Model-based Hypothesis Engineering for Supporting Adaptation to Uncertain Customer Needs*

Sebastian Gottschalk, Enes Yigitbas, and Gregor Engels

Software Innovation Lab, Paderborn University, Paderborn, Germany
{sebastian.gottschalk,enes.yigitbas,gregor.engels}@uni-paderborn.de

Abstract. To build successful products, the developers have to adapt their product features and business models to uncertain customer needs. This adaptation is part of the research discipline of Hypotheses Engineering (HE) where customer needs can be seen as hypotheses that need to be tested iteratively by conducting experiments together with the customer. So far, modeling support and associated traceability of this iterative process are missing. Both, in turn, are important to document the adaptation to the customer needs and identify experiments that provide most evidence to the customer needs. To target this issue, we introduce a model-based HE approach with a twofold contribution: First, we develop a modeling language that models hypotheses and experiments as interrelated hierarchies together with a mapping between them. While the hypotheses are labeled with a score level of their current evidence, the experiments are labeled with a score level of maximum evidence that can be achieved during conduction. Second, we provide an iterative process to determine experiments that offer the most evidence improvement to the modeled hypotheses. We illustrate the usefulness of the approach with an example of testing the business model of a mobile application.

Keywords: Hypothesis Engineering · Model-based · Customer Need Adaptation · Business Model · Product Features

1 Introduction

To build successful products, the developers have to continuously adapt their product features [7, 11] and the underlying business model [3, 2] to the actual customer needs. Due to the uncertainty in the actual customer needs, the usage of experimentation to validate and disapprove hypotheses with potential customers has been carried out as a promising approach. This experimentation is implemented by different similar concepts like rapid experimentation [2], discovery-driven planning [8], experiment-driven development [7], or data-driven

* 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)

software development [11] in the literature. These concepts are covered by the emerging research field of Hypothesis Engineering (HE) [10] as a subfield of Requirement Engineering (RE), in which different hypotheses have to be tested with evidence. Evidence refers to the certainty with which a hypothesis is either validated or disapproved. Both help developers to adapt their products to actual customer needs. For this, the usage of models can support the visualization and traceability of the HE process. Traceability, in turn, is important to document the adaptation to the customer needs and identify experiments, whose conduction increases evidence. To support this, introduce the hypotheses modeling and mapping support HypoMoMap (see Fig. 1) which can be used by a business developer to iterative test hypotheses about the product features and business models of the product. For this, we need to define a structure and a process.

