Hypothesis-driven Adaptation of Business Models based on Product Line Engineering

1st Sebastian Gottschalk
Software Innovation Lab
Paderborn University
Paderborn, Germany
sebastian.gottschalk@uni-paderborn.de

2nd Florian Rittmeier
Software Innovation Lab
Paderborn University
Paderborn, Germany
florian.rittmeier@uni-paderborn.de

3rd Gregor Engels
Software Innovation Lab
Paderborn University
Paderborn, Germany
gregor.engels@uni-paderborn.de

***Abstract*—The continuous innovation of its business models is an important task for a company to stay competitive. During this process, the company has to validate various hypotheses about its business models by adapting to uncertain and changing customer needs effectively and efficiently. This adaptation, in turn, can be supported by the concept of Software Product Lines (SPLs). SPLs reduce the time to market by deriving products for customers with changing requirements using a common set of features, structured as a feature model. Analogously, we support the process of business model adaptation by applying the engineering process of SPLs to the structure of the Business Model Canvas (BMC). We call this concept a Business Model Decision Line (BMDL). The BMDL matches business domain knowledge in the form of a feature model with customer needs to derive hypotheses about the business model together with experiments for validation. Our approach is effective by providing a comprehensive overview of possible business model adaptations and efficient by reusing experiments for different hypotheses. We implement our approach in a tool and illustrate the usefulness with an example of developing business models for a mobile application.**

***Index Terms*—business model decision line, business model adaptation, hypothesis-driven adaptation, software product line, feature model**

I. Introduction

The continuous innovation of its business models, which are defined by Osterwalder et al. as "the rationale of how the organization creates, delivers, and captures value" [1], is an important task for a company to stay competitive. This is one of the results of the GE Innovation Barometer 2018 [2], a study with over 2000 business executives. In this study, 64% of these executives have the "difficulty to define an effective business model to support new ideas and make them profitable" [2]. By comparing the results with a previous study of 2015 the challenge is getting even larger (59% of over 3000 executives). An important reason for this is that customers want solutions for perceived needs rather than just products [3]. This corresponds to the potential effect that the business model can be often more important than the latest technology of the product [4].

To bridge the gap between the business model and the customer needs, the approach of hypothesis-driven development can be used. Hypotheses-driven development continuously constructs hypotheses about the business. These hypotheses are validated or disapproved by conducting experiments with potential customers and the results are used to adapt the business model [5], [6]. To support this process, it would be effective to derive the different business models from a common superset to provide a comprehensive overview of possible business model adaptations. Moreover, it would be efficient to separate the construction of business models from conducting experiments, which allows reusing experiments for different hypotheses. Both, in turn, is possible with the concept of Software Product Lines (SPLs), which improve the product quality and reduce the time to market by deriving products for different customers from a common set of features, modeled as a feature model [7]. These features are separated from the software assets, which allows the reusing of the assets for different products.

In this paper, we introduce a Business Model Decision Line (BMDL) to derive different business models from existing business domain knowledge (in particular the market, competitors, own niche, and potential customers). The BMDL (see Fig. 1 for example) consists of a structure (i.e. *Business Feature Model*) and a process (i.e. combination of *Business Engineering* and *Customer Engineering*). As a structure to represent the business domain knowledge we are using a feature model. A feature model structures different features in a hierarchical order together with relationships between the features. We refine this model by using the nine building blocks (e.g. Customer Segments, Value Propositions) of the widely-used Business Model Canvas (BMC) [1] as the first hierarchy level. Each feature in a lower hierarchy level corresponds to a single business model decision (e.g. License Subscription as a business decision for Revenue Streams). These business decisions can be refined within the depth of the model (e.g. License Subscription could be refined to Standard Subscription and Premium Subscription). Each business decision is therefore also a hypothesis that needs to be validated. The process is adapted from SPL Engineering (SPLE) [8] and analyses business domain knowledge within a *Business Engineering* process to create a feature model. Within *Customer Engineering* process, these features are matched with the customer needs to derive possible business models.

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (CRC 901, Project Number: 160364472SFB901)

Preprint, cite this paper as:
Gottschalk S., Rittmeier F., Engels G. (2020) Hypothesis-Driven Adaptation of Business Models Based on Product Line Engineering. In: 2020 IEEE 22nd Conference on Business Informatics (CBI). IEEE

The final authentificated version is available online at:
https://doi.org/10.1109/CBI49978.2020.00022

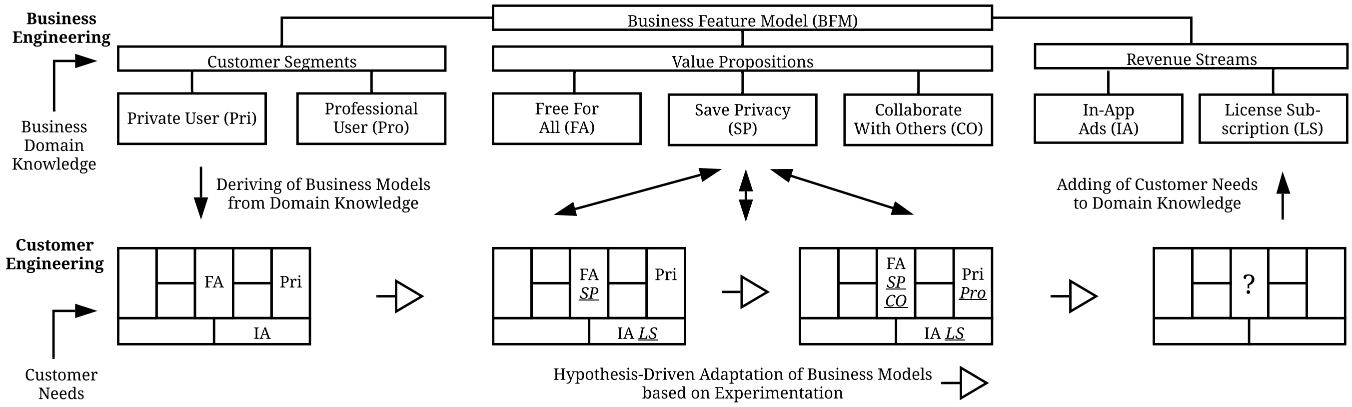


Fig. 1. Innovation of Business Models using Business and Customer Engineering

Moreover, the customer needs can lead to new features which need to be added to the business domain knowledge. The hypotheses in the derived business model can be tested by conducting experiments, whose results will be used to adapt the business model over time.

