaching methods for research and development have long traditions [6, 7, 8]. Especially in engineering education [9], the achieved learning outcomes, implications of those to curriculum design and to teaching methods are subjects to continuous scientific discourse [10, 11]. Concerning curriculum design and development, the current study concerns a course level approach [9, 12]. With respect to teaching methods, this study focuses on transferable working life skills, action and integrated teaching methods [7, 13]. Research in the field, in this study, is often conducted using action research and case study analysis methods [14].

Related work shows that using these teaching methods that are based on social constructionism and hands on learning, do catalyze the students cognitive learning process at a deeper level [7, 8, 9, 13]. Students not only learn disciplinary knowledge but also relevant and transferable working life skills such as communication skills, teamwork skills, project management and creative product development skills. Typically the challenges in this kind of courses lie in the assessment of the learning outcomes, which are subjective and context-driven. Often the cases cannot be directly compared to other similar studies. Even if the courses share similar structure and intended learning outcomes, the teaching methods or teachers can be different and the identification of differences is difficult whether they are qualitative or quantitative [15, 16]. The value of research based on course development and assessed learning outcomes is especially relevant to practitioners who can reflect their own experiences from course planning, teaching methods development, assessment of learning outcomes and the feedback of all stakeholders to their own praxis [9, 13].

There are similar proposals for using lean start-up methodology, or respective constructions, as a base for an education design. For example, [17] proposes a design for a game development accelerator based on the method. In [18], the authors describe an incubator that, to some limit, meets the principles of lean start-up methodology. Fur-

thermore, some experience reports exist. For example, Bosch et al. [19] tested an extended method with students in a start-up incubator.

Lean Software Startup shares similarities with Capstone courses [7,9-13,15-16]. It follows a product development process with emphasis on early phase iterations and active customer feedback through prototyping. Also the intended learning outcomes include transferable working life skills in addition to disciplinary knowledge.

There are a few proposals for innovative software engineering course with a special focus on entrepreneurship. Björkqvist et al. [20] report experiences on integrating entrepreneurship activities in a large project work course that involves both information systems and computer science students. Daimi and Rayess [21] describe a software entrepreneurship course for computer science students; however, their pedagogical approach is close to traditional lecturing while our pedagogical strategy is based on hands-on learning. Aaen and Rose [22] note that software entrepreneurship courses often utilize a plan-based approach. Thus, they developed a course that allows students to select from plan- based or agile alternatives an option to complete a software entrepreneurship course.

3 Course Design

3.1 Design Goals

The starting point for the course[1] design was to let students experience a product or service development project based on the lean startup method, as realistically as possible in the university environment. The use of a startup context where teams come up with their own business ideas is not an end in itself, but rather a means for achieving a setting where there is uncertainty and thus opportunity for innovation both in terms of product and business. If the customer and the problem were given, which is often the case in software engineering project courses, the learning would be limited to software engineering, project and teamwork skills. In this course we wanted to offer a wider scope. It must be noted that this is not an entrepreneurial course *per se*, but uses the lean startup method as a tool for creating innovative products and related business designs in a customer or user driven manner. It is clearly explained to students that the approach applies to any new product development under considerable uncertainties and risks.

A central principle in this course is that there are no explicit disciplinary knowledge learning goals for the course, thus no predetermined, planned knowledge to assimilate, and also no exam to ensure the learned substance. However, during the course the students will learn and apply theoretical knowledge, techniques and methods, and study various materials. The course has six focus areas:

Working with customers/users and other stakeholders.
The students experience how difficult it can be to learn about customers' problems

[1] https://nettiopsu.utu.fi/opas/opintojakso.htm?id=34761&lang=en

and how to help the customer understand his/her needs. . The students are responsible for finding and organizing the interaction with customers. They experience the value of talking to real customers. and learn to seek for feedback and take value also from criticism.

Lean Startup method in product development.
The students learn the basic ideas of the lean startup process and apply its core learning loop. The students understand that the MVP is a means for validating assumptions about the business being developed. The students experience throwing away code and changing central decisions about the customers, customer value, product and business. The students see how business and product development go forward in parallel.

