The Industrial designer's role in applying Lean Startup principles in IT-startups

A case study of Rendra AS

Christoffer Henne

Department of Product Design
Norwegian University of Science and Technology

ABSTRACT

Lean principles have proved to be effective in reducing waste in manufacturing processes and are now being applied to a growing number of industries through the Lean Startup methodology. This paper evaluates the Industrial designer's role in applying this methodology at Rendra AS, an IT-startup developing software solutions for the building industry. It looks at how principles from Lean Startup and the design process can be applied by executing pilot projects, summarizes lessons learned and argues that Industrial designers could play a key role in the implementation of Lean Startup principles.

KEYWORDS: Lean Startup, Industrial design, Software Development, TPD4505

1. INTRODUCTION

The Lean Startup methodology is all about making sure you are making what the customers want, by involving them at an early stage of the product development. This is also one of the key principles within Industrial design. This article compares the two methodologies, and argues for why and how Industrial designers can play a key role in the implementation of lean processes. The arguments are based on my background from Industrial design at NTNU and my personal experience from being CTO at the IT-Startup Rendra AS.

1.1 Method

This article is an attempt to share my insights from implementing Lean Startup principles in practise. Section 2 describes the methodologies used by Rendra during the scope of the study, and which are subject to discussion in section 4. Section 3 describes how the Lean Startup methodology was implemented in practice at Rendra and how Industrial designers were used in the process. Finally, section 5 summarizes main findings and conclusions.

2. BACKGROUND

2.1 Industrial design

Defining Industrial design is hard, as it covers many different directions. The Industrial designers Society of America defines it as follows: "Industrial design (ID) is the professional service of creating and developing concepts and specifications that optimize the function, value and appearance of products and systems for the mutual benefit of both user and manufacturer. ... Industrial design services are often provided within the context of cooperative working relationships with other members of a development group. Typical groups include management, marketing, engineering and manufacturing specialists. ... Industrial designers also maintain a practical concern for technical processes and requirements for manufacture; marketing opportunities and economic constraints; and distribution sales and servicing processes. They work to ensure that design recommendations use materials and technology effectively, and comply with all legal and regulatory requirements." (IDSA, 2010) The traditional design process have many variations, but some key elements are usually included; research, ideation, concept development, iterative testing and presentation. The result of the designer's work is finally handed over to the company for production, where necessary changes are made to make it producible and profitable. Even if the design department is in-house, there is usually a point where designing stops and implementation begins.

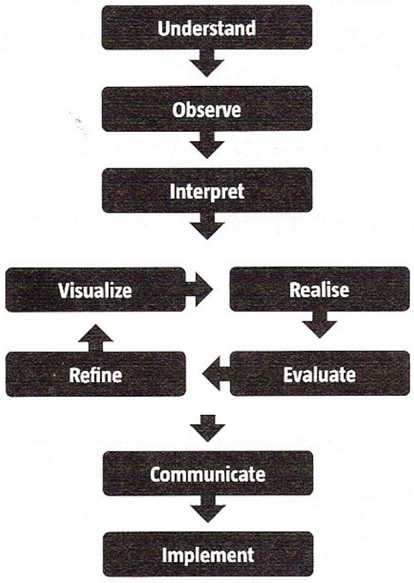

Figure 1: The IDEO process. Roope Takala, Turkka Keinonen, Jussi Mantere, Product Concept Design 2006

2.2 The Lean Startup

The term "Lean Startup" was first introduced in 2008 in a blog post by Eric Ries. In 2011 he published the book "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses". His principles build on thoughts from Lean Manufacturing, pioneered by Toyota, which focuses on reducing waste in form of overproduction, unnecessary inventory, waiting, defects among others, and ensure quality by testing every product or information before it is handed over to the next production step (EMS Consulting Group, 2012). Ries generalized these principles to make them applicable to any product development process. The basics of Lean Startup is that every assumption, every feature, every hour spent on development, should be validated at the earliest stage possible to reduce wasted development. In other words, as little as possible should be done before you know if it should be done at all. Ries calls this the "Build-Measure-Learn" loop; everything you build should be measured as early as possible, so you can learn from it. Based on what you learned, you either "steer" - adjust your development to accommodate the feedback - or "pivot" - change direction completely. Every Startup bases their big idea on a number of leap-of-faith assumptions; you assume you can build a sustainable business by providing a viable solution to a real need. To validate the set of assumptions this relies on, Ries stresses the importance of making Minimal Viable Products (MVPs). A MVP is that version of the product that enables a full turn of the Build-Measure-Learn loop with a minimal amount of effort and the least amount of development time (Ries, 2011). It can be a simple prototype, a paper mock-up or even a movie, which shows the envisioned product in use. This goes for all new features as well, don't just build - build, measure and learn.