To show our approach, we are using the running example of a company that changes the business model of its mobile to-do list app. The main functionality of the app is to provide customers an organization of their daily tasks. Other needed features are extracted in the value proposition development of the business model. An example of these changes is shown in Fig. 1, where the company is focussing on a customer segment of *Private Users (Pri)* by providing an app, which is *Free For All (FA)* by using *In-App Ads (IA)*. As an experiment, they analyze the app reviews and find out that their customers want to *Save Privacy (SP)*. As a result, they add *License Subscription (LS)* as an additional revenue stream. From this, they extract the *Professional User (Pro)* as a new customer segment, where a customer survey shows the needs for a feature to *Collaborate With Others (CO)*.

In the following, Section 2 describes the foundations of Business Models and Software Product Lines. Section 3 shows our research approach based on Design Science Research, whose solution concept is presented in Section 4 and demonstrated on an expanded example of business models for the to-do app. Section 5 critically discusses our approach and the results of the expanded example. Finally, a conclusion is given in Section 6.

II. Background and Related Work

A. Business Models

The explicit modeling of business models is getting high attention in recent years. A business model "describes the design or architecture of the value creation, delivery, and capture mechanisms it employs. The essence of a business model is in defining the manner by which the enterprise delivers value to customers, entices customers to pay for value, and converts those payments to profit" [3]. To create a business model, a structure and a process are needed.

The structure of a business model can be represented with a business model modeling language. Within these modeling languages, connection-based, geometric-based, and hybrid visualizations are used [9]. Connection-based approaches describe the business model as a network of different objects and their relations. For example, the e3 Value Model [10] defines actors which are connected through value interfaces to each other. Geometric-based approaches describe the business model in the form of a visual template.

The widely-used Business Model Canvas (BMC) [1] is also a geometric-based approach. The BMC divides the business model into the nine building blocks of Customer Segments, Value Propositions, Channels, Customer Relationships, Key Activities, Key Resources, Key Partners, Revenue Streams, and Cost Structure. An example of the BMC for the business model of a to-do app can be seen in Fig. 2. The example consists of different customer segments (e.g. Private User, Professional User) from which money can be generated through different revenue streams (e.g. In-App Advertisements, License Purchase, License Subscription). While, in practice, the structuring of different ideas in a single canvas is done with different colored sticky notes [11], the underlying work [12] also introduced a Business Model Ontology (BMO) for a more

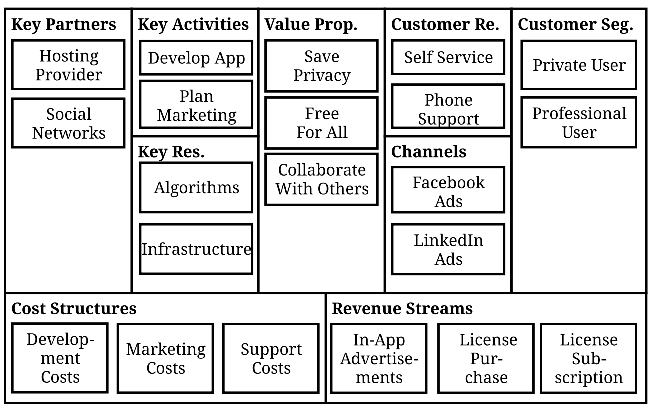


Fig. 2. Structure of the Business Model Canvas

specific structuring. This ontology can be directly used in an editor [11] but is also the basis for the concepts of dynamic business models [13] and meta-modeling [14]. To support this modeling, Osterwalder et al. [15] provide the idea to model different types of business models as taxonomies so that concrete businesses can be interpreted as instances of these taxonomies. These hierarchically ordered taxonomies can be also represented through feature models.

These structures are the source for different business model development tools like Strategyzer[1] or Canvanizer[2]. Szopinski et al. [16] analyze 22 business model tools to derive a taxonomy of their functions together with an agenda for future research. Two of these research points are the correctness checking and road mapping of business models. Both can be supported with feature models as a common superset to derive new business models. Augenstein et al. [17] develop design principles for a business model decision support system. They mentioned the importance of an underlying meta-model but not focus on its actual representation. The importance of meta-modeling and openness to uncertainty is also pointed out by Blaschke et al. [18]. In their work, they state that important requirements for business models in a model management system are the simplicity of the underlying meta-model, the openness to describe business models which have never been represented before, and the decomposability to particular information views. Dellermann et al. [19] create design principles for a hybrid decision support system to combine a formal analysis of the business model with the knowledge of human experts. They mentioned the importance of an iterative validation process with humans but not focus on the different experiments which are needed to gather the feedback.

The process of business model innovation is a challenging task, which often uses creativity and collaboration between different stakeholders [20]. To solve this challenge, Teece [3] argues that for business model innovation a deep analysis of the market, the existing competitors, and potential customers is necessary. From this analysis, the market with the customers can be divided into different segments for which matching value propositions and isolation mechanisms have to be defined. This corresponds to the opinion of Chesbrough [6] who stated that a good underlying business model can be more important than the latest technology of the product itself. Because different business models will also lead to different outcomes within the market, there is a need for experimentation to test these models within the market and justify possible business model adaptations [6].

This experimentation is referred to under different terms like rapid experimentation [21], discovery-driven planning [5], or experiment-driven development [22] in literature. All of them are based on similar basic principles. In this paper, we are using the term hypothesis-driven development to focus on hypotheses as the main artifact of our approach. Here, the business model can be interpreted as a collection of hypotheses that need to be validated or disapproved over time by conducting experiments. The results of these experiments are used to adapt the business model which leads to new hypotheses. To support this experimentation, different conceptual models, workshop settings and artifacts have been proposed in literature. Conceptual models support experimentation by structuring the engineering process. HYPEX [23] develops a feature backlog from strategic product goals. From this backlog, a feature with a high priority is selected and implemented in the product. By analyzing the gap between the expected and actual behavior of the feature, a decision to develop hypotheses about alternative implementations or continuing with the next feature of the backlog is taken. RIGHT [22] is similar to HYPEX by identifying and prioritizing hypotheses from the business model and strategy. These hypotheses are tested with experiments where matching features are implemented in the product and their usage is measured. After that, a decision is made if the implementation of the feature or the business model needs to change. Both HYPEX and RIGHT can be represented through the Build-Measure-Learn-Cycle of LEAN-Development [24]. While both approaches focus on the quantitive measurement, QCD [25] is a conceptual model to combine qualitative customer feedback with quantitative customer observations. Based on the business strategy, they build a hypothesis backlog from which a hypothesis is chosen and an experiment is selected. The result of the experiment is used to update the hypothesis backlog. In comparison to the discussed conceptual models, we focus our concept on building a comprehensive overview of possible business model adaptations and allow the reusing of experiments.

