

The Relationship Between Business Model Experimentation and Technical Debt

Jesse Yli-Huumo^(✉), Tommi Rissanen, Andrey Maglyas, Kari Smolander,
and Liisa-Maija Sainio

Lappeenranta University of Technology, Lappeenranta, Finland
{jesse.yli-huumo, tommi.rissanen, andrey.maglyas, kari.smolander,
liisa-maija.sainio}@lut.fi

Abstract. The use of lean software development methodology and business model experimentation has become popular in software companies in recent years. Business model experimentation is used to validate assumptions made on a product from real customers before the actual product is created. A minimum viable product is used to test the business model by gathering and measuring customer feedback. However, in many cases creating a minimum viable product requires the development team to take shortcuts and workarounds in the product. This phenomenon in software development is called ‘technical debt’, where companies trade long-term software quality to short-term gain in time-to-market. We investigated four software companies and conducted nine interviews to understand the relationship between business model experimentation and technical debt. The goal was to study how business model experimentation is affecting to technical debt. The results showed that business model experimentation has a clear relationship to technical debt.

Keywords: Business model experimentation · Technical debt · Case study · Startup company · Large company · Software development lifecycle · Minimum viable product

1 Introduction

Startups and increasingly also larger companies use business model experimentation as a way to accelerate their product development cycles. The well-known process of business model experimentation is the lean startup framework introduced by Ries [1]. The lean startup framework considers learning to be the essence of the product development process and everything else is waste, following the lean manufacturing thinking. A lean startup creates a minimum viable product (MVP) that is a simple prototype of the product attached with a business model. The product team measures different elements of the product functionality and the business model, learns from the customer feedback and builds a better product with an adjusted business model to start the cycle again.

When a company accelerates its product development cycle to create a minimum viable product instead of releasing a ready and complete product, the development

team has to make shortcuts in the implementation of the product. In the software development lifecycle this is called ‘technical debt’ [2]. The term technical debt refers to a situation in the software development lifecycle, where long-term quality is traded for short-term gains. Taking shortcuts and workarounds in the development can give a company an advantage to release faster and to acquire customer feedback earlier, but if this ‘debt’ is not paid back later, it can affect to the quality and further development of the product.

When a new product is launched, it rarely has the optimal business model. The business model has so many elements and variables that it is impossible to predict how all components of the business model pan out when it is in the market. The lean startup process allows the tweaking of the business model efficiently.

The objective of this paper is to study the relationship between business model experimentation and technical debt. We explore if conducting business model experimentation has any effect to the amount of technical debt occurring during the software development lifecycle. We study four case companies and interview their key persons related to business models and technical debt and analyze the interviews for theoretical results.

The rest of the paper is organized as follows. Chapter 2 provides the background and the terminology related to this research. Chapter 3 describes the research process and methodology used in this study. Chapter 4 introduces the results analyzed from the gathered data. In Chapter 5 we discuss about the results and Chapter 6 concludes the paper.

2 Background

2.1 Business Model Experimentation

Every business enterprise either explicitly or implicitly employs a particular business model [3]. There are multiple interpretations of the concept, however. The business model can be defined as a system of interdependent activities that enables the firm to create value and also to appropriate a share of that value [4]. It can also be defined as the logic of the firm, the way it operates and how it creates value for its stakeholders [5] or the basic unit of business and process or operational advantages [6]. Business models generate feedback loops or virtuous cycles that strengthen components of the business model through iteration [5]. There are many other slightly different interpretations of the concept. In this study the business model is defined as the way a firm creates value and appropriates a share of that value following the definition by Zott & Amit [4]. The difference between a strategy and a business model is not always clear. Casadesus-Masanell & Ricart [5] see the business model as a reflection of the firm’s realized strategy.