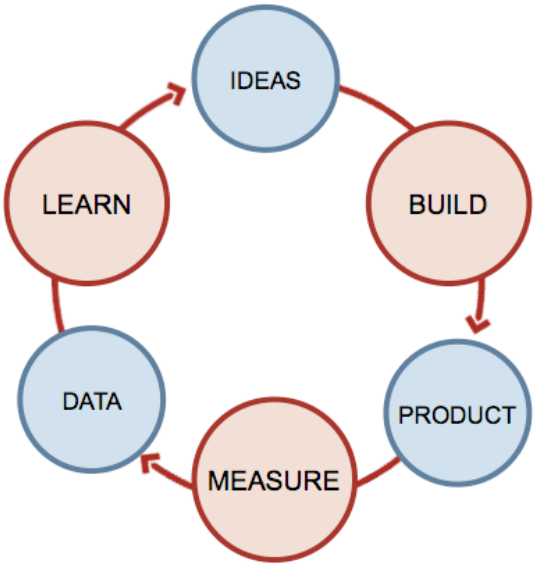

Figure 2: The Build-Measure-Learn loop, Eric Ries, The Lean Startup 2011

2.3 Agile development

Agile development is a common term for methods which apply timeboxed, iterative and evolutionary development, adaptive planning, promote evolutionary delivery, and include other values and practices that encourage agility - rapid and flexible response to change (Larman, 2004). Two of these methods are Scrum and Kanban.

2.3.1 Scrum

Scrum is a way for teams to work together to develop a product. Product development, using Scrum, occurs in small pieces, with each piece building upon previously created pieces. Building products one small piece at a time encourages creativity and enables teams to respond to feedback and change, to build exactly and only what is needed (Scrum.org, 2013). The three primary roles in the Scrum process are:

- **Product Owners** determine what needs to be built in the next 30 days or less.
- **Development Teams** build what is needed in 30 days (or less), and then demonstrate what they have built. Based on this demonstration, the Product Owner deter- mines what to build next.
- **Scrum Masters** ensure this pro- cess happens as smoothly as possible, and continually help improve the process, the team and the product being created.

2.3.2 Kanban

Another agile development methodology is Kanban, which Eric Ries recommends in his book "The Lean Startup". The basics of Kanban is to place all features into status categories, e.g. backlog, in progress, built and validated. Each of these categories or "buckets" can contain no more than three projects at a time. Once a bucket becomes full, it cannot accept more features. Different from Scrum, Kanban stresses the importance of validating features. Before a feature is validated it remains on the board, blocking for new features.

Backlog	In progress	Built	Validated
A B C	D E	F	

Figure 3: A simple Kanban board

3 Case: Rendra AS

Rendra AS was established in May 2012 by five students from the School of Entrepreneurship at the Norwegian University of Science and Technology (NTNU). The idea was to develop a solution for the building industry which enabled construction workers to display large 3D buildings on tablets and in a regular web browser. The starting point was an in-progress, break-through rendering technology which made this possible. During the scope of this study, the Rendra product development team consisted of two designers and five developers. Both designers and myself as CTO and front-end developer have backgrounds from Industrial design at NTNU.

3.1 Implementing the Lean Startup methodology at Rendra AS

The starting point for Rendra was an untested technology and a vision to launch a new product in a new market. Others had released iPad apps that let construction workers see 2D drawings and other documents at the construction site, but nobody had the technology to render gigabytes of 3D data on anything else than expensive desktop computers. There was great uncertainty related to the market, the technology and the customer value for the planned product. The economics was also a concern, as the company didn't want external investors to control the direction of the development or make it exclusive to any company or industry. Taking a Lean Startup approach gave us a way to finance the development by getting paid for the product while it was made, get close to the user group, get access to industry standard 3D data, get contacts and even ambassadors for the product in the industry and last, but not least, a chance to validate our assumptions regarding the products desirability and the customer's needs and willingness to pay. When the product was ready for launch, we would already have our first customers, reference projects and brand recognition.

