Pivots and Architectural Decisions: Two Sides of the Same Medal?

What Architecture Research and Lean Startup can learn from Each Other

Jan Salvador van der Ven
University of Groningen
Groningen, the Netherlands
mail@jansalvador.nl

Jan Bosch
Chalmers University of Technology
Gothenburg, Sweden
jan.bosch@chalmers.se

***Abstract* — Software architecture research has gained maturity over the last decades. It focuses on architectural knowledge, documentation, the role of the architect and rationale for the architecture decisions made. It is widely recognized that considering architecture decisions as first class entities helps in designing and maintaining architectures. In the entrepreneurial and new product development space, the lean startup movement is gaining momentum as one of the most notable ways to develop products. During new product development in highly uncertain environments, speed is the most important factor. Speed to get on the market, speed to learn from your customers, but also speed to tackle technological risks. Because the runway for new product development is short, it is important to experiment and make decisions quickly. The pivot plays a crucial role as a business decision for new product development. Both pivots and architectural design decisions can be seen as highly influential aspects for a product. In our research, we investigate what the fields of architecture research and lean startup could learn from each other. We focus our research on the two most important aspects of these movements: the architecture decision and the pivot, and show that they can be seen as two sides of the same medal representing the technical and the business side of the product.**

Keywords—Pivot; Architectural Decision; New Product Development; Lean Startup; Software Architecture.

I. INTRODUCTION

Every company changes direction multiple times during its lifetime. In the past, it took a company months or even years to change direction, especially in larger industry settings. In the last decade, the speed in which a company can adapt to changes has become one of the most competitive qualities [1]. The place where this effect is amplified is in new product development, either in small startups or in larger, established companies. Because these projects typically have a short runway to being successful, making decisions quickly is crucial.

Architects have the important role to align business strategy to the software architecture of the products [2]. Especially in the domain of new product development, this balance is an enormous challenge, because on the one hand the time to market is essential, and on the other hand the continuation of product and company is dependent on the solidity of the architecture. In new product development, there is also a bootstrapping problem. You need experiments with the Minimal Viable Product (MVP) in order to be able to validate your business assumptions, while you also need to have a piece of architecture to be able to create this MVP. This tension exists in many projects involving new product development.

As a software company, one of the most important aspects of your product is the software architecture, as it highly influences the capabilities (quality attributes) of the product. This architecture is formed by the decisions made during the development and maintenance [3]. Various authors emphasize the importance of these architectural decisions in software development [4, 5]. Models [6], classifications [7] and reasoning structures [8] have been posed to manage these decisions. Key concepts that are used in software architecture are: decision topic, rational, alternatives, choice, and risk.

Research literature studying new product development and startups [9, 10, 11] identifies a key type of decision that is extensively (and explicitly) used, the *pivot*. A pivot is the result of a business decision that is made to change the direction of the product. These decisions are based on different kinds of implicit or explicit experiments [1], in order to validate hypotheses about the product, its users or its business case. For the research described in this paper, we investigated what kind of decisions these pivots are, and what the relationships between pivots and the architectural decisions are. We currently focus on the pivots made at startups, because:

- At a startup, the runway is short, so the evolution of the architecture of the system is very high. Effects of pivots and architectural decisions are visible very quickly, and have a very high effect on the company's success.
- Larger companies are adopting startup techniques [1] to increase their own time-to-market, especially for new product development. This makes our research relevant as learning for large companies seeking new product development.

The contribution of this paper is threefold. First, we introduce a conceptual framework for new product development as an experiment system with pivots and architecture decisions as first class entities. Second, we identify the key concepts for architecture research and new product development, and identify the gaps between them. Third, we provide guidelines for the two fields that describe what they could learn from each other, based on the conceptual model and the identified concepts from both fields.

This paper is organized as follows. First, we introduce our conceptual framework. Then, we sequentially describe the concepts of software architecture (Section III) and new product development (Section IV) from a research and a

ISBN: 978-1-61208-304-9

practical perspective. In these sections, the key concepts are identified. Then, we describe the differences and similarities of the two as analysis in Section V. Based on this we present our guidelines for both fields. This paper ends with related and future work and some concluding words.