Another support of innovating business models can be done with workshops and artifacts. A common workshop technique is Design-Thinking [26], where the participants try to understand the underlying problem, explore different solutions, and test them with customers. A typical artifact for business model innovation is the business model pattern catalog [27]. For this pattern catalog, the authors have analyzed various companies to find 55 business patterns. They state that most new business models can be configured by combining these patterns. Moreover, there exists a catalog of 44 experiments to test hypotheses about the business model [21]. To use the artifacts in the workshops, the BMILab is providing pattern cards both for Business Model Innovation[3] and for Business Model Testing[4]. While our approach can be used to structure the workshop results, it can also integrate the artifacts during the construction of business models and conduction of experiments.

B. Software Product Lines

Software Product Lines (SPLs) can be defined as “a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market

[1]Strategyzer: https://www.strategyzer.com/
[2]Canvanizer: https://canvanizer.com/
[3]BMI Innovation Cards: https://bmilab.com/pattern-cards
[4]BMI Testing Cards: https://bmilab.com/testing/explore

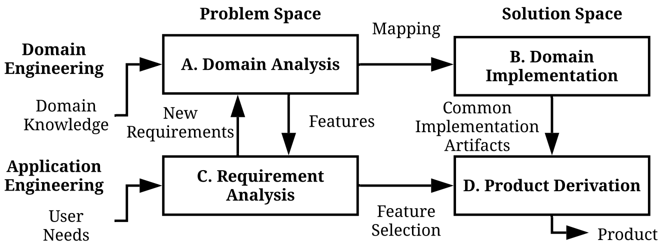


Fig. 3. Engineering Process for SPLs (based on Apel et al. [8])

segment or mission and that are developed from a common set of core assets in a prescribed way" [7]. To use these SPLs, a structure and an engineering process are needed.

The structure of SPLs can be represented using hierarchical feature modeling which is shown in Fig. 5. Features can be mandatory or optional for the model instances. Moreover, there can be Or (at least one sub-feature is selected), and Xor (exactly one sub-feature is selected) relationships between a parent and a child feature. To refine the model instance, cross-tree constraints for requiring and excluding dependencies can be made. A big issue in SPL development is to find the right granularity for the features [28]. Another representation to model variabilities for SPLs are Orthogonal Variability Models (OVMs) [29], which consider just the variability points of the product. In [30], we are using OVMs to introduce a Business Variability Model for business model decisions. In [31], we used the idea of feature models for the intertwined development of a business model and product functions for mobile applications. Moreover, we first introduced the term of a Business Model Decision Line (BMDL). In contrast to that work, this paper provides a holistic approach to create BMDLs for different domains by matching business domain knowledge with customer needs.

The engineering process for SPLs is shown in Fig. 3. The process can be divided into the *Domain Engineering*, which consists of the analysis of the domain and the development of reusable artifacts, and the *Application Engineering*, which uses these artifacts for the development of a specific software product for a user group. Moreover, the *Problem Space* describes the user perspective on the requirements of the software product, while the *Solution Space* covers the developer's perspective on the design and implementation of the software product. Based on this classification the process consists of the following four steps [8]:

- **A. Domain Analysis** identifies the domain scope of the different products which can be developed with the SPL. From this analysis the reusable artifacts are identified and modeled as a feature model.
- **B. Domain Implementation** develops the different reusable artifacts (e.g. source code, test scripts) for further usage in the product derivation.
- **C. Requirement Analysis** extracts the requirements of a single user for the product. This requirements can lead to a feature selection of the SPL or the adding of new requirements to the domain analysis.
- **D. Product Derivation** is the matching of user requirements and reusable artifacts to build a product.

The division of domain and application engineering can be used to separate the overview of possible different business models from selected ones. Moreover, the division of problem and solution space can be adapted to separate the construction of business models and conduction of experiments. To support the engineering process for SPLs, Krueger [32] points out three modeling approaches: The *(1) Proactive Approach* uses an extensive domain analysis in the beginning to create a complex model for all product variabilities. The *(2) Extractive Approach* collects features from a set of existing products. To simplify the process, the developer can start with a small set of features and incrementally refactor the SPL. The approach is used when developers want to develop a common SPL for existing single products. The *(3) Reactive Approach* starts with a minimal software product line which is extended incrementally when new requirements appear. It is used if new users lead to unpredictable requirements.

Because of the high dynamic of business models, we adopt the idea of Alves et al. [33] to build an extractive feature model of business model decisions that is extended in a reactively by analyzing customer needs.

III. Research Approach

This paper uses the Design Science Research process [34], which consists of the six stages of Identify Problem & Motivate, Define Objectives of Solution, Design & Development, Demonstration, Evaluation, and Communication.

The paper shows the first cycle of this process and starts with a problem-centered initiation as an entry point of the process. In the *Identify Problem & Motivate* step, we have identified the problem of finding an effective business model with a study of the GE Innovation Barometer [2]. An important reason for that is that nowadays customers want solutions for their needs instead of just products [3] for which experiments with customer needs to been conducted [6]. Here our motivation is to support this experimentation with an effective and efficient approach. This support in effectivity and efficiency are also our *Define[d] Objectives of Solution*. For this solution, we combine the engineering process of Software Product Lines with the structure of the Business Model Canvas. The choices for this solution are reasoned in the next section. In the *Design & Development* step, we will use both concepts to design our solution concept which consists of a structure and a corresponding process. Moreover, we develop the BMDL Feature Modeler as an implementation. As *Demonstration*, we show the expanded example of business models for a to-do app. We provide a preliminary *Evaluation* in the form of a discussion about the effectiveness and efficiency of our approach together with limitations we found by developing business models for the to-do app. Finally, our *Communication* step will be done with the publication of this paper and the developed BMDL Feature Modeler.

IV. Business Model Decision Line

As a solution to separate the construction of business models from conducting experiments, which allows the explicit modeling of business domain knowledge and reuse of experiments for different hypothesis, we propose the Business Model Decision Line (BMDL). Inside the BMDL we can define a set of business model decisions, which can be matched with customer needs, to derive specific business models.

After defining some initial considerations about concepts behind our solution, we present the engineering process and the structure of the BMDL. Moreover, we present web-based tooling for the BMDL.