3.2 Pilot projects

Potential customers were included in the development though Pilot projects. The deal was that they would pay a certain amount to cover the development cost of the product, give Rendra access to their models, participate in workshops and give continuous feedback on the product. In return they would get access to the product and all it's new features as they were being developed, and also get a period of free use after the pilot was over and the product was launched. They would also, effectively, have a chance of affecting the development in their favour, but unlike external investors, they would try to make the product better for themselves, not just more profitable in a short term perspective.

In our case the customers were companies, more specifically building entrepreneurs. We ran pilot projects with two companies in parallel, with ten test users from each company, and the time scope for both were six months. The same arrangement can be done with individuals, the number of test users can vary and the time scope of the pilots can be shorter or longer. It depends on the complexity of the product, available resources, the diversity of the product's target group and expected development time.

3.3 Preparation

Before the pilots could be initiated, some elements were in place; an initial product strategy and long term product vision, an initial plan which included milestones and time allocation for different tasks, both for the design team and development team, and our first MVP. When it comes to planning, Ries says: "...although we write the feedback loop as Build-Measure-Learn because the activities happen in that order, our planning really works in the reverse order: we figure out what we need to learn and then work backwards to see what product will work as an experiment to get that learning." At Rendra, we tried to follow this principle by first finding out what we needed to learn about the most uncertain features, then mock up designs which would give us the answers, then use what we learned when implementing the features and finally validate them in a real-life context. Another factor when making the plan was uncertainty regarding development; how long will it take and is it possible at all? The last factor we considered was dependencies. Some features simply needed to be developed first to enable other features.

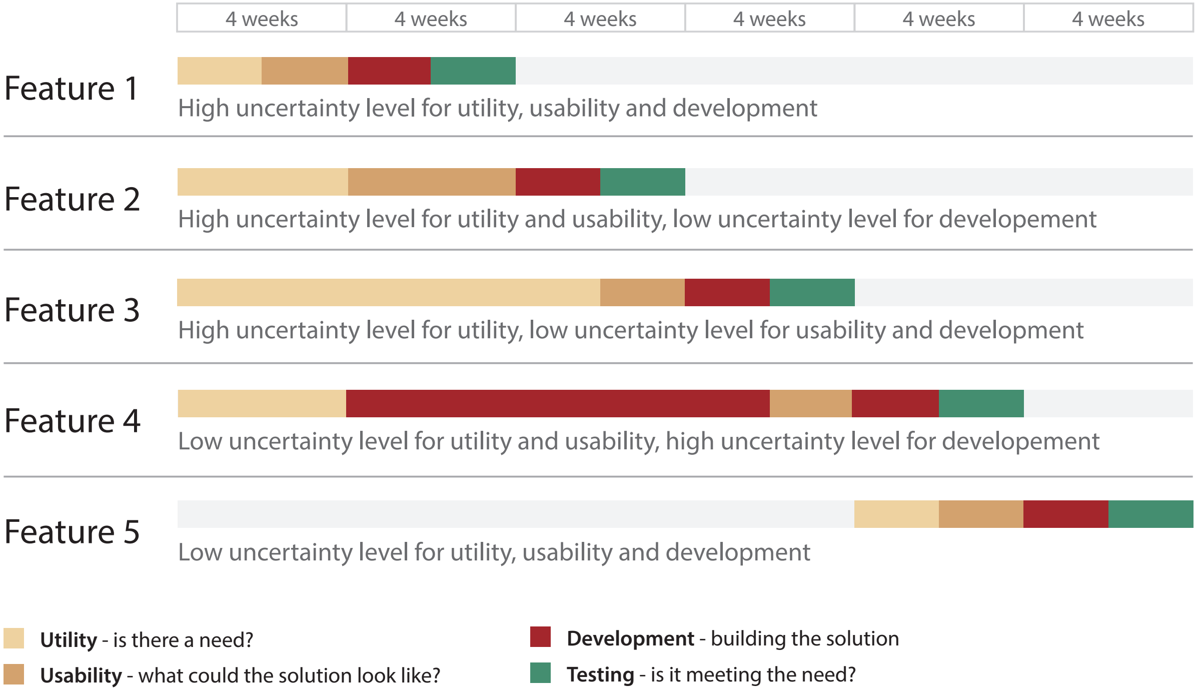

Figure 4: Example of a product development plan based on reducing uncertainty

Figure 4 shows how different tasks were prioritized for each feature of the product. The utility task meant validating whether there was a real need for the feature. This was validated through observation, workshops with paper prototypes and talking the potential customers through scenarios. If the feature didn't prove to have sufficient value, it was put on hold for a future version, changed to fit a different need or cancelled. If it turned out there was a need, the usability task followed, focusing on finding a good solution for the need. This was done with more detailed paper mock-ups, and the development team was included in the ideation phase. Once the design was validated by both users and developers, the development could start. At last, a MVP of the feature was shipped with a product update and tested in real life.