II. Conceptual Model

The premise of this paper is that this experimentation, both in the business domain and the technical domain, is a critical technique to increase the chances of success in new product development. Based on our literature findings, we have constructed a conceptual framework for running new product business as a set of decisions. In Fig. 1, our conceptual framework is visualized. On the top, two essential risks are shown as input for the business: market risk and technology risk. Which risks are most important depends on the context: the problem addressed, the market, the competitors, the solution chosen, the technical possibilities, etc. Based on which risks are most eminent, hypotheses are formulated to reduce uncertainty of the associated risk. To test each hypothesis, one or more experiments are performed. These experiments can be explicit (e.g., conducting a planned usage test, running a Proof of Concept, predict usage statistics), or implicit (e.g., a coincidental encounter, different product use by end-users). Then, based on the results of the experiments, decisions are made for the direction of the product. These decisions steer the direction of the product and the associated business, affecting the market and/or the software architecture, and in the end the product itself. In new product development, pivots are illustrative examples of these decisions. Therefore, the naming of the decision types is based on the phases described by Maurya [11]. In the initial Problem / Solution (P/S) fit stage, the decisions don't affect the system at all, since there is typically no product yet. In the second, Product / Market (P/M) fit stage, the focus of the experiments is to validate the Minimal Viable Product. This can result in pivots that influence the business as well as the product. For these decisions, the market fit is the most important; so, the architectural impact is subordinate. In the following phase, assuming that the product / market fit is validated, still experiments need to be conducted to figure out how to scale the product when usage (e.g., number of users or usage per user) grows. Aside from direct business requirements, in each stage software architecture decisions need to be made, for example to support increasing scale, reduce technical debt or support an alternative use case after a solution pivot. This paper focuses on pivots as decisions that arise from experiments that affect the business as well as the architecture of a product.

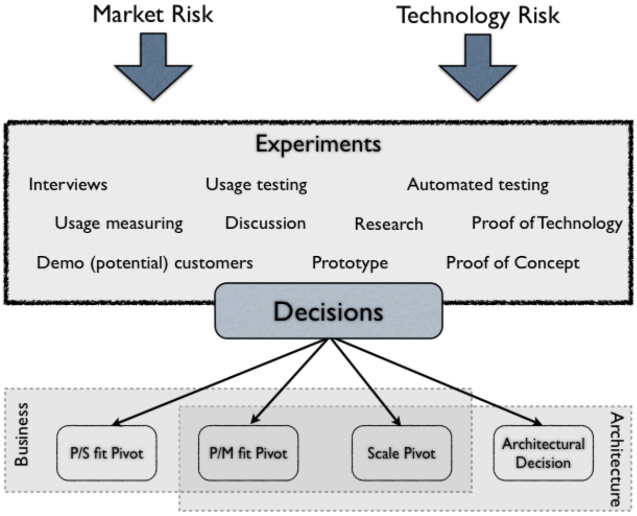

Fig. 1. Graphical representation of the conceptual framework for decision-based new product development.

The validation speed is very important in this context. Validating a hypothesis takes time and effort. This effort should result in new insights in the product or the market. If the product changes direction later (a pivot, or abandoning a pivot), the effort should pay itself by what is learned by it. So, it is important to keep validation speed short, and create hypothesis focused on learning. This is why validation speed it essential in our model.

When looking at product development through our conceptual model, it is possible to see that pivots and architectural decisions are actually the ways to mitigate risks by experimentation. However, they both have a different risk they are addressing, while affecting each other constantly. So, they can be seen as two sides of the same medal, one side showing the market challenges, while the other side shows the associated technological risks. It is virtually impossible to encounter one without the other, as market risks are typically tackled with technological solutions (e.g., the business drivers for the architecture), and technology always affects the business.

In the following sections, we will describe how this model can be used in both software architecture and new product development.

III. Software Architecture

A. *Software Architecture Research*

Software architecture has been researched extensively in the last decades [12, 13, 14]. In this research, architectural knowledge [6, 15] and more specifically architectural decisions [5, 16, 17] play a vital role. What we can distill from this research is that creating architectures is essentially a risk-mitigation process where the balance has to be found between non-functional requirements (e.g., quality attributes), business risks and technological challenges. Often the long-term view is more important then short-term project goals for making the right architectural decisions. In high-pressure situations (e.g., deadlines), it is easy to give in on these long-term issues, causing design erosion [18], technical debt [19] or even worse, project failure. In the next section, three cases are described that show how architecture decisions are used in practice. From this, we identify key concepts for comparing architectural decisions and pivots.