A. Initial Considerations

As the foundation for our solution, we adapt the concept of SPLs because it is a generic concept that provides high flexibility for customization. Our choice is based on two objectives: First, we want to derive the different business models from a comprehensive overview of all possible business model adoptions. This should be effective because a comprehensive overview allows the discovery of multiple possible configurations [3]. Second, we want to separate the construction of business models from conducting experiments. This should be efficient because reusing assets is used in software products to reduce costs [8] and we assume a similar cost reduction in artifacts for experiments. Moreover, we build the structure and process of the BMDL as follows.

The structure of the BMDL is based on feature models. We use them because they provide an easily understandable structure, where basic dependencies can be defined. A similar concept is used in [15], where taxonomies are defined as a model for different businesses of the same type. We refine the model with the Business Model Canvas [1] because it is de-facto [9] in the industry, which should lead to easy usage.

The process of the BMDL is divided into the *Problem Space* and the *Solution Space*. In the *Problem Space*, the analysis of the market, the competitors, and niche together with the segmentation of customers and the definition of a funnel are based on the concept of Teece in [3]. Out of the analysis, we build an extractive feature model as mentioned in [33]. In the *Solution Space*, we split the approach in *Business Engineering* and *Customer Engineering*. In *Business Engineering*, we separate the experimentation settings from the experimentation artifacts to allow reusing of a single artifact for different settings. In *Customer Engineering*, we separate the business model derivation from the experiment derivation to ensure a flexible decision about the prioritization of different experiments. The conduction of experiments is similar to other conceptual models of hypotheses-driven development [22], [23], [25].

B. Engineering Process of BMDLs

The engineering process for SPLs is shown in Fig. 4. The process can be divided into the *Business Engineering*, which analyses the different business model decisions and implements reusable artifacts, and the *Customer Engineering*, which analyses the needs of the customer and derives a specific business model for the product of the company. Moreover, the *Problem Space* describes the customer perspective on the requirements of the business model, while the *Solution Space* covers the business developer's perspective on the implementation of the business model.

Based on this classification, the process consists of the following four steps of *Business Analysis*, *Business Implementation*, *Customer Analysis*, and *Business Derivation*. For each of these steps, we provide an execution guideline together with the results, which we derived from our expanded example. In the example, we create an adaptive business model for a to-do app in the market of mobile applications. The intermediate results of each step can be accessed through the BMDL Feature Modeler.

1) Step A. Business Analysis: The *Business Analysis* can be divided into the parts of *Domain Scoping* and *Domain Modeling*. The *Domain Scoping* collects the information about the target market, existing competitors, and the niche of the own product. The *Domain Modeling* uses this information to create a feature model.

Step A.1. Domain Scoping collects the information, which is needed to create a domain model. In the beginning, the business developer needs to define the range of the scope. If the main business decision points of the business model are already clear and the focus is to refine and validate these decisions, a small scoping is chosen. In contrast to that, a broad scoping allows to discover a wider range of possible business models and focuses on the development of new business model ideas. After this decision, the collection of information is based on experience in the domain (e.g. expert knowledge, competitor's business models) and newly generated knowledge (e.g. customer surveys, market reports about future trends). While SPLE is mostly based on the analysis of existing software systems, our approach is focusing on the analysis of the market, the individual stakeholders, and the specific product positioning. To improve the results of the information retrieval, we structure it into *Market Analysis*, *Competitor Analysis*, and *Niche Analysis*.

- **Step A.1.1. Market Analysis** has the goal to gather an overview of the business domain and the existing stakeholders. Typical questions of the analysis are: What is the target market size and what are typical customer segments within the market? How are these customers reachable? Which revenue streams are accepted by the customers? To get into the market, interviews with domain experts or the analysis of market reports can be done.
  In our example, we used the existing business domain model in [31] as a foundation. It was created for mobile applications and consists of 89 variation points of different business model decisions.
- **Step A.1.2. Competitor Analysis** is based on the results of the *Market Analysis* and has the goal to analyze how their competitors solve a customer problem and which business models are accepted by the potential customers in the market. Typical questions of the analysis are: What

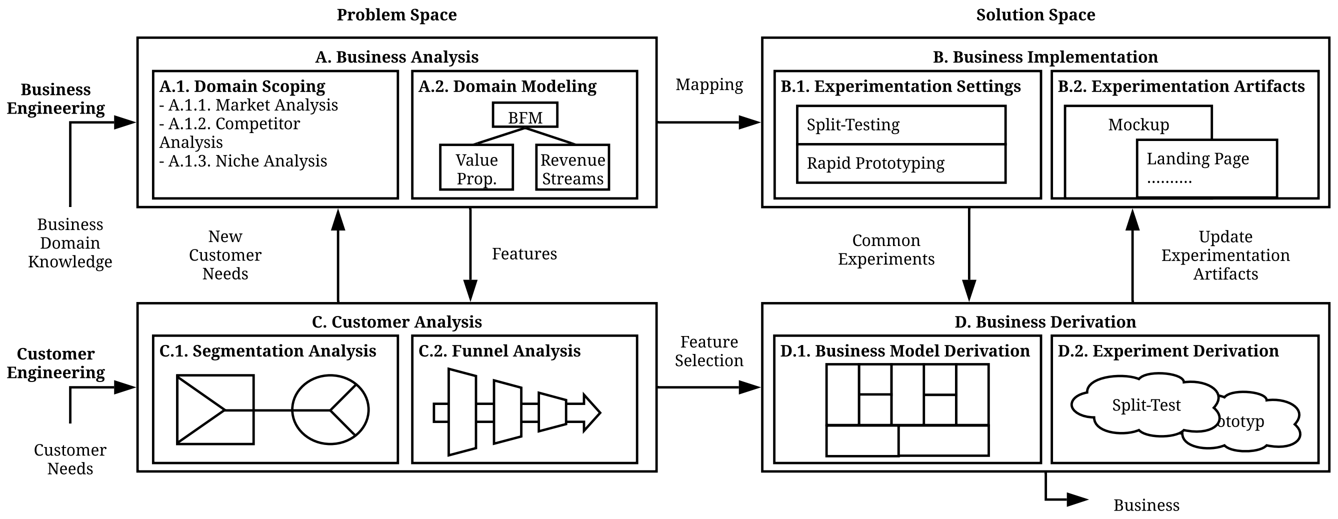


Fig. 4. Overview of the Engineering Process for BMDLs

problems are solved by the competitors? What parts of their business model did they change in the past and why? How do they reach their customers? To analyze the competitors, the business developer can read news articles or execute the product and map their decisions to the Business Model Canvas using the guiding questions of the Business Model Generation book [1].
In our example, we analyzed the to-do apps named Microsoft ToDo (which consists of 47 business model decisions), Wunderlist (60 decisions), Any.do (57 decisions) and Todoist (57 decisions), together with the on-board applications of Apple's iOS (27 decisions) and Google's Android (26 decisions). Here, we found out that most apps provide a generic version that does not focus on a specific customer segment and based their revenue streams on a subscription model.