Many business model studies take the dynamic nature of the business model into consideration. The current dynamic business environment with a multitude of simultaneous changes shortens the lifecycles of business models and requires companies to be constantly able and ready to adapt their business models. McGrath [6] points out that business models can rarely be anticipated in advance but rather learned over time

based on experiences and learning. Doz & Kosonen [7] also emphasize the need for companies to transform their business models more rapidly, more frequently and more far-reachingly now at the era of global competition, discontinuities and disruptions. Business model innovation is the term often used to refer the development of new business models. Business model innovation has been described as “a type of organizational innovation in which firms identify and adopt novel opportunity portfolios” [8], “the discovery of a fundamentally different business model in an existing business” [9] and “the search for new logics of the firm and new ways to create and capture value for its stakeholders: it focuses primarily on finding new ways to generate revenues and define value propositions for customers, suppliers and partners” [10]. Following the chosen business model definition, the business model innovation definition of Casadesus-Masanell & Zhu [10] is best suited for this study.

Minzberg & Waters [11] separated deliberate and emergent strategies and defined entrepreneurial strategy to be relatively emergent but able to emerge depending on the entrepreneur. Emergent strategy formation is therefore closely linked to business model experimentation, which is one distinct way of doing business model innovation. McGrath [6] sees experimentation as a way to discover which are the most effective models of allocating resources in the market, considering the constraints that are set by the competitive environment. Dunford et al. [12] see experimentation as one of the four processes in business model replication of an internationalizing multinational company. Companies conduct business model experimentation in most cases only after external innovations have disrupted their existing business model, because there are several barriers especially in large companies for creating experiments [13]. Many startup companies have utilized business model experimentation using the specific lean startup method, which originates from Steve Blank’s Customer development methodology [14] and was made popular by Eric Ries with his book *The Lean Startup* [1]. The lean startup methodology is based on validated learning where every action a startup does that does not increase learning how its products can serve customers better is considered waste. In addition to startups, also larger companies have started using the lean startup method for boosting their internal startup activities.

2.2 Technical Debt

The concept of technical debt was introduced by Cunningham as a metaphor to financial debt: “every minute spent on not-quite-right code counts as interest on debt. Entire engineering organizations can be brought to a stand-still under the debt load of an unconsolidated implementation, object-oriented or otherwise” [2]. Technical debt has recently become widely used for describing all shortcuts and workarounds in software development processes and artifacts though it was initially used for coding only [15]. As a result, there is a number of corresponding terms to describe shortcuts and workarounds related to other than coding processes and artifacts like quality debt, testing debt, documentation debt [15]. These types of technical debt are considered as subtypes of technical debt but their distinctive characteristics has not been established [16]. Therefore, this article uses the term technical debt to refer to any type of debt taken in the process of developing a minimal viable product.

In general, technical debt is an action or plan to circumvent a problem without developing a proper solution to solve it [17]. This is often done through developing a quick fix that is supposed to be replaced with a proper solution later but it is never done in practice. The temporary solutions that can be implemented in a shorter time in comparison with proper solutions provide companies with a competitive advantage to release new products to the market faster than their competitors. In a longer perspective, temporary solutions accumulate over time having a negative impact to the code-base maintainability [15].

The development of a minimal viable product done in startup companies or special internal startup departments of large companies through corporate venturing and experimentation requires the generation and testing of numerous ideas [18]. However, only a few ideas can potentially generate significant revenue to the company. The selection of the ideas for implementation is often done through experimentation by developing a product that is not fully functional but has primary features partially implemented for testing the product in the market [19]. The trade-off between releasing the product faster and having features properly implemented requires a company to take technical debt. By accepting that time-to-market is more critical than code quality, the company incurs intentional technical debt according to the McConnell's taxonomy [20]. In addition to intentional technical debt, any company is prone to unintentional technical debt. The sources of unintentional technical debt are out of control and the company can be even unaware of them. For example, it can be the result of significant changes in the product architecture that were not planned in advance but suddenly became essential for the product success in the market.

Overall, intentional and unintentional technical debt contribute to uncertainty of the environment in which the company operates [21] by setting limitations on features that can be implemented and time required for their implementation. Finding the right balance between time-to-market and amount of technical debt accumulated in the product can be seen a success factor of experimenting with various ideas and delivering these ideas to the market in forms of products that provide value to the customer.

3 Research Methodology

The study began with a literature review on business model experimentation and technical debt. Based on the literature, we argue that the current knowledge about the relationship between technical debt and business model is not well-studied and requires more examination. Therefore, this study is exploratory in nature and the goal is to find the linkages between the constructs and understand the relationship. We decided to use case study as the research methodology. We conducted multiple inductive case-studies with semi-structured interviews to gather data from the companies' representatives. Semi-structured interviews can provide rich and detailed data for a specific research question. Interviews bring forth the respondents' own perspective and provide insight to particular experiences they have had with the topic [22].