B. *Cases*

In order to be able to compare software architecture practices to lean startup movement, we have to identify what parts are eminent for both fields. To do this for the software architecture space, we have conducted a literature research combined with our experience as participant researchers in several cases. We have analyzed the practices of software architecture in new product development in several cases [7]. In this paper, we summarize the cases that contain relevant information about how software architecture is used in practice. The cases are anonymized to protect the companies and customers involved. The cases are not selected at random. From the experience of the authors, other cases could have been chosen. However, as Eisenhardt [20] poses, in case study research it is ”neither necessary nor preferable to randomly select cases”. We have chosen to discuss the cases that considered new product development, while being large enough to be relevant as industrial cases. A more extensive description of these cases can be found in our previous work [7], where we focused on the role of the architect in the software development process. In this work, we describe our findings of these cases that consider pivots and architectural decisions.

Case Alpha involved the construction of a software system that had to replace a legacy Geographic Information System (GIS) for a large harbor. The new system had to be coupled with several legacy backoffice systems. The customer, a large harbor company in the Netherlands, initiated the project. The solution was service oriented, and consisted of several systems communicating with each other through an Enterprise Service Bus. Most of the software was written in Java. The coupling was one of the most challenging issues in the project. This case consisted of a pilot and a realization phase, three and six months, respectively. Ten to twenty people were involved during the various phases of the project.

Case Alpha was a typical example of a project that was driven by risk management in order to get the architecture of the system right. Several techniques were used to experiment in order to mitigate risks. In the pilot phase, the time was fixed, and the goal was to show the most important (technical) risks could be tackled. This resulted in a biweekly iteration that focused on tackling the top-priority risk. In this phase, a PoT (Proof of Technology) and a PoC (Proof of Concept) were made, involving many architectural decisions. Both the PoT and the PoC were demonstrated to the customer as well as the end-users to validate critical assumptions.

Case Beta was conducted at a medium sized product company in the Netherlands. The project involved a new administrative software system for specific departments in Dutch hospitals. Changing regulations and different working environments needed to be taken into account. The project was executed by a multidisciplinary team of seven people, assisted by the architect from the company. A Java stack (JSF, Spring, Eclipselink) was used for creating this product from scratch, while a different team of approximately seven people developed a part of the backend separately. This separate development was one of the most challenging architectural parts of the project. The development of the product took place for a period of 12 months.

In case Beta, several architectural experiments were conducted, the major one consisting of how to manage the introduced complexity of the platform. A prototype was constructed early on. Also, interviews were held with key users in the field. However, often the experiments were conducted ad-hoc without a concrete hypothesis to validate. The architectural question if the generic backend part of the system could be reused was validated continuously by using this component in another project, too.

A small startup company working on a web based product for the consumer market was the scene for case Gamma. The project contained high-risk technological challenges, where the architecture needed to be flexible in the beginning, to be able to handle the expected high number of users. The application was created in Ruby on Rails[1] with a NoSQL backend based on MongoDB[2] and Redis[3]. The main architectural challenges were to be able to potentially scale up the application when lots of consumers are using the system, while being able to adopt the system to changing requirements from the customers.

Case Gamma consisted of constant experimentation. As the product of the company was being developed, several hypotheses were considered, resulting in either small pivots (e.g., users would like to see the results in a stream-like view), or architectural decisions (e.g., the graph database could be best modeled in Redis). However, again the experiments were setup implicitly, e.g., without forming a hypothesis or validating if the results were expected.

We have seen the experimental nature in all of these cases. Also, in all of the cases a clear Build, Measure, Learn (BML) loop [10] was used. In cases Alpha and Beta, this loop was used implicitly (never mentioned), while in case Gamma the BML loop was known and explicitly used.

C. Key concepts

Several key concepts come back in most of the research about architectural decisions [21]:

- **Architecture Design decision.** Design decisions are the building blocks for software architecture. These decisions consist of the following parts:
- **Decision topic.** The decision topic is the actual problem that needs to be solved. Often, these topics arise from previous decisions (we decided to base our application on NoSql technology, which specific database product are we going to use?), or from non-functional requirements (how are we going to ensure our up- time is high enough?)
- **Choice.** The choice, or decision, is the result of the decision process. Often, this is the only part that is communicated (discussed or documented).
- **Alternatives.** A typical decision has more than one alternative to chose from. Alternatives can be just named (e.g., different component names), or sometimes architecture parts are considered as alternatives (different styles or patterns, or comparing specific implementations of components). In rare cases, the alternatives are realized and compared as a Proof of Concept or Proof of Technology.
- **Rationale.** The rationale of a decision describes, often in plain text, why the chosen alternative(s) solve(s) the problem at hand, and why the chosen decision is the best solution.