- **Step A.1.3. Niche Analysis** is based on the results of the *Competitor Analysis* and has the goal to differentiate the own product from the competitors using a unique selling point. Typical questions of the analysis are: What parts of the own solution are providing a competitive advantage? Do the competitors ignore some specific customer segments? Are there some acquisition channels which they are not using? To analyze the niche, the developer can analyze the weaknesses of competitors and conduct interviews with potential customers.
In our example, we found different niches (31 decisions), which are currently not implemented by the analyzed competitors. Examples would be a focus on specialized customer segments (e.g. fitness improver). and the usage of time management features within todo applications (e.g. time tracking, using of Pomodoro technique).

Step A.2. Domain Modeling creates a feature model out of the information of the *Domain Scoping*. To derive the feature model, the business developer has to identify the commonalities and differences in the information, which can be done by building a taxonomy. The idea of using taxonomy development for different types of business models is based on [15]. For this, exiting methods like Nickersons Taxonomy Development [35] or Framework Analysis [36] can be used. In [31], the taxonomy development method of Nickerson was adopted to create a domain model for apps. To cross-check the constraint quality of the feature model, the business developer can create possible and impossible business models with the Business Model Canvas to check them against the feature model. In our example, we used the business model decisions to abstract a feature model with 152 features.

2) Step B. Business Implementation: The *Business Implementation* implements experiments to test the hypotheses of the *Business Analysis*. For this, each experiment consists of a setting for conducting the experiment together with possible artifacts, which are used during the evaluation. To reuse the artifacts for different settings, we split the step up into *Experimentation Settings* and *Experimentation Artifacts*.

- **Step B.1. Experimentation Settings** provide an overview of the experiments which can be used to validate the different hypotheses. Each experiment setting consists of a name and a summary together with a description of how to prepare, execute, and analyze the experiment. Moreover, each of them can be linked to the artifacts which can be used during the experimentation. In [21], Bland et. al introduce a catalog of 44 different experiments, which can be used for the testing of hypotheses.
In our example, we used five experiments, which are mentioned by Bland et al. [21]. These are a customer survey, a landing page, a split test, a clickable prototype, and a single feature minimum viable product. These experiments could be used to test most business model decisions of the *Business Analysis*.

- **Step B.2. Experimentation Artifacts** are assets that can be used to support the experiments. Instead of the (semi-)automated execution in the case of selecting different variabilities in the code of software products, our solution provides experimental artifacts to reduce the experimentation time to validate the different hypotheses of the business model. These artifacts have to be detailed enough to perform concrete experiments and abstract enough to handle a wide range of configurations of the feature model. While in SPLs most code is built for reusability, BMDL uses the experimentation artifacts just for a part of the hypotheses. This is because some experiments are too specialized for efficient reuse of artifacts. Moreover, in contrast to SPLs, the artifacts are updated during the experimentation.
  In our example, we used three artifacts. These were a Facebook page, which could be used for the experiments of testing a landing page and providing a split test, a web-based landing page, which was used for testing a landing page and a mockup, which was used for clickable prototypes and single feature minimum viable products.

3) Step C. Customer Analysis: The *Customer Analysis* connects the knowledge of *Business Analysis* with the concrete needs of the customers by using various processes (e.g. customer journeys [37]) and artifacts (e.g. Value Proposition Canvas [38]). To improve the results of the information retrieval, we structure it into *Segmentation Analysis* and *Funnel Analysis*.

- **Step C.1. Segmentation Analysis** defines a small customer segment and corresponding value propositions for possible early adopters. Typical questions of the analysis are: To which different customer segments can I serve my products? How these customers currently solving their issues? How does this customer segment differentiate from the other customers in the same market? To find these target groups, artifacts like value proposition design or processes like personas analysis can be used.
  In our example, we focused on different customer segments of private users for which we proposed different value propositions to solve their problems. Here, we have the fitness improver, who will get special features for meal and gym times, and life improver, who we will propose to save a work&life balance. Moreover, we have business improver, where we will use the proposition to optimize workflows to get more done at the same time.
- **Step C.2. Funnel Analysis** is based on the *Segmentation Analysis* and has the goal to acquire new customers and retain existing customers. This, in turn, provides the scalability of the business model. Typical questions of the analysis are: Where to reach the customers of the target group analysis? Can different key partners be used as mediators to sell the product? How will the product integrate into the life of the customer? To create a funnel, customer surveys or customer journeys can be performed.
  In our example, we found out that a broader range of customer segments can be reached with Facebook advertisements, while dedicated customer segments can be found in specialized online communities.

Ideally, the satisfaction of customer needs can be met with a simple feature selection. If this is not possible, the business developer can decide to mark the needs as an out-of-scope need or add the new customer need to the *Business Analysis*.

4) Step D. Business Derivation: The *Business Derivation* matches the results of the *Customer Analysis* with the feature model of the *Business Analysis*. Out of this matching, a business model together with experiments for validation is derived from the *Business Implementation*. To ensure a flexible decision of experimentation prioritization, we split the step into the *Business Model Derivation* and *Experiment Derivation*.

- **Step D.1. Business Model Derivation** matches the results of the *Customer Analysis* with the feature model of the *Business Analysis* to derive a valid business model. The business model can be seen as a set of features, which need to be validated over time by conducting experiments.
  In our example, we derived different business models for the customer segments of fitness improver, life improver, and business improver. Example business model decisions which need to be tested are a Facebook group for customer acquisition and calories tracker as value offering for fitness improver, social media ads for customer acquisition and optimize schedule as value offering for life improver and customer support as relationship and workflow tracking as value offering for business improver.
- **Step D.2. Experiment Derivation** receives the experiments from the *Business Implementation*. These consist of the settings of how the experiment can be conducted together with artifacts that can support the validation. The experiments can be used to test the hypotheses of the *Business Model Derivation* and adapt the business model. Moreover, during the experimentation, the updated artifacts are pushed back to the *Business Implementation*.
  In our example, we could use a split-testing for the customer segments with Facebook advertisements to test different conversion rates of a prepared landing page. Moreover, we could discuss possible features and value propositions of the app in specialized online communities using our prepared customer survey and designed mockups. Results could be that a customer segment needs to define more precisely or some segments prefer specific pricing options.