Practical software engineering skills.
The teams choose the development tools and environments according to what is needed and put them to use. Learning new tools and languages is a normal activity during the course.

Working life skills.
The students experience practical teamwork, will put up communication and collaboration tools as needed, learn to present and pitch their project on several occasions and to different stakeholders. Overall professionalism and taking responsibility is required.

Idea generation and business development.
The students learn to see opportunities and generate business ideas. They understand the importance of focusing on customers and problem first and not starting with the solution. They experience how the business idea is refined based on feedback.

Overall business knowledge.
The students will learn overall business skills and knowledge. These are not however actively taught in the course; instead, these are discussed when the issues arise during the course.

The projects need not, and generally will not, achieve a state where they could be said to be ready and finished. It is more important that the lean startup method is used, the team experiences several rounds of its core learning cycle and the business idea as well as the product is adapted based on this feedback. In essence, the journey is important, not the endpoint.

Team formation is one of the key success factors in the course. Team size has varied during the three course instances but 4-5 members seems to work best, providing for enough skills and workforce to get things done, yet still small enough that all members are engaged, feel responsible and find a role within the team. One of the students acts as a ‘team liaison’ with a responsibility to keep up with the team’s status and to be the connection point with the instructor. However, the team liaison is not a team leader in traditional sense as the teams will organize and manage themselves like

agile teams. When the teams are formed, the instructor ensures that all teams have sufficient software engineering and business skills.

3.2 Pedagogical setting and learning environment

A guiding pedagogical principle in this course is that learning happens in the context of doing and experiencing things in practice. Two central pedagogical methods to achieve this are the actual work that the students carry on in teams and weekly team mentoring sessions. Other used methods include introductory lectures, idea generation workshops, progress gates, use of supporting materials, learning logs for self-reflection and a course debriefing in the end.

Teamwork.
Similar to a real software-based startup, the students have to work on a wide range of different issues. Examples of tasks that the teams typically face include configuring development environments and tools, designing and implementing the product, designing the user experience, launching the product, contacting potential customers and other stakeholders, getting customer feedback with interviews and product use analysis, learning about the business, making business and product decisions, organizing the teamwork internally, planning and allocating work, *etc*. The teams have practically full freedom to choose what they will work on and how they will work. Many of these things are new to the students and they simply need to learn for what is needed. Often, one of the students has experience in a particular area and the students learn from each other.

Mentoring.
Individual one-hour mentoring sessions are held roughly once a week with each team. The sessions serve several purposes. First, in the sessions the instructor can directly give advice, teach relevant theory and point out materials to help the team go forward in a particular situation. Second, the instructor ensures that the lean startup- and customer development methods are being used. Third, the instructor helps the team resolve whatever issues there are that hinder the teamwork, especially in the beginning. It is very important that the instructor does not take the role of an authority that requires weekly progress reports, but rather the role of a more experienced team member. Unless the atmosphere is open and encouraging, the students will not present the true status of the project and ask for help with difficulties they are facing. The instructor does not interfere with the business idea and business development itself, unless it needs to be pushed to a pedagogically more fruitful direction; for instance, out of a customer segment where it would be impossible to reach customers.

Gates.
There are four progress gates in the course. At each gate, each team gives a short presentation on their project, followed by comments and discussion by the other teams. The purpose of the gates is first to create structure for the four month long course, second to enable learning between teams and third to provide presentation

opportunities for students. The four gates have different focuses: In the first gate the focus is on the initial business idea, team organization and technical development issues. In the second gate the teams focus on customers and stakeholders, value proposition and the first working product (MVP). In the third gate, the teams will explain what feedback they have gathered, what kind of business and product decisions they have taken based on the feedback, and present the evolved MVP. The fourth gate is similar to the third, incorporating one more lean startup learning loop. Whereas the presentations in the previous gates are more traditional, the presentation in the fourth gate is a typical startup business pitch for a wider audience than just course participants.