Based on our case material, we have seen two other key concepts that are important around software architecture design decisions:

- **Risk.** Decisions are often made to mitigate a risk. So, in order to address a concrete market or technological risk, certain decisions need to be made. Risks can be seen as triggers for decision topics.

[1] http://rubyonrails.org/

[2] http://www.mongodb.org/

[3] http://redis.io/

- **Experimentation.** To make sure you make the right decisions often, besides the rationale already discussed, experiments are conducted to make viable that the suggested solution is correct. This can be done either as a PoT, PoC or something else.

In the following section, we will describe what the nature of new product development is and how the lean startup movement influences it.

IV. New Product Development

A. Research

Experimentation in Research and Development (R&D) as a basis for decision-making is the normal approach in a variety of domains, including the manufacturing, automotive, mechanical engineering, medical, and pharmaceutical industry [22]. From the experiential perspective, frequent iterations of products in terms of prototypes or multiple design iterations, testing, and more frequent milestones are associated with faster product development [23]. In the software industry, innovation through experiments with customers is becoming more and more discussed [24], primarily in the web 2.0 and Software as a Service (SaaS) fields. However, in the software industry, these experiments are currently primarily performed in pilot stages for validating architectural decisions or on feature optimization.

B. Interview Setup

We have conducted interviews with founders and architects of startup companies, to identify what pivots were made in new product development, and what the nature was of these decisions. In our interviews, we have chosen to focus on pivots as an entrance to talk about the most important decisions and the decision process. We interviewed representatives of the five different companies. In these interviews, we discussed a total of nine pivots. Two of the companies were located in the Netherlands, two in the USA, and one in Sweden. All the companies were product companies, delivering web-based software.

As our research has an exploratory nature, we have chosen to use semi-structured interviews for acquiring our data. The interviews lasted from one to two hours. We have recorded all of the interviews to be able to listen again to the conversations during the analyses phase. In addition to this, the interviewer made notes during the interview. Based on the notes and the recordings a log is created with results after the interview. These logs were the basis for our analyses.

The interviews were structured as follows. First an introduction was given about the current status and the goal of the research. The interviewee was asked permission to publish about the results and if it was okay that the interview was recorded. Then general questions about the company and terminology was asked, after which the interviewee was asked to tell about several pivots he was involved in. The interviews in the Netherlands were done face-to-face in Dutch, while the interviews with Sweden and the USA were done via videoconference in English.

We have used interview questions as guidance through our open-ended interviews. First, basic questions about the interviewee and company were asked, including if the company worked according to lean startups principles and if the architecture of the system was considered explicitly. In order to relate the results of the different interviewees to each other, we have asked them to describe what they mean by three key terms in our research: *pivot*, *architecture*, and *architecture decision*. Then, we used a set of questions to let the interviewees reason about the their pivot. As we wanted to focus on the decision process around pivots, we have not extensively questioned the technical details, but focused on the decision part of the pivots. The interview questions that were used are shown in Table I.

TABLE I. Interview Questions

Question
Can you give a short description of the pivot?
Who were involved in the decision process?
What triggered the pivot?
Did you validate the success / results of the pivot? How did you do that? How long did it take to do this validation?
Were there any alternatives evaluated? If so, what alternatives?
What were the results of the pivot?
Did the pivot affect the (software) architecture of your system / product?
What were the results on the architecture?

These questions were used as a baseline for the interview. Where viable, additional questions were asked, or explanation was asked for. In some cases, when the answer to a question was already told or when the question was irrelevant for the context, the question was skipped and later noted based on the recordings and notes.

For our research to be generic, we have selected a variety of interviewees and companies. On the other hand, we had to narrow our research in order to make sure the interview results would be comparable. We used the following criteria for selecting the companies:

- Companies from software industry in the startup phase, or a close startup origin.
- Companies at least one year in business at the time of the discussed pivot(s).
- Companies that produce a product or service (no consulting).
- Companies with more than one employee.