In the example illustrated in figure 4, Feature 1 has high uncertainty regarding both utility, usability and development, thus it is a big risk to not get those answers early. If the feature turns out to be focusing on a false need or the solution is hard to design and/or develop, this will potentially force us to rethink the entire product strategy.

Feature 2 has high uncertainty regarding utility and usability, but not so much regarding development. It means that we should validate early if we need it at all and work on the design of it, but we don't have to start developing it early, as we expect the development to be straightforward.

Feature 3 represents a feature that is expected to be easy to design and develop, but we are not sure about the need for it. Thus, we should wait till the need is validated before we spend time on the design and implementation of it.

Feature 4 is special in the way that it has more of an R&D profile with very high uncertainty connected to development. To avoid wasted work, we need to first validate the need for it, and if there is a sufficient need, start the development as early as possible to validate if, and how, it can be done at all.

Feature 5 has low uncertainty regarding both utility, usability and development, thus is can wait till all other features have been validated.

This doesn't mean it's the least important feature, it means it is safer to wait with it and not risk having to change it after it has been implemented based on new insights related to the other features.

3.4 Processing input

The initial plan was based on untested assumptions, thus it needed to be validated and changed constantly as the team gained more knowledge of the product's user group, market, and technical challenges. At Rendra, the Product Manager was responsible of continuously updating the plan and communicate this throughout the company. This meant gathering ideas, feedback, requests and concerns from all the stakeholders in the project, evaluate this in the context of the product strategy and prioritize and plan tasks based on uncertainty. This is visualized in figure 5 and explained further below.

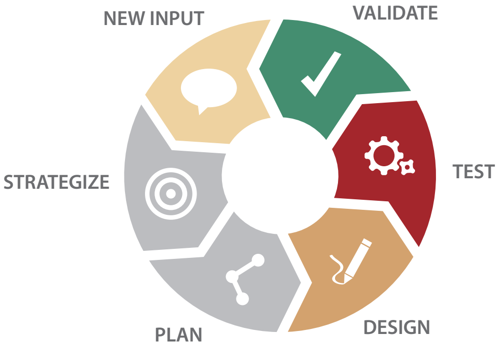

Figure 5: How input during the pilot projects was processed at Rendra

3.4.1 Get input

Input can basically come from anywhere; first hand observation, internal or external workshops, industry events or articles, ideas or concerns from within the company, availability of new technologies, new announcements from competitors, new time estimates for a certain feature from the development team etc.

3.4.2 Strategize

The product strategy states what needs the product aims to fulfil for who, what value it provides, how it is positioned in the marketplace, how, when and to whom it will be sold and how it should be perceived by potential customers. As the initial product strategy is based on assumptions, it needs to be validated. At Rendra the product strategy was re-evaluated and adjusted constantly during the first pilot projects. All inputs were evaluated in the context of the product strategy; is the input relevant for the product given the product strategy and, in either case, should the product strategy change based on the input?

3.4.3 Plan

If the input was found to be relevant and implied changing the product, the product plan was re-evaluated. Every change in the design of the product can potentially affect the entire company. Even seemingly small changes can make big impacts for the development or market team. Therefore, the first task that was planned was how to answer the questions: should this change be made and if so, what will it affect? The next question to be answered was: can it be done and how? and the last question, after the change was tested: does it work in practice?

3.4.4 Design

The design phase included ideation and concept development. Ideation sessions were arranged both internally, externally and with a mix of designers, developers and test users.

3.4.5 Test

Mock-ups were made to communicate concepts to both test users and developers. The objective for the first mock-ups was to validate the idea itself and answer the question: should it be built? If the answer was yes the objective of the next mock-ups was to answer how it should be built. All feedback was processed as new input and the product strategy, plan, and design was changed accordingly.

3.4.6 Validate

Once the design was accepted by both the test users and the development team, the design was implemented into the working prototype and tested in practice. This would answer the last questions: is it being used, does it work in practice and is it really adding value to the customer? If the answer to these questions were yes, the design was frozen until commercial launch, giving the development team time to make it production ready. If for any of the questions the answer was no, we discovered why and treated this as new input.