The cases selected for this study were three large companies in different fields of business and one small startup. From one of the large companies multiple informants

were interviewed to ensure the understanding of the whole phenomena of conducting business model experimentation and the effect on technical debt. In other companies only one informant was interviewed in a company. These interviews were used to confirm the findings in the first company with multiple interviewees.

The data collection was initiated with the large company that had multiple informants. They are studying technical debt in their own processes quite closely and the idea of investigating the relationship between technical debt and business model experimentation came up in discussions with this first case company. The research questions were drawn from those discussions and more informants were selected to increase the understanding in this company. In order to validate the findings, other companies were needed to be interviewed. The initial large case company is in the software development industry. The three other companies were chosen to represent other industries and company sizes; one large media company, one medium-sized software consultancy and one startup in software services business. Interviews with representatives from these companies enforced and proved the findings made in interviews with the first large case company.

The fact that there are six informants in one case company and one from each of the remaining three companies is a limitation in this paper. A wider selection of informants from the other three companies would have validated the findings more soundly.

The informants were experts in the particular area in companies. The interviews were semi-structured and conducted in November-December 2014. The duration of the interviews varied from 28 minutes to 52 minutes. In total there were nine interviews. The roles of the interviewees are shown in Table 1.

We analyzed the interview data with Atlas.ti software by making a thematic analysis, concentrating on the aspects related to technical debt and business model experimentation and identifying elements that played a role in their relationship. In the analysis, the following elements emerged: intentional and unintentional technical debt, the amount of focus on business model experimentation, emphasis on product quality and competence of the development team. As this was not a cross-case analysis trying to identify and examine the possible company-specific differences in the relationships between technical debt and business model experimentation, we present our findings by discussing the results on the level of the phenomenon itself.

Table 1. The roles of the interviewees

ID	Company	Role
A1	A	Test manager / project manager
A2	A	Project owner
A3	A	Technical coordinator
A4	A	Software developer
A5	A	Software developer
A6	A	Lead developer
B1	B	Development manager
C1	C	Managing partner
D1	D	Chief executive officer

4 Results

4.1 Finding 1: Business Model Experimentation Creates and Requires Intentional Technical Debt

We were able to identify a clear relationship between business model experimentation and technical debt. The studied companies used often a lean methodology and experimentation to build new idea, feature or service in iterative cycle with a minimal effort to product quality to receive faster feedback from the customer. The companies' goal was to test the assumptions of the current business model by experimenting the idea first at the customer before the actual development. To have minimal effort to the quality and fast feedback cycle, the development team had to take shortcuts and workarounds to produce a simple demo or prototype for the customers to use. This demo or prototype consisted only the most minimal amount of source code necessary and sometimes they were just graphical presentations done on the paper to demonstrate the possible functionality in the real version.

"We have done this product in few iterative steps and always tried to produce the minimal amount to validate the next steps and hypotheses. This has worked for us really well and we have gone always one step forward, but on the other hand we have accumulated technical debt there during that." – B1.

When companies got an idea to improve the current business model by creating a new feature or a service, the assumption that it would improve the current business model needed to be validated with an experimentation before the actual development phase could start. The companies did not want to waste time and money to first build something and realize afterwards that the assumption of beneficial feature or service was not correct. The reason was that it would have resulted to a significant loss in the development time, because the feature or service would not have been valuable to the customer and therefore to companies' new business models. This was the reason why the case companies first created a demo or a prototype from the idea and experimented it at the customer to receive a fast feedback that would help the company to make the decision for further development.

"Every thought, idea, or a single feature in the product that you have in mind must be validated somehow before you start to implement it. Otherwise you could use valuable time to build something that does not necessarily have value." – B1.

The demo or prototype created by the companies were usually developed as fast as possible with minimal amount of source code. At this stage companies made a decision to intentionally take technical debt to the product, as the quality of the feature was really low compared to what it should have to be in the future if the experimentation turned out to be successful. This resulted to situations where a company gave the customer a demo or a prototype of the feature that had a lot of usability issues and bugs, but that would still somehow demonstrate the main functionality that the company assumed would make customers interested.