TABLE II. Overview of companies

Company	Location	Role	Domain	Size
Voys	NLD	Founder	Voice over IP, telecom for small business	~23
Certive	USA	Lead Engineer	Enterprise analytics software	~20
Data-provider	NLD	Founder	Data	~10
Burt	SWE	Chief Architect	Analytics for publishers	~28
Zevents	USA	Lead Engineer	Local search advertising	~50

This resulted in the selection of a set of 5 companies, as shown in the Table II. In the columns, the Company name, the

TABLE III. OVERVIEW OF PIVOTS

#	Comp.	Pivot / Decision	Prioritized risk (type)	Experiments and validation	Alternatives evaluated	Results on architecture
P1	Voys	Business model change	Unknown	Accidently showing internally used functionality to a customer.	None	The architecture became more of a 'Christmas tree'
P2	Voys	Architecture reconstruction	Maintainability decrease	Technological exploration	1) Buy functionality from other suppliers and 2) merging with other company	Reworked architecture, the system was now manageably growing
P3	Voys	Change of product packaging	Customers misused the product	Usage testing and measuring	None	Unknown (currently in progress)
P4	Certive	Radical change in business	Unknown	Demonstrating a mock-up to potential customers at a conference	Unknown	Moved more to hosted and cloud-based services
P5	Data-provider	Scaling the indexing possibilities	Technical possibility to scale product	Technological pilots, automated performance validation	All different kinds of NoSql solutions were evaluated	Possible to index sites at a high speed.
P6	Data-provider	Enhance defect efficiency	Data not accurate enough	Usage Measuring and experimentation at customer site	1) External provider for data and 2) buying data from others	Not much, the major change is in the way the application was used (the customer can decide the error rate)
P7	Burt	Change of customers from advertisers to publishers	Advertiser market is uncertain business	Usage measuring and discussion	1) Stay on advertisers and 2) move to both publishers and advertisers	Better distributed scalable architecture. Many principles were decided on (e.g., start with two on anything)
P8	Burt	Change in product from advertiser tool to analysis tool for advertisers	Customers are not able to judge the market value of product	Prototype, Demonstrate to potential customers	Several prototypes of different ideas were tried	Change from desktop to web based platform
P9	Zevents	Change in focus on search instead of publisher oriented site	Business of publisher sites was going down.	Discussion, prototypes	Lot of discussion about other alternatives tool place. One alternative was offering 'deals' to for local companies.	Architecture and tooling became more 'generic', making it harder for the company to distinguish itself against others.

geographical location, the role of the interviewee, the domain of the company and the size of the company (number of employees) is described. From each of the companies, we interviewed one of the key persons involved in the pivot(s) that occurred.

C. Interview results

First, we had to identify our interviewees' point of reference. To do this, we asked them about what three key terms in this research mean to them.

- **Pivot.** Even thought the term pivot is widely used in software industry, there was some difference in the explanations about what a pivot is. Two points came back in all interviews: that it is a radical interruption against the 'previous' way of working/thinking and that often, different users/customers were targeted after a pivot. So, the business strategy of a company changed. One person emphasized that layoffs are often the result of a pivot, making it 'scary' for employees when a pivot occurs.
- **Software Architecture.** The traditional view on architecture was dominant at the interviewees. All of them identified connectors/interfaces as one of the most important parts of architecture. Also, the mapping of business (requirements) on the technical design of the system was mentioned often.
- **Architectural Design Decision:** One of the interviewees had no idea what an architectural decision meant. The others noted that it is a conscious decision, where a specific direction is chosen for the architecture of a system (a branch-point).

We have summarized the results from the interview in Table III. In this table, after the name of the company and a short description of the pivot, the risk that was tackled by the pivot is described. The next column describes what experiments were conducted to validate the pivot. This information was derived from what the interviewees discussed based on the interview questions (e.g., the trigger for the pivot and the alternatives evaluated). Then, evaluated alternatives are shown, and in the last column of the table the results on the architecture are described.

Although Ries [10] identifies ten different types of pivots, he does not discuss the effects that pivots have on the architecture. From our interviews we have found that it is possible to typify pivots by the impact they have on the architecture, as described in our conceptual framework. Business (product/market fit) pivots were found in six of the pivots and scale pivots were identified in three of the pivots. Although all interviewees stressed the fast-paced, dynamic and uncertain nature of new product development, the importance of employing a structured, systematic approach to decision making was recognized as important.