Lectures and idea generation.
In the two first weeks there are three to four two hours lectures covering the overall course concept and introducing the main points of central tools used during the course – Ries' lean startup method [3], Blank's customer development [1] and Osterwalders' business model canvas [23]. Opportunity recognition and idea generation techniques are introduced and immediately used in an idea generation workshop, wherein the initial business ideas for the teams are generated.

Deliverables.
There are three deliverables in the course, the learning log, the daily diary and the team log, all of them created gradually during the course. There is a checkpoint during the course to make sure all students are actively creating these deliverables. The purpose of all deliverables is explained to students.

The most important deliverable is the learning log, a semi-structured template where the students, once a week, write their perceptions on what they learned during the week (see the template in Appendix A). The primary purpose of the learning log is to make the students think back and reflect upon what they have learnt in the unstructured and sometimes messy work during the week. The secondary purpose is to provide feedback for the course instructor about what the students really learn.

The daily diary is a log where the students make a one line entry every time they work on anything in this course. The entry contains the date, number of hours worked, what did the student work on, and with whom. The primary purpose of this diary is to ensure an even workload between team members.

The team log makes the team's journey visible. Once a week, or more frequently if the team chooses to, the team writes briefly what did they do since the last log entry, what did they learn about their business case, how it affects their assumptions, and what are currently the important concerns that the team will act on next. It has turned out that reading through this log at the end of the course is an invaluable learning experience for the students. It clearly shows the 'searching by experimentation' nature of the lean startup method.

Reflection discussion.
At the end of the course, there is a moderated discussion aiming at providing final emphasis the most important learning goals of the course and providing course feed-

back for the instructor. This has not worked well due to lack of motivation. The students put a lot of effort into the fourth gate, and once it is passed, they feel that the course is over. This is one of the last activities before the end of semester and that might explain the lack of motivation at this point.

3.3 Course structure

An example course structure is given in Table 1. The course consists of four parts as described in the table. At our university, a semester is divided into two periods, roughly 7-8 weeks each. The "Lean software startup" course lasts two periods, ending before the summer break. Currently, the course corresponds to 10 ECTS (European Credit Transfer and Accumulation System) credits.

Table 1: The normal structure of the course

Week	*Content*
Part I: Introduction	(all sessions are 90 min).
Week 1: 1st	Course introduction. Practical issues. Lecture: "Concurrent business and product innovation".
Week 1: 2nd	Team formation. Deadline for dropping the course! Lecture: "Lean startup basics", "Customer development basics".
Week 2: 1st	Lecture: "Opportunity recognition and business idea generation". Idea generation warm-up.
Week 2: 2nd	Idea generation workshop, presenting the ideas.
Part II: The foundations	
Week 3	Team mentoring, individually 60 min / team. Focus on business idea, team organization, project tooling, team organization and management, team roles.
Week 4	Team mentoring. Same as above.
Week 5	Emergency mentoring available by appointment if needed. **1st Gate**: Presentation of the business idea and how the team is organized. 10 minutes presentation + group discussion.
Part III: Execute	
Week 6	Team mentoring
Week 7	Team mentoring
Week 8	Team mentoring
Week 9	Emergency mentoring available, by appointment – in case a team is in trouble. **2nd Gate**: First product (MVP) demonstration and customer acquisition, evolution of the business idea.
Week 10	Team mentoring
Week 11	Team mentoring
Week 12	Team mentoring
Week 13	Emergency mentoring, again only by appointment **3rd Gate**: Product demonstration (MVP) and customer feedback, evolution of business idea and the product.
Part IV: End game	
Week 14	Team mentoring
Week 15	Team mentoring
Week 16	**4rd Gate**: Demo Day. Business pitch, 5 min., followed by group discussion to share experiences from the last weeks.
Week 17	**Retrospective**: Moderated discussion: What did we learn. Course closing.
Week 18	Deadline for all course deliverables.