“The goal is not to code everything when you have an idea. For example we had a lot of weird usability issues in the prototypes we had this summer, or actually in the beta version, but we decided not to fix them. It is in the accordance of MVP method that you must be little bit ashamed of your product that is going to customer first time.” – B1.

The opinion of most interviewees was that technical debt is bad for the company and product, because it starts to hurt overall quality and it is challenging to manage. However, one informant thought that taking technical debt is not necessarily a bad thing to do in the beginning of the business model experimentation. The reason was that when companies are looking for the correct business model, it does not matter if technical debt keeps accumulating, because the goal is to find the correct business instead of developing something that does not have value to the business model. It would be easy for companies to just throw away the demo or prototype consisting of technical debt, if it would not be good part of the business model.

“I think that in the beginning start-up does not have to be worried about technical debt, because at that point you have not even validated if your idea good and does it grow to actual business. So technical debt at that point... just get features released and it might even be that the whole product will go to trashcan and also the technical debt at the same time. At that point let's just do something else.” – D1.

The results indicate that while business model experimentation was clearly creating intentional technical debt, it was also required to be taken. The goal of the business model experimentation was to acquire customer feedback as fast as possible to confirm the assumptions made in the business model. This is the reason why companies had to take technical debt intentionally. It made the customer feedback cycle much faster and hypothetically decreased the possibility of unintentional technical debt as the next software development steps were validated with customer.

4.2 Finding 2: Development without Business Model Experimentation can Create Unintentional Technical Debt

Business model experimentation has also a relationship to unintentional technical debt. The interviewees described situations where the companies did not use business model experimentation as a tool to develop the business model. Instead, when companies got an idea to improve the business model with a new feature or a service, the software development was begun immediately without conducting customer validation first. We were able to see scenarios where the new ideas were successful without experimentation and the companies were able to improve the business model. However, we also saw scenarios where the idea got developed and after the release the company realized that customers had no need for that certain feature or service.

“When you think portfolio companies we have worked with that have not used any iterative development of business model, instead they have just gone after some big idea, they have also made huge mis-steps in their technology.” – C1.

The reason for not to use experimentation was that the new idea appeared to be so good that the company decided to begin the development immediately. In addition, experimentation was seen as time consuming and expensive practice to do that could give competitor an edge to be first on the market. Instead, the company could just develop the feature instantly without losing any time while trying to get the feedback from the customers. One of the interviewees also mentioned that reason not to conduct experimentation was that customers were not always willing to take part to the experimentations, since the customer might not be interested in intermediate results.

“Sometimes it happens like that but not all the time customers are actually Interested in the intermediate results, so sometimes they don’t want to be involved in that cycle. They just want the feature because they have a business need for it and they think everything is clear and it should be just implemented.” – A3.

Sometimes companies go after a big idea and start the development instantly without first conducting customer validation through experimentation. These are examples where companies can incur technical debt to the product unintentionally. Even if the new idea would be developed really well with good scalability for the future ideas, if the idea does not fit to the current business model and the customer does not have any need for it, the unnecessary time used for the development can be seen as technical debt.

“Actually you could say that if we would now put a lot of effort and development to the idea we think is good and would develop it really well, we would not make a lot of technical debt. But actually if the business model would be wrong at that point, we would great a huge amount of technical debt.” – A6.

4.3 Finding 3: Both Intentional and Unintentional Technical Debt can be Reduced with Business Model Experimentation

Business model experimentation can cause accumulation of technical debt because the goal of lean startup methodology and business model experimentation is to create a viable product with minimum effort. It requires shortcuts and workarounds in the development that is considered technical debt. However, business model experimentation can reduce both intentional and unintentional technical debt if used properly. We were able to identify situations where the business model experimentation was used to reduce intentional technical debt and to prevent unintentional technical debt.

The reason for the reduction of intentional technical debt was the customer feedback, which was acquired through business model experimentation that gave companies information how to prioritize the developed components in the product. With customer feedback, the companies were able see what was the most important for customers and were able to reduce previously intentionally taken technical debt from those areas.