C. Structure of BMDLs

The structure of the BDML is based on a feature model. Each building block of the Business Model Canvas is translated to a feature of the model and each business model decision inside a block is modeled as a subfeature. These subfeatures can be further refined to build a hierarchy of business model decisions. A major issue of this refinement is to find this right granularity for the features which correspond

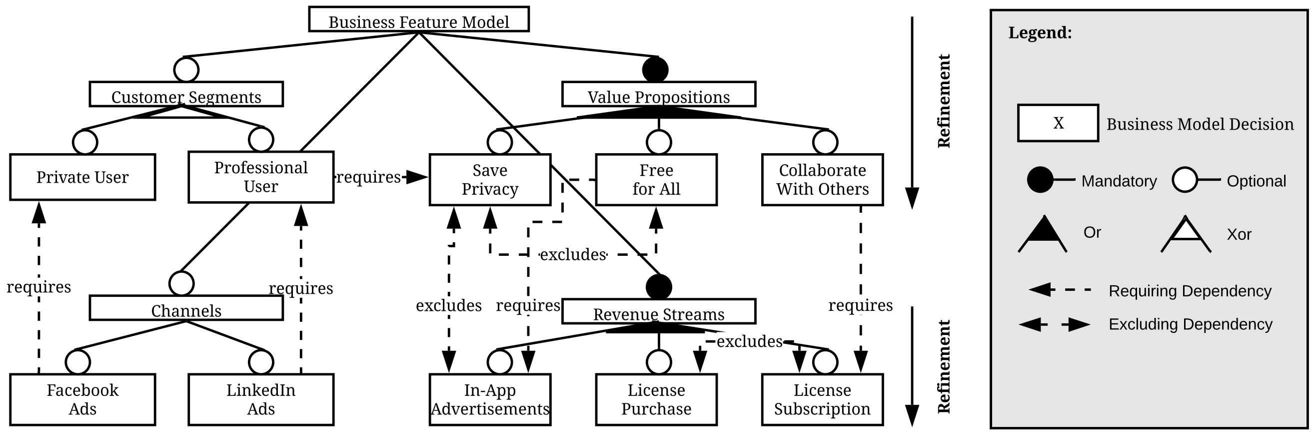


Fig. 5. Feature Model Excerpt of our Business Model

to the same issue in SPLs [28]. Here, we model features vertical to each other (e.g. Advertisement, Subscription) if they correspond to different decisions and refine them if different characteristics lead to constrains to other features (e.g. Monthly Subscription, Yearly Subscription).

We have modeled an excerpt of the business model of our to-do app in Fig. 2 as a feature model in Fig. 5. Here, the business developer can choose different features for the building blocks of Customer Segments, Value Propositions, Channels, and Revenue Streams which corresponds to different dependencies. For example, *Professional User[s]* require an option to *Save Privacy*, which is excluded from using *In-App Advertisements*. The complete feature model of our expanded example can be accessed through our BMDL Feature Modeler.

D. Tooling of BMDLs

We implemented our solution in a web-based tool called BMDL Feature Modeler[5]. The tool is based on the Angular Framework and the PouchDB database to run directly in the web browser[6]. Within the modeling tool, see Fig. 6, it is possible to create feature models, derive specific business models, and check the conformance between feature model and business models.

Fig. 6 provides different views of our tool based on our example of business models for a to-do app in Fig. 2. The first view (a) shows a list of the features with the possibility to add new features, modify the existing features, and create dependencies between these features. The second view (b) provides an overview how business models can be derived from the feature models. Here, the business developer can create canvas representations of different business models and adapt them over time. To fill the canvas he can choose existing features from the feature model or add new business model decisions. These decisions will be directly reflected in the feature model to save consistency. Based on the canvas representation the business developer can check the conformance of his model in the third view (c). In our example, the *In-App Advertisement* is excluded from *Save Privacy* and the *Facebook Ads* requires *Private User[s]* as customers (see Fig. 5 for constraints). All conformance errors are explained in the form of a text so that they can easily be corrected by the business developer.