4 Evaluation

In the following, we evaluate how well the goals of the course were achieved, using the learning diaries and the self-reported data from the students. The course has been organized three times, first in the autumn semester of 2011–2012 (started in September and ended in December, 2011); second time in the autumn of the academic year 2012–2013 and third time in the spring semester of 2013–2014 (started in January and ended in June, 2014). While the structure of the course has remained the same, there were no business students in the first instance. Therefore, in the evaluation we focus only on the last two instances.

Table 2 shows the number of students that participated in the course in each instance. In addition, there were a few students who enrolled in the course but dropped out the first week of the course.

Table 2. The number of participants in the course by their major disciple

Discipline	*Autumn 2012*	*Spring 2014*
Technology	19	16
Business	10	8

To evaluate the implementation of the course, we used the semi-structured learning diaries, the 'learning logs', written by the students during the course as the primary data source. As a secondary data source, we used a questionnaire sent to the participants of the course in December 2014.

As a part of the course, the students were required to write and update a structured, weekly learning log. An example of the structure is given in Appendix A. The shortest learning diaries were 2 pages and the longest 15 pages, the average being 5 pages. The learning diaries were analyzed by the authors by reading them carefully and counting how many times the student had written about learning something in each of the six learning areas. The learning area was given the score 0 if the student had no learning experience on a learning area, score 1 for at least one reported learning experiences, score 2 for repeated reported learning experiences and score 3 for repeated and deep learning experiences. The distinction between scores 2 and 3 is subjective and reflect the difference between how strongly the students described the learning. The averages for each learning area were then calculated, for all students together and separately for technology and business students (Table 3).

Learning logs.

The findings from the learning log data indicate that a moderate to good amount of learning takes place in all learning areas. Furthermore, there was not a single student that reported not learning something in most of the learning areas. Interestingly, both technology and business students experienced the most learning in the "idea generation and business development" area. The possible reason for this is that the students had not previously participated in a hands-on business development course, even though many business students had studied the topic in previous courses. The hands-

on nature of the course is seen also in high "work life skills" learning experience. As anticipated, technology students learned more about software engineering, but interestingly this was the only area where there was a notable difference between the two student groups. The "general business knowledge" learning area had the poorest learning outcome.

Table 3. Learning experiences in learning areas

Learning area	*All students*	*Technology*	*Business*
Working with customers/users/ stakeholders	2.2	2.1	2.3
Lean startup method	1.8	1.9	1.8
Software engineering skills	1.9	2.2	1.5
Working life skills	2.3	2.3	2.3
Idea generation and business development	2.6	2.5	2.6
General business knowledge	1.3	1.3	1.2

In the learning log, the students also reported the source of the learning experience (Table 4). These learning sources were analyzed in a similar manner to the learning areas. Unsurprisingly, the team was the most common source, reflecting the constant peer learning during the teamwork. The mentor was also a frequent source for learning which indicates that the mentoring concept is a useful and working pedagogic method. Self was the third frequent source for learning, interestingly more for business students. Two other sources that were probed in the learning logs, customers and other teams, got only sporadic mentions in the reports. It is somewhat surprising that the group discussions in the four gates and frequent contacts with customers and stakeholders did not seem to provoke learning.

Table 4. Sources of learning

Source of learning	*All Students*	*Technology*	*Business*
Self	2.0	1.7	2.4
Team	2.4	2.5	2.4
Mentor	2.3	2.0	2.6
Other teams	0.4	0.6	0.3
Customers	0.6	0.4	0.8

Survey.

The results of the survey of the participants of the course are used as the secondary data source. The questionnaire was submitted via e-mail to students who have participated into the course during its previous instances. We targeted only students who were still active at the university and whose university email accounts were working. In total, the questionnaire was sent to 41 students (50% of the total course participants). The students had four weeks to answer the questionnaire. A reminder email was sent after two weeks. Finally, a total of seven usable answers were received, thus yielding the response rate of 17.1%.

While the learning diaries were written during the course, we were keen to see how the attitudes of the students changed awhile after the course. None of the respondents continued the project after the course. Four respondents reported that they have had discussion with their teams on the continuance of the course work towards a commercial product. However, they deemed that the idea or the team were not mature enough. Nevertheless, all answerers had a positive attitude towards founding their own software firm and working as an entrepreneur.