3.5 Example

A feature which changed several times during the pilot projects at Rendra was deviation reporting from the construction site. We initially planned to let users use our iPad app to report issues discovered at the site and send these back to the office. This was one of the most frequent requests we got from talking to potential customers. There were already web based systems for handling deviation reports in building projects, and our test users stressed the importance of integrating our solution with several of these systems for the construction workers to use it. This was planned to be the subject of several workshops, and we prepared to partner with at least one 3rd party to deliver what the potential customers asked for. Because of the complexity of the feature, we wanted to validate the need by first testing a simpler feature. We enabled users to make bookmarks in the 3D model of the building, attach comments and photos and share them with the rest of the test users. It turned out that, despite what they said earlier, this was sufficient in most cases. Since they liked our quick solution better than what they had, they now asked for printing and e-mailing options instead of integrating it with their current systems. Not only did this discovery save us from wasted development, it changed our entire product definition from a viewer and reporting tool, to a real-time problem-solving tool. Could we have reached this conclusion without testing our first implementation in a real life context? Maybe, but it would be very risky, and hard to sell it without having the pilot projects as good references.

4 Discussion

4.1 "Leanifying" the design process

The traditional design process is done in a static context. It starts with analysing the market and the resources within the company and ends with a deliverance in form of a report, presentation and/or prototype. The actual development of the product is not started before the design is done, which means all necessary adjustments to the design, to optimize it for and during development, are done after the conclusions for the design are made. This potentially creates a gap between the implemented solution and the intentions from the design team, and the final product is not validated in a real-life context until it is launched. The problem is clearly illustrated in the IDEO design process (figure 1), which has "Communicate" as a single stage in the process before "Implementation". This is done after the iterative part. In many cases this might be acceptable, especially if the product makes use of matured technology and the use cases of the product can be realistically simulated.

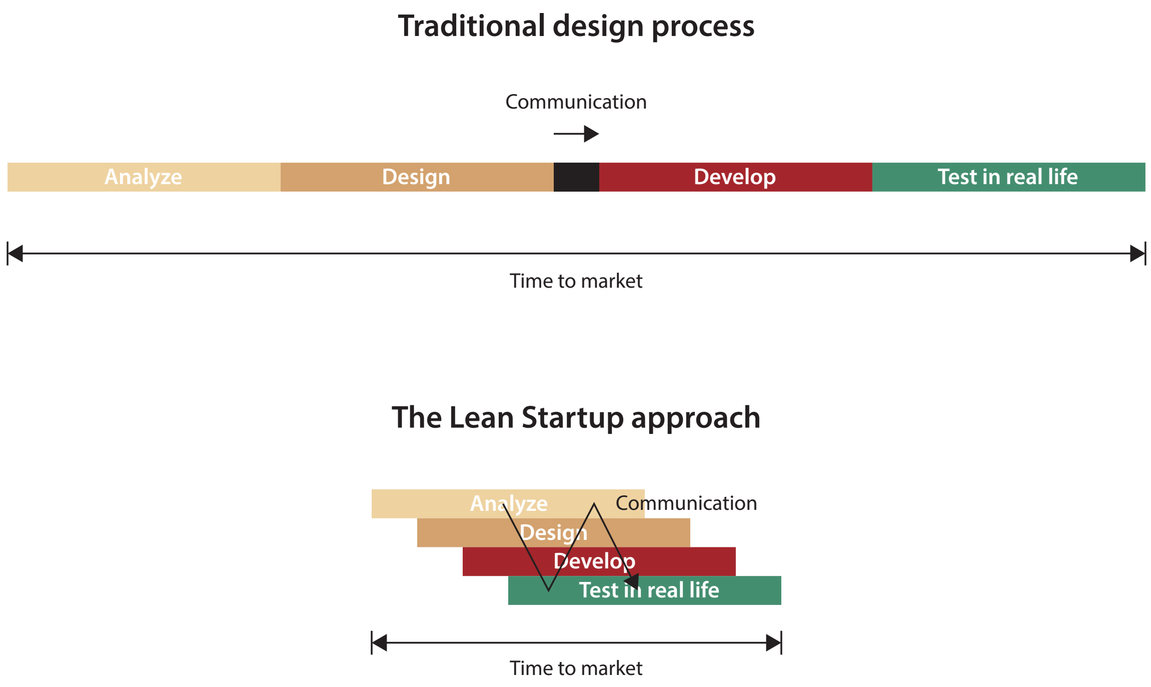

Figure 6: Comparison of the traditional design process and the Lean Startup approach

The Lean Startup approach closes this gap by developing the design and implementation in parallel. My experience from Rendra indicates that, when developing a new product in a new market, this substantially reduces the risk of not targeting the needs of the customer and also greatly reduces time to market (see figure 6). However it sets very high demands on planning and communication throughout the company. Especially between designers and developers, whose culture is, from my personal experience, fundamentally different.