D. Key Concepts

The following key Concepts involving new product development are extended from the literature:

- **BML / Experiment.** The basis of the lean startup lies in the Build Measure Learn (BML) loop, as described by Ries in [10]. This means that in order to find a sustainable business, one has to continuously execute experiments (build), measure the effects, and learn from the results.
- **MVP.** The Minimal Viable Product (MVP) is the first version of the product that can be used to start the BML loop. This can be a first version of a product, but it can also be something else (e.g., a landing page, video) as long as the hypotheses about the product can be validated.
- **Hypotheses.** In order to be able to know if one goes in the right direction, you have to know where you want to go. This is posed in a hypothesis that can be tested by experimentation.
- **Validation.** Key to understanding the results of a build step is to identify how to validate or invalidate a hypothesis.
- **Measuring.** Even though validation is concerned one of the most important parts of the BML loop, the measuring is always an arduous part. Measuring can be done either qualitative (e.g., interviews), or quantitative (surveys, usage measuring, A/B testing).
- **Pivot.** A pivot is a key concept in the lean startup movement as a decision to change direction for a product. Several types of pivots have been identified by Ries [10].

Based on the interviews, an additional concept comes back:

- **Risk.** Most of the pivots that were discussed in the interviews mentioned that they were done in order to mitigate some risk. The identification of this risk was often the starting point for the pivot.

V. ANALYSIS

In this section, we summarize what similarities and differences are between the architecture research space and the startup spaces, by comparing the most characteristics aspects of both: architectural decisions and pivots. The introduced concepts of both software architecture and lean startup / new product development are compared in Table IV.

One of the biggest differences is the focus. As the architecture community focuses on long-term non-functional requirements, the lean startup community focuses on rapid validation of business assumptions (hypotheses). This also has a cost implication. For lean startups, the speed of validation is the most important aspect. So, the experiments should be as fast and cost-efficient as possible, to be able to change direction quickly if market or technology demands that. This contrasts the approach of the architecture community where the focus is much more on making correct decisions to reduce cost later in the development.

Several parts come back in both worlds. Both consider risks as primary triggers for making a decision, and both have an explicit description of what needs to be solved, the decision topic and the hypothesis. Further, both parts use experimentation to see if the decision is correct, even though these experiments have different forms. The minimal version to validate your decision is correct also comes in different forms, in architecture this is often a technological proof while in new product development this typically involves customers and end-users.

Further, as can be seen from the table, several concepts from one field seem to be nonexistent in the other field. The

TABLE IV. COMPARING CONCEPTS

Architecture Decision Concept	Lean Startup Concept	Software Architecture	New Product Development
Architectural Design Decision	-	First class entity for the architecture	-
-	Pivot	-	Radical change in business model
Decision topic	Hypotheses	Decision topics are typically hierarchical (caused by previous decisions), or caused by arising or expected risks.	
Choice	-	Often referred to as the decision self, this is the selection of the best alternative	The choice is not explicitly mentioned in new product development space.
Alternatives	-	Are often made explicit in documentation	Alternatives are rarely made explicit.
Rationale	-	Existing in the heads of the developers, or (ideally) written down explicitly	Less relevant as the results are measured quickly.
Risk	Risk	Often the focus is on technological risks. Is addressed by reasoning, often the cause of an decision topic and thus a design decision	Focus is on the business risks. Is addressed by experimentation
Experimenation	BML / Experimentation	Automated testing (e.g., performance tests), Research, Discussion	Interviews, Usage measuring, Demonstration, Discussion, Prototyping, Research, Usage testing
PoC / PoT	Minimal Viable Product	In order do address certain risks, PoCs or PoTs are conducted. Main goal is to validate the viability of the concept or the technology, not the business	One of the main goals for a product under development. Main goal is to start validating the business model as quick as possible.
-	Measuring	Rarely done	Measuring is the only way to validate the hypotheses
-	Validation	Is often not done, if it was done, it was done by reasoning. It is often hard to validate a NFR	Direct business validation. Often the existence of company validates pivot.

explicit parts of the decision in the software architecture field (Choice, Alternatives, Rationale) do not exist in the Lean startup field. Alternatives are evaluated (as seen in the interviews) and rationale is used to argument decisions or pivots, but decisions as first class entities are not common in the lean startup field. On the other side, the measuring and validation that is key in the lean startup is not considered in the architecture space.

A. Threats to Validity

This research is based on a limited set of cases and interviews. To a certain extent interviews bare some subjectivity in them, because it is a conversation between two individuals. Because of the exploratory nature of our research, using semi-structured interviews was a good way to validate our model. However, this research could be extended by more interviews, and by gathering more quantitative data based on surveys, as described in the future work.

For interview validity reasons, we have not presented our framework or model to our interviewees. This would have biased our interviewees, and perhaps changed the way they described the pivots, and answered the questions.

VI. GUIDELINES

In addition to confirming the conceptual framework, the data presented in this paper allowed us to derive a set of guidelines about what the field of software architecture and new product development could learn from each other.