[5]Source Code: https://github.com/sebastiangtts/bmdl-feature-modeler/

[6]Live Demonstration: https://sebastiangtts.github.io/bmdl-feature-modeler/

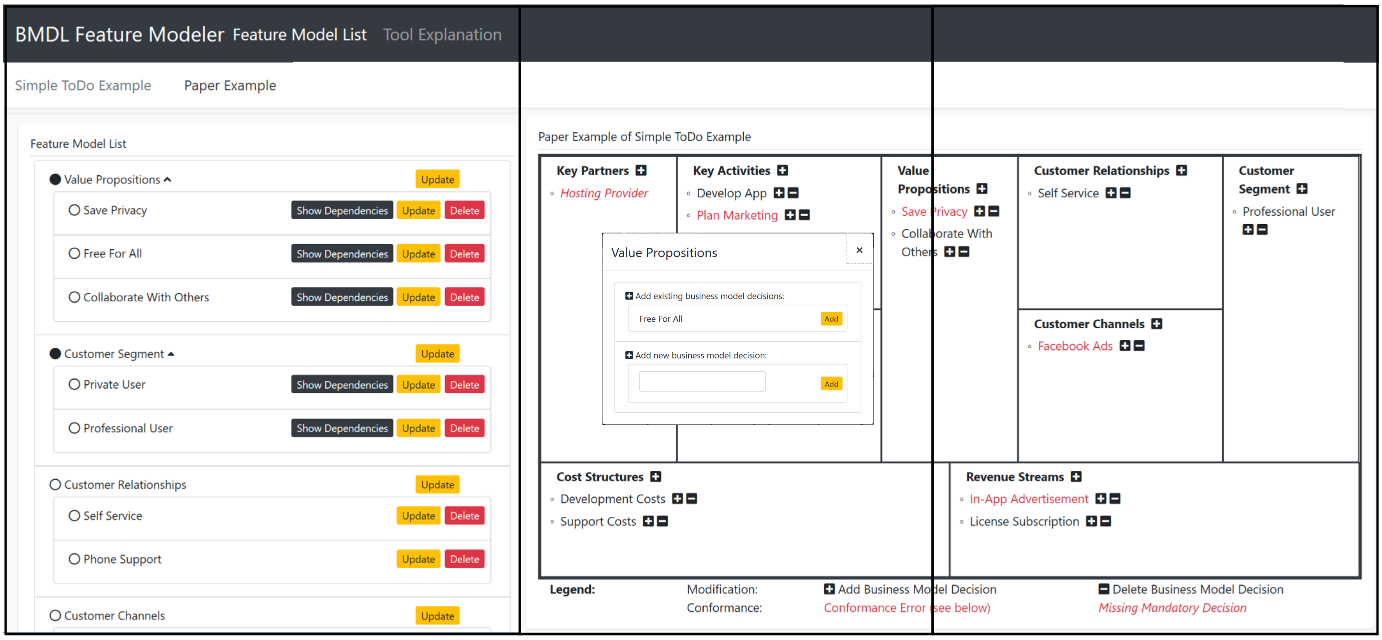
V. Discussion

During the execution of our example of developing business models for a to-do app, we preliminary evaluated our approach in terms of effectiveness and efficiency and found potential limitations. Both, we want to discuss in this section.

First, we analyze the effectiveness and efficiency of our approach.

The effectiveness of our approach is supported by the derivation of business models from a common superset of possible business model adaptations. The common superset is created on a deep analysis of the market, the competitors, and the customers, which was proposed in [3]. While deep analysis has an upfront-invest for the deep analysis, we discovered in our example, that it has advantages in combining working business model decisions of competitors with our own niche decisions to gain a competitive advantage. Moreover, we observed the advantage to use the feature model which is developed for one product as a starting point for the development of the business model of another product. The problem of this deep analysis is discussed in the limitation of the *Scoping Process*.

The efficiency of our approach is supported by the separation of the construction of business models from conducting experiments. The concept is based on the idea of asset reusing in software engineering [8]. While the separation has an upfront-invest in the identification of reusable artifacts, we discovered in our example that is has an advantage if mostly software artifacts are used during the experimentation. Moreover, we observe that by splitting up the construction and conduction it is easier to find experiments that can test multiple hypotheses at the same time. The problem of this reusing is discussed in the limitation of the *Artifacts Development*.



a) Create Feature Model **b) Choose Business Model Decisions** **c) Check Conformance**

Fig. 6. Screenshots of the BMDL Feature Modeler

Second, we identify the potential limitations of our approach. We divide them into the topics of *Scoping Process*, *Modeling Granularity*, and *Artifact Development*.

The *Scoping Process* is a multi-step process (analysis of the market, competitors, and niche) at the beginning of the *Business Analysis* to derive the main business domain knowledge. In comparison to existing idea generation processes like conducting design thinking workshops [26], it is comparatively time-consuming to analyze the market and competitors instead of starting directly with the development of a business model. Nevertheless, we are convinced that this holistic overview of the market and competitors has an advantage in terms of building a business model on working concepts and finding a niche to start with the development. Moreover, once the extractive scoping process is finished, the domain model can be improved in a reactive way [32] (e.g. finding a new competitor) and reused for different customer segments and products of the company.

The *Modeling Granularity* is a major issue both in business modeling and traditional SPL engineering [28]. This issue is supported by the fact, that a finer granularity of the modeling can lead to a large time increase in the scoping process (e.g. found features at single competitors need be to cross-checked with other competitors to derive variabilities). To solve this issue, we propose starting with a coarser granularity to model the main variation points of different business model decisions and refine this granularity when it is required (e.g. targeted attributes of a customer segment for running advertisements).

The *Artifact Development* takes place during *Business Implementation* and should provide reusable artifacts to speed up the further experimentation process. While, at the beginning of our research, we assumed that most of the artifacts for experimentation are reuseable, we realized that some artifacts are too specific to reuse them one-to-one for different hypotheses. This works especially for experiments whose artifacts are not based on software like customer surveys. But in the end, this result matches with our assumption that reusing works good in software engineering [8]. Nevertheless, we see it as an advantage for business developers to think about which artifacts are reuseable and which former artifacts can be evolved for reusing to speed up the business model adaptation.

VI. Conclusion

The continuous innovation of its business models is an important task for a company to stay competitive. During this process, the company has to validate various hypotheses about its business models by adapting to uncertain and changing customer needs efficiently and effectively. These hypotheses are validated or disapproved by conducting experiments with potential customers. To support this adaptation process, we introduced the Business Model Decision Line (BMDL), which is based on the concept of a Software Product Line (SPL). It consists of an engineering process with different steps (*Business Analysis*, *Business Implementation*, *Customer Analysis*, *Business Derivation*), and a structure based on the Business Model Canvas (BMC). The BMDL is effective by deriving different business models from a common superset of possible business model adaptations and efficient by separating the construction of business models from conducting experiments to reuse artifacts for experimentation. With the BMDL the company can analyze its business domain knowledge to generate a feature model. The feature model can be used to derive new business models by matching the features with the customer needs. These business models consist of different hypotheses that need to be validated or disapproved by conducting experiments. The results of these experiments are used to adapt the business model over time. We illustrate the usefulness with an example of developing business models for a to-do app. Moreover, we preliminary evaluate our approach by discussing the effectiveness and efficiency of our approach together with potential limitations in terms of the *Scoping Process*, the *Modeling Granularity*, and the *Artifact Development*.

Our future work is twofold and mainly deals with optimizing the engineering process and evaluating our approach: First, we want to use questionnaire models to improve the derivation of business knowledge and customer needs. Moreover, we want to integrate business model patterns in our tool to support idea generation. Second, we want to evaluate the applicability of our approach by providing a user study in the form of a student seminar on the lean development of mobile applications at the university. In the seminar, our approach should be used over time to analyze the evolvement of the feature model and the reusability of the experiments.

REFERENCES

[1] A. Osterwalder and Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Hoboken: John Wiley & Sons, 2010.

[2] General Electric Inc, "GE Global Innovation Barometer 2018." [Online]. Available: https://www.ge.com/reports/innovation-barometer-2018/

[3] D. J. Teece, "Business Models, Business Strategy and Innovation," *Long Range Planning*, vol. 43, no. 2-3, pp. 172–194, 2010.

[4] H. Chesbrough, "Business model innovation: it's not just about technology anymore," *Strategy & Leadership*, vol. 35, no. 6, pp. 12–17, 2007.

[5] R. G. McGrath, "Business Models: A Discovery Driven Approach," *Long Range Planning*, no. 43, pp. 247–261, 2010.

[6] H. Chesbrough, "Business Model Innovation: Opportunities and Barriers," *Long Range Planning*, vol. 43, no. 2-3, pp. 354–363, 2010.