4.2 Working with developers

For designers it doesn't really matter what happens under the hood, as long as the user experience is good. The quality of the product is what the user sees, and testing a product means usability testing. For many developers the perfect GUI is still the terminal. For them, good quality means effective code, scalability, brilliant software architecture and robust error handling, and testing the product means writing test units, which basically run macros which checks if the code returns certain values given certain input. Here are some of the lessons I have learned from working with developers:

1. Know each other's frameworks. There are various methods for agile development within software development, like Scrum and Kanban, and these can very well work together with the Lean Startup principles. In fact, they both increase transparency and provide a good overview of the development process even for non-technical co-workers, and thereby contribute to a good communication platform between developers and designers. No matter what framework the development team makes use of, the design team should know it and use it in communication. The same goes for the development team. They should be introduced to the design process used by the design team so they understand their mindset better.

2. Prepare developers for possible changes. As mentioned in section 3.4.3, even small changes in the design can make a big impact on the development. For instance, if the product makes use of databases, changing relations between data might require the entire database architecture to change, and lots of code to be rewritten. The more optimized the code is for the current database architecture, the more work needs to be done to change it. That is why it is absolutely crucial that the design team prepares the development team for possible changes, and makes sure the development team shares the "build for testing" mindset.

3. Not all development is visible. Building for testing can only go that far before the code becomes a mess, and adding new features becomes close to impossible. It is hard for designers to appreciate development which is not visible neither for them nor the users, still giving the developers time to clean up the code (refactoring) after a feature is validated, is absolutely necessary.

4. Include developers in the ideation phase. Minor tweaks in the design can make a huge difference in developing time. To be able to spend the least amount of developing time to test a feature in a real life context, designers and developers have to work together to find the minimal viable solution. The key is often to find the right level of fidelity and faking. How good does it have to look and what data can be faked in order to give the users a realistic enough user experience? The answers of these questions have to be communicated clearly, so the development team doesn't spend weeks perfecting the looks from a mock-up which was never intended to visualize the final solution.

4.3 Bridging the gap between design and engineering

The Industrial design education was initially intended to bridge the gap between designers and engineers, which is why the Industrial design curriculum includes many engineering courses. Many Industrial design programmes offer classes on Flash, Java and other programming environments. The purpose of these classes seems however to enable designers to build working prototypes for their designs, instead of giving them a basic understanding of software development in general. For product designers it is very useful to know basic mechanics and material science to give rough estimates on things like dimensions, production costs and sustainability. It also prepares them to communicate with engineers. For interaction designers working on software solutions, giving equivalent estimates would help them make more feasible designs, and more importantly, help them communicate and co-operate with software developers.

5 Conclusions

The following conclusions are based on my personal experience from working in a software startup, developing a new product in a new market, using unproven technology.

Despite the narrow scope of my study, it reveals the need for a universal role, which I believe Industrial designers are well suited for; being facilitators for the Lean Startup process. Managing the process of turning needs and resources into products in a cross-disciplinary setting involving users and engineers, is what Industrial designers are trained for. The study confirms the relevance of the Lean Startup approach, in particular the advantage of designing and developing the product in parallel, given that the process and the necessary communication between disciplines are well facilitated. It also suggests that, to regain the role as a bridge between design and engineering, Industrial designers working with software developers need to know the basics of software engineering. Not only to enable them to develop prototypes, but to enable them to communicate with developers, which the study identifies as the most important success factor for Lean Startups.

REFERENCES

IDSA (2010). *What is Industrial Design*? http://www.idsa.org/what-is-industrial-design [Accessed 27-04-2013]

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

EMS Consulting Group (2012). *Lean Manufacturing* http://www.emsstrategies.com/lean-manufacturing.html [Accessed 27-04-2013]

Larman, C. (2004). *The Agile Software Development Series. No. 5.* Addison-Wesley Professional

Scrum.org (2013). *What is Scrum?* http://www.scrum.org/Resources/What-is-Scrum [Accessed 28-04-2013]

Takala, R. and Keinonen, T., (2006) *Product Concept Design – A Review of the Conceptual Design of Products in Industry*, Springer