The benefit of lean startup methodology and business model experimentation was the identification of wrong assumptions in the business model early and avoid wasting developer time on matters that customer’s do not need or want. In these cases there is

a possibility for a quick adaptation based on customer feedback. If the company learned that some feature did not have any business value, it was easy to just throw that part of the product to the trash without having a huge damage, since the solution was done already with major shortcuts and it would in any case have required refactoring and rewriting.

“I think that if we move forward by doing demos it is a good thing. When we have like weekly sprints, it does not matter if we go to wrong direction, we have only lost that one week by then, and sometimes not even that much.” – A4.

“On the other hand we have thrown so much stuff to the trash can that we developed really fast previously and they should have been refactored, but we did not need them anymore because they were not important to customer.” – D1.

Business model experimentation was also used to prevent unintentional technical debt. One of the interviewees explained us a situation that happened when a team had a great new business idea. One of the managers in the company assumed that the feature was so brilliant that there was no need for experimentation and customer feedback before development. However, the lean startup team insisted on gathering customer feedback to confirm the assumptions. The result was that the majority of customers thought the feature was useless and there was no need for it.

“So we had this good idea and we had little time to do the experiment design. But one of the managers was like “well I think that this is not necessary because it is so good idea”. Anyways a team went to interview 20-30 customers and when they came back they said “Dammit, no one was interested, people thought it sucks.” – B1.

In this case, by conducting the experimentation, the company was able to prevent unnecessary work and technical debt from happening. If the company would have skipped the experimentation and started to develop the feature, the amount of technical debt would have been huge, since all the work of the developers would have gone to waste and company would have not needed that feature in the business. However, now the company was able to prove that the assumptions of the current business model were wrong and it got valuable customer feedback to not develop the feature.

4.4 Finding 4: Focusing Too much on Business Model Experimentation and not on Technical Debt Reduction can have Consequences to the Product Quality

Business model experimentation is a great way for companies to receive fast customer feedback and to realize how to improve or change the current business model and the product. However, it can also create some challenging consequences in a long-term. We were able to identify some long-term problems that the case companies were facing when using the business model experimentation. The biggest challenge was the balance between developing new features and improving already existing features. Some of the interviewees felt that the business model experimentation is creating too much pressure to the development team and it is hard to improve features already consisting technical debt, because there is all the time a need for new features and

prototypes demanded by customers. It can be argued that this has not been business model experimentation in the same sense that the lean startup method suggests, however.

“That is the problem because you also get a lot of features requested by the product line, and the problem is because they actually set deadlines on them. The thing is that those deadlines are not even related to the release window that we have. Although writing the code is quite easy, getting it in requires this downtime cycle. The downtime cycle is the biggest legacy or technical debt that we have. So architectural decisions have been made based on our customer and those decisions are killing us.” – A3.

The consequence of continuing business model experimentation instead of paying technical debt back in already existing product was that the code base started to become too complex and challenging for further development. This resulted to slowness, breakdowns, bugs and scalability problems and the companies had to conduct a lot of refactoring and rewriting to fix the issues.

“Yes it is really complex at the moment and you really do not know what happens if you change some part of the code. Another problem is the scalability issues that is currently really weak. So we have had discussions that should we write this again.” – A3.

“For example we talk now a lot about architecture because we just got three new developers and they told us that the product is slow and when you change something you will break something else. The team and product is getting bigger, so we must have some process to get technical debt in control, because otherwise nothing gets developed anymore.” – D1.

The balance between business model experimentation and technical debt reduction is something that companies need to improve in the future. However, it is challenging because the competitive business environment forces companies to constantly improve and change their business model to gain advantages over competition. When the majority of company's focus goes into finding new business model possibilities through a series of experimentations, the focus on technical debt decreases and that can have consequences to the product quality.

5 Discussion

When combining the experiences and examples described by the interviewees, we can see that the growth of the business and product quality were connected with business model experimentation, reduction of technical debt and competence of the development team. We were able to see that companies had two ways to test their current business model and its assumptions. The first one was to develop the idea with a good design and scalability and release it to a customer when it was ready. We saw situations where companies developed the idea with a good design and then the release was a success. However, we could also identify cases where the well-designed new

features were not that successful. The reason was in most cases wrong assumptions about the actual customer needs. According to McGrath [6] business model cannot be fully anticipated in advance and it should be rather learned through experimentation in discovery and development.