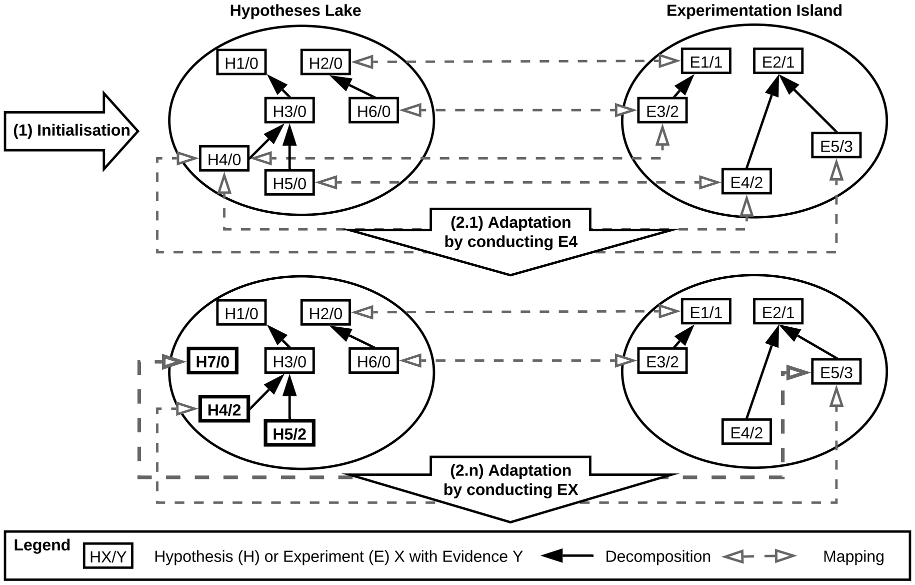

Fig. 1. Overview of the Modeling and Mapping Approach

The structure (i.e. nodes in *Hypotheses Lake* and *Experimentation Island*) consists of hierarchies of hypotheses and experiments. While the hypotheses consist of a state (validated, disapproved, untested) together with a score level of the current evidence and priority, the experiments consist of a score level of maximum evidence that can be achieved during conduction and costs that are generated during conduction. Fig. 1 provides a simplification by assuming that all hypotheses and experiments have the same priority, state, and cost.

The engineering process consists of a single phase for (1) Initialisation and repetitive phases for (2) Adaptation. In the **(1) Initialisation**, the business developer models the first set of hypotheses and corresponding experiments in a structure with hierarchical interrelationships. Moreover, the developer creates a mapping of the hypotheses to all experiments, which conduction can validate or disapprove the hypotheses. As an example in Fig. 1, we have modeled six hypotheses (i.e. *H1* to *H6*) and five experiments (i.e. *E1* to *E5*) together with mappings between them (e.g. *H2* and *E1*). In the repetitive **(2) Adaptation**, our approach selects the experiment that provides the best ratio of increased evidence scores among all hypotheses and generated costs. The developer conducts the experiment and uses the result to update the hypotheses by changing the evidence score of each tested hypothesis. Moreover, the developer adds new hypotheses that can occur by the conduction and remove mappings to experiments, which can not improve the scores of the hypotheses. This adaption phase is repeated until there exists no mapping between hypotheses and experiments, which means that it is not possible to improve the evidence of the hypotheses with the experiments. By conducting *E4* in Fig. 1, the evidence scores of *H4* and *H5* are increased to 2, and *H7* is derived manually from *E4*. Next, we have mapped *H7* to *E3* and removed the mapping from *E4* to *H4* and *H5*, because both are tested with the experiment. Finally, we have removed the mapping from *E3* to *H4* because the maximum score of evidence of *E3* is less than *H4*.

The paper shows the first design cycle of a Design Science Research process [13] and starts with a problem-centered initiation as an entry point of the process. In the *Identify Problem & Motivate* step, we have identified the lack of traceability in Hypothesis Engineering which has the advantage to document the adaptation to the customer needs and identify experiments that conduction increases the gained evidence at most. As motivation, we want to develop a traceability approach, which supports Hypothesis Engineering in an effective and efficient way. This effectivity and efficiency are also our *Define[d] Objectives of Solution.* For this solution, we analyze current approaches of Hypothesis Engineering in terms of their structure and their process in the *Design & Development* step in Section 2. Based on their current limitation, we develop our model-based approach HypoMoMap in Section 3. As *Demonstration*, we apply our approach to the example of testing the business model of a mobile application in Section 4. 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 app. Finally, our *Communication* step will be done with a conclusion in Section 5 and the publication of the paper.

2 Background and Challenges

Hypothesis Engineering [10] is used to validate and disapprove hypotheses by running experiments and gather qualitative and quantitative results from metrics. In their paper [10], the authors found Hypothesis Experiment Data-Driven Development (HYPEX) [11] and Rapid Iterative value creation Gained through High-frequency Testing (RIGHT) [4] as two models for evaluating product features with quantitative metrics. Based on our review of literature, we add Qualitative/quantitative Customer-driven Development (QCD) [12] for qualitative metrics of product features together with Discovery-Driven Planning (DDP) [8]