A. Solve both business and architecture as experiments

For new product development, explicit experimentation is common. Architects can learn from this by doing similar explicit experiments to validate the architectural decisions at hand. This helps architects to speed up development and develop business quicker.

B. Business as a set of decisions

As shown in our conceptual model new product development can be treated as an iterative process of running market and technology experiments. The experiments are driven by the risks that need to be tackled, and the result of the experiments is a set of decisions that form the business and the product. As we have shown that an architecture can be seen as a set of decisions, we think this view can be extended when considering pivots as business decisions. In this view, the business can actually be seen as the set of taken decisions based on the results of experiments.

By making the decisions in new product development more explicit, it is possible to piggyback on the experience that the software architecture research already developed. It can for example be used to trace the decision process, change decisions when the situation changes, and see the dependencies that decisions have on each other.

C. Creative validation of architectural decisions

Even though some efforts are made to validate architectural decisions, the field of software architecture could benefit much from the creative way that lean startups validate their hypotheses. Of course, the horizon for both decisions is not always the same, but the tendency to validate an architectural decision by reasoning could be enhanced by more objective ways of validation (e.g., usage statistics, A/B testing).

D. Sometimes, architecture can be added later

We have seen that in highly uncertain environments pivots affect the balance in the development of new products. Since p/m pivots put the emphasis on validating the business, the architecture of the product is often minimal supported. This can cause design erosion and technical debt. However, we have seen that there are several strategies used at our investigated companies to overcome this:

- **Pivot away.** The first strategy we identified was that in some cases the pivot was so radical, that the current architecture was thrown away. So, no matter how unbalanced the scale was, the complete business changed and the complete architecture of the system changed too. Off course the experience of the team and the business knowledge is reused, but the system itself was largely or completely rebuild. Sometimes a complete new technology stack was adopted (P2, P4, P8), while in other cases existing components were reused (P1, P5, P7).
- **Add architecture later.** When a product/market fit is found, but the architecture of the system is unable to facilitate the next phase (scale, as described in [11]), then architecture needs to be added later. So, in order to handle certain (non-functional) requirements for scaling, like performance or changeability, the architecture of the system need to be improved. As we have seen in our interviews (P2, P5), this is possible even though it can be expensive.

VII. RELATED WORK

Although the field of new product development is not new, lean startup is quite new, and within the research community there has not been much research about this topic. The basis for our model, experimentation, lies in the work of Thomke [25] and Davenport [24]. This was extended with the methodologies from the lean startup community [9, 10, 11]. From our own work on architectural design decisions, we generalized the idea of running a business as an explicit set of decisions [7], based on the experiments [1].

The relationship between business and architecture has been extensively studied from the product line perspective, for example BAPO [2]. We have shown that two types of decisions are extremely important in new product development: business (e.g., pivot) and architecture decisions.

VIII. FUTURE WORK

Based on the encouraging results from our research, we are planning to extend it in several ways. First, we are planning to interview more people, to extend our data set and further validate and refine our findings. For example, we have not had any of the interviewees talk about hypotheses, even though the literature emphasizes hypothesis-based experimentation. Second, we are planning to extend our question set to a questionnaire that can be send to a larger group of people for a more quantitative validation.

Also, we are planning to test the usage of our model in industrial settings. For this, we are planning to conduct case studies at several companies, where we would guide the company into using the conceptual model, and reflect on the efficiency. This could sharpen our framework and it would give further validation of the viability of our proposed work.

Last, we would like to extend our guidelines to even more actionable guidelines that could be used in the various stages a product can be in.

IX. CONCLUSIONS

In this research, we have shown that new product development is based on two types of decisions: architectural decisions and pivots. We have presented a conceptual framework that addresses both decisions in the context of an experimental risk-based process. This framework can help practitioners to structure their new product development process. From our interviews we derived a set of guidelines that emphasized the importance of decisions in experiments. Both architectural decisions as well as pivots play a vital role in the development of new products, as two sides of a medal representing the technical and the business part of a decision.

ACKNOWLEDGEMENTS

We would like to thank the following interviewees for taking the time to talk with us about their pivots: Mark Vletter, Gordon Rios, Christian Branbergen and Theo Hultberg.

REFERENCES

[1] J. Bosch, “Building Products as Innovation Experiment Systems”, in ICSOB, Springer, 2012, pp. 27-39.