In almost every feedback, the teacher was mentioned as one of the most important source for learning. As noted by one business student majoring in Marketing:

> *"The thing that I liked the most in this course was how involved the teacher was in our work. This is really the only course where I think I could learn through one on one interaction if thesis courses don't count."*

Furthermore, both technology and business students praised the collaboration over the faculty borders. This was often mentioned as a learning outcome in the course.

> *"Teamwork with Software Engineers (it is really different than Teamwork with only other Business students)…"* – Marketing major.

5 Discussion

5.1 Experiences

The overall experience of the teacher in the three course instances is positive; the course is inspiring and even fun to teach. Most students are committed, enthusiastic and hardworking, and the student feedback is good. Clearly the course succeeds in engaging the students and thus provides a good foundation for learning. However, despite the word 'startup' in the course title, and words 'entrepreneurial' and 'software business' in the title of this paper, we would not consider this as a full entrepreneurship course, nor as a software business course. Rather, the course combines some elements from both themes into an intensive, multifaceted learning experience.

On the entrepreneurial theme, the students experience working and making decisions under uncertainty, taking responsibility on issues outside of their current skills, experience failing and learning from it, and gain insight into the entrepreneurial attitude.

On the software business theme, the students get to design business models. However, the discussions, designs and experiments concerned only the value proposition, distribution and marketing channels, customer segments and revenue models, thus providing a somewhat narrow view on software business development.

As a third theme, the students get to experience on experimental, user and customer driven innovation process that we believe is becoming widely used in cases where both the product and the distribution channel are digital.

5.2 Challenges

One of the challenges, noted during the execution of the course, is the requirements

that a multidisciplinary course imposes upon the primary teacher. The teacher needs to understand and be able to help both in business related and in technological issues. In our implementation, the teacher had a decade of experience of teaching software engineering as well as experience with running his own startups. This greatly helped to transfer the knowledge to the students; however, as a downside, the course is highly dependent on a single teacher and increases his workload. A second teacher with business background and specialization at software startups was also present in the four gates and gave invaluable insights – mainly on the business ideas and their potential and flaws. Getting a fresh, second opinion was clearly beneficial for the students.

This kind of course is also challenging for students, who in previous courses, have mainly been required to solve clearly defined problems. This course requires a different mindset: As a team, they need to (by themselves) identify what must be done, learn the required knowledge and skills and solve the problems. Frequent mentoring sessions with the teacher in the first weeks of the course were extremely important to the students.

Unsurprisingly, one of the major challenges in the course is the team formation. The objective is to have sufficient business and technical skills in every team. This has not always succeeded. In two cases the technical skills in a team turned out to be insufficient which was handled by swapping student volunteers between two teams. Lack of business skills has not been a problem. Either the commitment of business students has been higher or technical students have been able to learn business issues easily and contribute to business side as well.

5.3 Engineering professional: CDIO

We also examine the Lean Software Startup course in the context of the CDIO (Conceive–Design–Implement–Operate) engineering education structure and find four shared objectives and similarities that can be explicitly identified across the two.

The CDIO engineering education framework and structure was originally designed to better equip graduating engineers with transdisciplinary and scalable working life skills such as communication, project management, teamwork, and problem solving capabilities in addition to the actual disciplinary knowledge and skills [9, 13, 24]. This structure and framework level intended learning outcome is also the first joint surface boundary with CDIO framework and Lean Software Startup course [25, 26, 27].

The CDIO framework is based on 12 standards that emphasize a focus on learning outcomes instead of taught content, versatile assessment of learning, integrated curriculum, active learning and the learning of the engineering problem solving cycle: Conceive, Design, Implement and Operate which is also where the acronym CDIO derives from [27]. Entrepreneurial practices based on iterative engineering problem solving cycles such as that used the in the course “Lean Software Startup” are very much aligned with the CDIO framework and learning philosophy with a) emphasis on integrated curriculum, b) active learning methods, which emphasizes learning by doing, c) teamwork setting, d) and design-implement experiences, which are an essen-

tial part of both the CDIO as well as the Lean Software Startup course. This is the second joint boundary surface.