The second way was to test the business model with experimentation. There the companies figured out the minimal way to experiment with the customer if the assumptions were right or wrong before even starting the actual development. When a company had a clear vision about the business model and all the assumptions were confirmed, the company started to improve the feature that was previously developed with shortcuts for experimentation purposes. In these situations the overall development time was often longer and more expensive, since companies had to conduct series of experimentations before starting the development. Chesbrough [23] claims that some companies do not use business model experimentation, because it is time-consuming to create, conduct, obtain, interpret and understand the experimentations. This is why some companies prefer to just grow the current business model [23]. However, the experimentations conducted in studied cases helped a company to find the correct business model instead of using the wrong one. Most of the interviewees thought that even though using experimentations might take a longer time to create and release the features to the customer, it is still a better way to grow the business and create a good-quality product.

Another factor for a business to grow and create a quality product is the competence of the development team [24]. The use of a lean methodology and business model experimentation required a lot of competence to experiment and develop features in fast iterative cycles with a product in minimum viable state. When the development team had to work with the code base that had incurred already technical debt during the experimentation, it required a lot of experience and knowledge to be able to create solutions that have high quality and scalability, when the business model is evolving in the future.

Having a growing business and quality product can also depend on the reduction of technical debt. The companies in this case study were eager to make experimentations and try out demos and prototypes in fast phase to find out possible new business ideas and areas to great more successful business. However, when companies had a high focus on creating new businesses and features to answer to the demand of customer, the focus on improving existing features and reducing technical debt was low. The improvement and refactoring of existing code is important part of product overall quality [25,26]. We were able to identify situations where technical debt started to affect to the success of business and product quality. Sometimes there were situations where too much technical debt started to show as slowness and bug errors in the product. The quality of the product has a strong relationship with the customer satisfaction [27]. The problems in the product could transfer to negative customer satisfaction that can have consequences to the business of the company. At this stage companies had to start massive operation to refactor and rewrite parts of the product, which led to significant economic costs.

6 Conclusion

This paper has explored the relationship of business model experimentation and technical debt in the context of software development. Our analysis reveals that technical debt should be divided into intentional and unintentional in this context, and that product quality and the competence of the development team are elements that need to be considered. The overall result is that with business model experimentation, the amount of technical debt can be reduced. However, there may be an inverted U-shaped curve concerning the benefits of business model experimentation – it is a balancing act to do enough experimentation but not too extensively, and simultaneously pay careful attention on the amount of accumulating technical debt. The targets of experiments must be well-chosen and the competence of the development team sets pragmatic limitations on the amount of experiments that can be executed with a reasonable time-to-market goal. Further research could compare and measure both the amounts of technical debt and business model experimentation in specific projects and compare the levels to the success of the products and business model launch to learn more about the interrelationships of these constructs. As a limitation, this research mainly used informants from R & D. To get a more complete picture of this phenomenon, also marketing and product managers' viewpoints could be incorporated in the analysis more strongly.

Acknowledgement. We would like to thank the companies for being a valuable source of information and all interviewees who found time to share their experiences. This research has been carried out in Digile Need for Speed program, and funded by Tekes (the Finnish Funding Agency for Technology and Innovation).

References

1. Ries, E.: *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, 1st edn. Crown Business, New York (2011)
2. Cunningham, W.: *The WyCash Portfolio Management System, Experience Report* (1992)
3. Teece, D.J.: Business Models, Business Strategy and Innovation. *Long Range Planning* 43(2-3), 172–194 (2010)
4. Zott, C., Amit, R.: Business Model Design: An Activity System Perspective. *Long Range Planning* 43(2-3), 216–226 (2010)
5. Casadesus-Masanell, R., Ricart, J.E.: From Strategy to Business Models and onto Tactics. *Long Range Planning* 43(2-3), 195–215 (2010)
6. McGrath, R.G.: Business Models: A Discovery Driven Approach. *Long Range Planning* 43(2-3), 247–261 (2010)
7. Doz, Y.L., Kosonen, M.: Embedding Strategic Agility: A Leadership Agenda for Accelerating Business Model Renewal. *Long Range Planning* 43(2-3), 370–382 (2010)
8. Bock, A.J., Opsahl, T., George, G., Gann, D.M.: The Effects of Culture and Structure on Strategic Flexibility during Business Model Innovation. *Journal of Management Studies* 49(2), 279–305 (2012)
9. Markides, C.: Disruptive Innovation: In Need of Better Theory*. *Journal of Product Innovation Management* 23(1), 19–25 (2006)