[2] F. van der Linden, J. Bosch, E. Kamsties, K. Känsälä, and J. H. Obbink, “Software Product Family Evaluation”, in SPLC, Springer Verlag, 2004, pp. 110-129.

[3] A. G. J. Jansen and J. Bosch, “Software Architecture as a Set of Architectural Design Decisions”, in Proceedings of the 5th IEEE/IFIP Working Conference on Software Architecture (WICSA 2005), IEEE Computer Society, 2005, pp. 109-119.

[4] J. S. van der Ven, A. G. J. Jansen, P. Avgeriou, and D. K. Hammer, “Using Architectural Decisions”, in Second International Conference on the Quality of Software Architecture (Qosa 2006), Karlsruhe University Press, 2006, pp. 1-10.

[5] P. Kruchten, “An Ontology of Architectural Design Decisions in Software Intensive Systems”, in 2nd Groningen Workshop Software Variability, 2004, pp. 54-61.

[6] R. C. de Boer, R. Farenhorst, P. Lago, H. van Vliet, V. Clerc, and A. Jansen, “Architectural knowledge: getting to the core”, in Proceedings of the Quality of software architectures 3rd international conference, 2007, Springer-Verlag, 2007, pp. 197-214.

[7] J. S. van der Ven and J. Bosch, “Architecture Decisions: Who, How and When?”, in ASA, Agile Software Architectures (to be published), Elsevier, 2013, pp. unknown.

[8] D. Tofan, M. Galster, and P. Avgeriou, “Reducing Architectural Knowledge Vaporization by Applying the Repertory Grid Technique”, in Proceedings of the 5th European Conference on Software Architecture (ECSA), Springer LNCS, 2011, pp. 244-251.

[9] S. Blank, “The Four Steps to the Epiphany: Successful Strategies for Products that Win”, Lulu.com, 2008.

[10] E. Ries, “The Lean Startup: How Constant Innovation Creates Radically Successful Businesses”, Penguin Books Limited, 2011.

[11] A. Maurya, “Running Lean: Iterate from Plan A to a Plan That Works”, O'Reilly Media, Incorporated, 2012.

[12] J. Bosch, “Design and use of software architectures: adopting and evolving a product-line approach”, ACM Press/Addison-Wesley Publishing Co., 2000.

[13] C. Hofmeister, R. Nord, and D. Soni, “Applied Software Architecture”, Addison-Wesley, 2000.

[14] P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, J. Ivers, and R. Little, “Documenting Software Architectures: Views and Beyond”, Pearson Education, 2002.

[15] P. Kruchten, P. Lago, and H. V. Vliet, “Building up and Reasoning about Architectural Knowledge”, in in Proceedings of the Second International Conference on the Quality if Software Architectures (QoSA), Springer-Verlag, 2006, pp. 43-58.

[16] J. Tyree and A. Akerman, “Architecture Decisions: Demystifying Architecture”, in IEEE Softw., vol. 22 (2), 2005, pp. 19-27.

[17] J. S. van der Ven, A. Jansen, J. Nijhuis, and J. Bosch, “Design Decisions: The Bridge between Rationale and Architecture”, Rationale Management in Software Engineering, Springer, 2006, pp. 329-348.

[18] J. van Gurp and J. Bosch, “Design erosion: problems and causes”, in J. Syst. Softw., vol. 61 (2), 2002, pp. 105-119. [19] P. Kruchten, R. L. Nord, and I. Ozkaya, “Technical Debt: From Metaphor to Theory and Practice”, in IEEE Software, vol. 29 (6), 2012, pp. 18-21.

[20] K. M. Eisenhardt, “Building Theories from Case Study Research”, in Academy of Management Review, vol. 14 (), 1989, pp. 532-550.

[21] J. S. van der Ven, “Making the Right Decision: Supporting Architects with Design Decision Data”, in Software Architecture - 7th European Conference, ECSA, Springer, 2013, pp. 176-183.

[22] S. Thomke, “Enlightened Experimentation: The New Imperative for Innovation (HBR OnPoint Enhanced Edition)”, Harvard Business Review, 2001.

[23] K. M. Eisenhardt and B. N. Tabrizi, “Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry”, in Administrative Science Quarterly, vol. 40 (1), 1995, pp. 84-110.

[24] T. H. Davenport, “How to Design Smart Business Experiments”, in Harvard Business Review, vol. 87 (2), 2009, pp. 68-76.

[25] S. H. Thomke, “Experimentation Matters: Unlocking the Potential of New Technologies for Innovation”, Harvard Business School Press, 2003.