The third similarity and shared objective in this course and in CDIO is the emphasis Problem Based Learning (PBL) as the learning approach. The students actively construct knowledge coached and facilitated by the teaching team in a hands-on learning environment [6, 15, 29] instead of traditional lecturing where the teacher transmits information or his own interpretations of knowledge to the passively listening students.

The fourth explicit joint surface boundary [9, 11] between CDIO and Lean Software Startup course is the societal impact at which it is targeted. Lean Software Startup course catalyzes students' innovation skills and the construction and adaption of knowledge needed in the challenges of future industries as well as the building of global societies.

6 Conclusion

This paper presented the blueprint for the "Lean software startup" course that is built upon the customer development and lean startup methods. We reported the used course design and pedagogical strategy and evaluated the course with students' learning logs and with a short survey. The results show that this kind of a course can teach software engineering, software business and entrepreneurship skills to software engineering and business students. The course design relies on hands-on learning in multidisciplinary teams, which has been praised by the participants. No course is perfect and we will continue to develop the course in future. Furthermore, we call for experience reports, course designs and education evaluation by software business and entrepreneurship teachers to share knowledge and to further develop the field of software business education.

References

[1] Blank, S.: The Four Steps to the Epiphany: Successful Strategies for Products that Win. Second edition. Cafepress.com, San Mateo (2005)

[2] Ries, E.: The lean startup. Accessed on April 7, 2013 http://www.startuplessonslearned.com/2008/09/lean-startup.html

[3] Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. First edition. Crown Business, New York (2011)

[4] Cooper, B., Vlaskovits, P.: The Entrepreneur's Guide to Customer Development. CooperVlaskovits, USA (2010)

[5] Ries, E.: Minimum Viable Product: a guide. Accessed on April 7, 2013 http://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html

[6] Daniels M.: Developing and Assessing Professional Competencies: a Pipe Dream?, Digital Comprehensive Summaries of Uppsala Dissertations from Faculty of Science and Technology 738, AUU (2011)

[7] Leifer, L. J., Steinert, M. Dancing with ambiguity: Causality behavior, design thinking, and triple-loop-learning. Information Knowledge Systems Management 10, 151–173 (2011)