10. Casadesus-Masanell, R., Zhu, F.: Business model innovation and competitive imitation: The case of sponsor-based business models. *Strat. Mgmt. J.* 34(4), 464–482 (2013)
11. Mintzberg, H., Waters, J.A.: Of strategies, deliberate and emergent. *Strat. Mgmt. J.* 6(3), 257–272 (1985)
12. Dunford, R., Palmer, I., Benveniste, J.: Business Model Replication for Early and Rapid Internationalisation: The ING Direct Experience. *Long Range Planning* 43(5-6), 655–674 (2010)
13. Chesbrough, H.: Business Model Innovation: Opportunities and Barriers. *Long Range Planning* 43(2-3), 354–363 (2010)
14. Blank, S.: *The Four Steps to the Epiphany*, 2nd edn. K&S Ranch (2013)
15. Yli-Huomo, J., Maglyas, A., Smolander, K.: The Sources and Approaches to Management of Technical Debt: A Case Study of Two Product Lines in a Middle-Size Finnish Software Company. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (eds.) *Product-Focused Software Process Improvement*, pp. 93–107. Springer International Publishing (2014)
16. Tom, E., Aurum, A., Vidgen, R.: An exploration of technical debt. *Journal of Systems and Software* 86(6), 1498–1516 (2013)
17. Halbesleben, J.R.B., Wakefield, D.S., Wakefield, B.J.: Work-arounds in health care settings: Literature review and research agenda. *Health Care Manage. Rev.* 33(1), 2–12 (2008)
18. Corbett, A., Covin, J.G., O'Connor, G.C., Tucci, C.L.: Corporate Entrepreneurship: State-of-the-Art Research and a Future Research Agenda. *J. Prod. Innov. Manag.* 30(5), 812–820 (2013)
19. C. ©. M. I. of Technology and 1977-2015 All rights reserved, “Why Companies Should Have Open Business Models. MIT Sloan Management Review
20. McConnell, S.: Technical Debt-10x Software Development | Construx (November 1, 2007), http://www.construx.com/10x_Software_Development/Technical_Debt/ (accessed: December 02, 2014)
21. Bourgeois III., L.J.: Strategic Goals, Perceived Uncertainty, and Economic Performance in Volatile Environments. *The Academy of Management Journal* 28(3), 548–573 (1985)
22. Hannabuss, S.: Research interviews. *New Library World* 97(5), 22–30 (1996)
23. Chesbrough, H.: Business model innovation: it’s not just about technology anymore. *Strategy & Leadership* 35(6), 12–17 (2007)
24. Chow, T., Cao, D.-B.: A survey study of critical success factors in agile software projects. *Journal of Systems and Software* 81(6), 961–971 (2008)
25. Mens, T., Tourwé, T.: A Survey of Software Refactoring. *IEEE Trans. Softw. Eng.* 30(2), 126–139 (2004)
26. Middleton, P., Joyce, D.: Lean Software Management: BBC Worldwide Case Study. *IEEE Transactions on Engineering Management* 59(1), 20–32 (2012)
27. Jun, M., Yang, Z., Kim, D.: Customers’ perceptions of online retailing service quality and their satisfaction. *Int. J. Qual. & Reliability Mgmt.* 21(8), 817–840 (2004)

Software Business

6th International Conference, ICSOB 2015, Braga,

Portugal, June 10-12, 2015, Proceedings

Fernandes, J.M.; Machado, R.J.; Wnuk, K. (Eds.)

2015, XVIII, 289 p. 39 illus., 1 illus. in color., Softcover

ISBN: 978-3-319-19592-6