[7] P. Clements and L. Northrop, *Software product lines: Practices and patterns*, 7th ed. Boston: Addison-Wesley, 2009.

[8] S. Apel, D. Batory, C. Kästner, and G. Saake, *Feature-Oriented Software Product Lines*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[9] T. John, D. Kundisch, and D. Szopinski, "Visual Languages for Modeling Business Models: A Critical Review and Future Research Directions," in *Proceedings of the 38th International Conference on Information Systems (ICIS)*. AIS, 2017.

[10] J. Gordijn, "e-Business value modelling using the e3-value ontology," in *Value Creation from E-Business Models*. Elsevier, 2004, pp. 98–127.

[11] B. Fritscher and Y. Pigneur, "Supporting Business Model Modelling: A Compromise between Creativity and Constraints," in *Task Models and Diagrams for User Interface Design*. Springer, 2010, vol. 5963, pp. 28–43.

[12] A. Osterwalder, "The Business Model Ontology: A Proposition in a Design Science Approach," Dissertation, University of Lausanne, Lausanne, 2004.

[13] F. Cosenz and G. Noto, "A dynamic business modelling approach to design and experiment new business venture strategies," *Long Range Planning*, vol. 51, no. 1, pp. 127–140, 2018.

[14] L. O. Meertens, M. E. Iacob, L. Nieuwenhuis, M. J. van Sinderen, H. Jonkers, and D. Quartel, "Mapping the Business Model Canvas to ArchiMate," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, 2012.

[15] A. Osterwalder, Y. Pigneur, and C. L. Tucci, "Clarifying Business Models: Origins, Present, and Future of the Concept," *Communications of the Association for Information Systems*, vol. 16, 2005.

[16] D. Szopinski, T. Schoormann, T. John, R. Knackstedt, and D. Kundisch, "Software tools for business model innovation: current state and future challenges," *Electronic Markets*, vol. 60, no. 11, p. 2794, 2019.

[17] D. Augenstein, C. Fleig, and A. Maedche, "Development of a Data-Driven Business Model Transformation Tool," in *Designing for a Digital and Globalized World*. Springer, 2018, vol. 10844, pp. 205–217.

[18] M. Blaschke, M. L. Wuetherich, U. V. Riss, and P. Papakostas, "Business Model Management System Design: Manifesto, Requirements, and Prototype," in *Proceedings of the 3rd International Conference on Enterprise Systems (ES)*. IEEE, 2015, pp. 153–160.

[19] D. Dellermann, N. Lipusch, P. Ebel, and J. M. Leimeister, "Design principles for a hybrid intelligence decision support system for business model validation," *Electronic Markets*, vol. 29, no. 3, pp. 423–441, 2019.

[20] M. J. Eppler, F. Hoffmann, and S. Bresciani, "New business models through collaborative idea generation," *International Journal of Innovation Management*, vol. 15, no. 06, pp. 1323–1341, 2011.

[21] D. J. Bland and A. Osterwalder, *Testing business ideas*. Hoboken: John Wiley & Sons, 2020.

[22] F. Fagerholm, A. Sanchez Guinea, H. Mäenpää, and J. Münch, "The RIGHT model for Continuous Experimentation," *J. Syst. Softw.*, vol. 123, pp. 292–305, 2017.

[23] H. H. Olsson and J. Bosch, "The HYPEX Model: From Opinions to Data-Driven Software Development," in *Continuous Software Engineering*. Springer, 2014, vol. 14, pp. 155–164.

[24] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. USA: Crown Business, 2014.

[25] H. H. Olsson and J. Bosch, "Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation," in *Software Business*. Springer, 2015, vol. 210, pp. 154–166.

[26] M. Geissdoerfer, N. M. Bocken, and E. J. Hultink, "Design thinking to enhance the sustainable business modelling process – A workshop based on a value mapping process," *Journal of Cleaner Production*, vol. 135, pp. 1218–1232, 2016.

[27] O. Gassmann, K. Frankenberger, and M. Csik, *The business model navigator: 55 models that will revolutionise your business*. Harlow, England and London and New York and Boston and San Francisco: Pearson, 2014.

[28] C. Kästner, S. Apel, and M. Kuhlemann, "Granularity in software product lines," in *13th International Conference on Software Engineering (ICSE)*. ACM, 2008, p. 311.

[29] A. Metzger, K. Pohl, P. Heymans, P.-Y. Schobbens, and G. Saval, "Disambiguating the Documentation of Variability in Software Product Lines: A Separation of Concerns, Formalization and Automated Analysis," in *15th IEEE International Requirements Engineering Conference (RE)*. IEEE, 2007, pp. 243–253.

[30] S. Gottschalk, F. Rittmeier, and G. Engels, "Business Models of Store-Oriented Software Ecosystems: A Variability Modeling Approach," in *Business Modeling and Software Design*. Springer, 2019, vol. 356, pp. 153–169.

[31] ——, "Intertwined Development of Business Model and Product Functions for Mobile Applications: A Twin Peak Feature Modeling Approach," in *Software Business*. Springer, 2019, pp. 192–207.

[32] C. Krueger, "Easing the Transition to Software Mass Customization," in *Software Product-Family Engineering*. Springer, 2002, vol. 2290, pp. 282–293.

[33] V. Alves, P. Matos, L. Cole, P. Borba, and G. Ramalho, "Extracting and Evolving Mobile Games Product Lines," in *9th International Conference on Software Product Lines (SPLC)*. Springer, 2005, pp. 70–81.

[34] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J Manag Inf Syst*, vol. 24, no. 3, pp. 45–77, 2007.

[35] R. C. Nickerson, U. Varshney, and J. Muntermann, "A method for taxonomy development and its application in information systems," *European Journal of Information Systems*, vol. 22, no. 3, pp. 336–359, 2013.

[36] R. Jane and S. Liz, "Qualitative Data Analysis for Applied Policy Research," in *The Qualitative Researcher's Companion*. Thousand Oaks: SAGE Publications, 2002, pp. 305–329.

[37] K. N. Lemon and P. C. Verhoef, "Understanding Customer Experience Throughout the Customer Journey," *Journal of Marketing*, vol. 80, no. 6, pp. 69–96, 2016.

[38] A. Osterwalder, Y. Pigneur, G. Bernarda, A. Smith, and P. Papadakos, *Value proposition design: How to create products and services customers want. Get started with*, ser. Strategyzer series. Hoboken: Wiley, 2014.