[8] Froyd, J.E, Wankat, P.C., Smith, K.A.: Five Major Shifts in 100 Years of Engineering Education, Proceedings of IEEE 100, 1344–1360 (2012)
[9] Crawley, E., F., Malmqvist, J., Östlund, S., Brodeur, D.R.: Rethinking Engineering Education, The CDIO Approach, Springer, USA (2007)
[10] Dym C.L., Rossmann J.S., Sheppard S.D.: On Designing Engineering Education: Lessons Learned at Mudd Design Workshop IV, Int. J. Engng Ed. 20, 470-474 (2004)
[11] Wesner J.W.: What We Have Learned in Mudd Design Workshop V: Learning and Engineering Design, Int. J. Engng Ed. 22, 685–688 (2006)
[12] Edström K., Kolmos A.: PBL and CDIO: complementary models for engineering education development, European Journal of Engineering Education 39, 539–555 (2014)
[13] Atman, C.J., Sheppard, S.D., Turns, J., Adams, R.S., Fleming, L.N., Stevens, R., Streveler, R.A., Smith, K.A., Miller, R.L., Leifer, L.J., Yasuhara, K., Lund, D.: Enabling Engineering Student Success: The Final Report for the Center for the Advancement of Engineering Education, Morgan & Claypool Publishers, San Rafael, CA (2010)
[14] Eisenhardt, K. M.: Building Theories from Case Study Research, Academy of Management Review 14, 532-550 (1989)
[15] Taajamaa, V, Sjöman, H, Kirjavainen, S, Utriainen, T, Repokari, L, Salakoski, T.: Dancing with Ambiguity – Design thinking in interdisciplinary engineering education, Design thinking conference, Shenzhen, China (2013)
[16] Taajamaa, V., Westerlund, T., Liljeberg, P. & Salakoski, T.: Interdisciplinary Capstone Project. 41th SEFI Conference, Leuven, Belgium, (2013)
[17] Järvi, A., Mäkilä, T., Hyrynsalmi, S.: Game Development Accelerator – Initial Design and Research Approach, Proceedings of From Start-ups to SaaS Conglomerate: Life Cycles of Software Products Workshop, Potsdam, Germany, 47–58, (2013)
[18] Callele, D., Boyer, A., Brown, K., Wnuk, K., Penzestadler, B.: Requirements Engineering as a Surrogate for Business Case Analysis in a Mobile Applications Startup Context, Proceedings of From Start-ups to SaaS Conglomerate: Life Cycles of Software Products Workshop, Potsdam, Germany, 33–46, (2013)
[19] Bosch, J., Holmström Olsson, H., Björk, J., Ljungblad, J.: The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups, Lean Enterprise Software and Systems. LNBIP 167, 1-15, (2013)
[20] Björkqvist, J., Petre, L., Rönnholm, K., Truscan, D.: Integrating Innovation Activities in a Master Level Capstone Project Course. International Conference on Engineering Education. Research reports 38, Turku University of Applied Science, 1065–1072, (2012)
[21] Daimi, K.., Rayess, N.: The Role of Software Entrepreneurship in Computer Science Curriculum," in Proc. International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'08), Las Vegas, Nevada, 53–62 (2008)
[22] Aaen, I., Rose, J. A Software Entrepreurship Course – Between two paradigms. 15th Annual Interdisciplinary Entrepreneurship Conference. St. Gallen and Zurich (2011)
[23] Osterwalder, Y., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. First edition. Wiley (2010)
[24] Taajamaa V., Westerlund, T., Liljeberg, P., Salakoski, T.: Interdisciplinary Capstone Project, 41th SEFI Conference, Leuven, Belgium (2013)
[25] Levina, N., and Vaast, E.; The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation Technology 738, AUU (2011)
[26] Couto, V., Mani, M., Lewin, A. Y., and Peeters, C.: The Globalization of White-Collar Work: The Facts and Fallout of Next-Generation Offshoring, Booz Allen Hamilton Inc. (2006)

[27] Levina N., Vaast E.: Innovating or Doing as Told? Status Differences and Overlapping Boundaries in Offshore Collaboration, MIS Quarterly 32, 307–332, (2008)
[28] http://www.cdio.org/implementing-cdio/standards/12-cdio-standards
[29] Savin-Baden, M.: Problem-Based Learning in Higher Education: Untold Stories, The Society for Research into Higher Education and Open University Press, Buckingham, UK, (2000)

Appendix A A template for a structure learning log

Learning Log

You do not have to answer to all questions every week, only when you have something to say. However it is important that you use this learning log as a weekly "check list", think and reflect back every question.

Please start any of your comments by entry Wxx where xx is the week number. There is an example in the third question how your log should eventually look like… so you are supposed to just use a single learning log document, not one for each week.

The learning log serves two purposes:

- Thinking through what you have learned in an unstructured work amplifies your learning (there is clear scientific evidence on this :-)
- Feedback for the course instructor on how this type of course could be improved

The Questions

1. What inspired you this week?
2. What was surprising?
3. What did you learn about entrepreneurship/business development?
   W38 Business ideas need only to be good enough to get started; the idea will evolve as learning about the customers, product, markets etc. takes place.
   W39 Long and detailed business plans do not work as tools for developing a business.
   W39 Osterwalder's business plan canvas as a tool for business development, just the idea.
4. What did you learn about product development?
5. What did you learn about software technology?
6. What did you learn about software engineering development practices / tools?
7. Comments on teamwork, good or bad. Anything that worked well or did not?
8. Where did most of the learning come from, from the mentor, your team fellows or you? Anything specifically worth mentioning?
9. Free comments