	HYPEX	**RIGHT**	**QCD**	**DDP**	**TBI**
Hypotheses	✓	✓	✓	✓	✓
with State	✗	✗	✗	✓	✓
with Priorities	✓	✓	✓	✓	✓
with Evidence Scores	✗	✗	✗	✗	✓
with Refinement	✗	✗	✗	✗	✗
Experiments	Feature Test	Feature Test	✓	✓	✓
with Evidence Scores	✗	✗	✗	✗	✓
with Costs	✗	✗	✗	✓	✓
with Refinement	✗	✗	✗	✗	✗
Artifacts	Product Feature	Product Feature	✓	✓	✓
Mapping	Single Experiment	Single Experiment	Single Experiment	✓	Single Experiment
with Metrics	Actual Behavior	Resulting Data	✓	✓	✓

Table 1. Challenges for Modeling from the Hypothesis Engineering Approaches

and Testing Business Ideas (TBI) [2] to consider also the business model. We analyze these five approaches to derive the challenges, we need to consider in our mapping and modeling approach. *HYPEX* [11] is a conceptual model for analyzing product goals to develop a prioritized feature backlog. For this, they assume an expected behavior, implement the feature, and calculate the possible gap between expected and actual behavior based on quantitive observations. If there is a gap, they develop new hypotheses about the feature, otherwise, they continue with the next feature in the feature backlog. *RIGHT* [4] is similar to HYPEX by deriving hypotheses from the business model, test them in a product feature, and use the result to update the business model and product features. *QCD* [12] improves the concept of HYPEX and RIGHT, which use only quantitative measurements, by adding qualitative customer feedback to the continuous experimentation. *DDP* [8] has its foundation in the adaptation of business models. For this, they put the experiments in relation to hypotheses that can be tested together and focus on the created costs. *TBI* [2] is an approach that provides an iterative process from a good idea to a validated business. For this, the authors provide a catalog of 44 experimentation types and focus on the different levels of costs and evidence these experiments provide.

All approaches consist of a process where an initial set of hypotheses is defined, which is further tested by selecting hypotheses and conducting an experiment. This iterative testing is similar to the cycles in LEAN-Development [14], experiments in Experimental Software Engineering, or sprints in SCRUM. All processes except TBI focus on the test of hypotheses with high priority but neglect the overall evidence gain by testing multiple hypotheses through a single experiment. In our approach, we adopt the concept of the LEAN-Development cycle to build an iterative process together with the usage of evidence score as mentioned in TBI. Moreover, the process of incremental refinement can be supported by our previous work in [5], where we intertwine the development of a business model and product functions based on feature models.

None of the existing approaches provides an explicitly modeled structure representing the hypotheses and the experiment. Therefore, we analyze the existing approaches to derive the modeling challenges for our approaches (see Table 1) from their implicit requirements. In this analysis, we can see that HYPEX and RIGHT are quite similar in their modeling structure. Both consist of a hypotheses backlog, where hypotheses together with a priority are stored. To test the hypotheses they can run a single experiment of a feature test, where the usage of the feature is measured. QCD improves this by running different experiments and gather the corresponding feedback. DPP works on the limitation of a single mapping from a hypothesis to an experiment by testing multiple hypotheses with a single experiment. TBI covers the different evidence scores of the hypotheses and experiments but only covers a single hypothesis per experiment. Nevertheless, none of the approaches covers the refinement of hypotheses which can be used to decompose the hypotheses into smaller, easier validatable assumptions. Refinement is one advantage of a model-based approach, which is also used for product features [1], non-functional properties of products [17], and business models [15]. In our approach, we adapt the structure of Goal Oriented Requirements Engineering (GORE) [17]. The adaption of GORE will allow us to easily adapt the formal verifiction of models with techniques like deductive reasoning, which are not the focus of our first design cycle. We refine this structure with states of hypotheses (untested, validated, disapproved), score levels (priorities, evidence scores, costs), and a mapping between hypotheses and experiments.

3 Modeling and Mapping Approach

In this section, we present our hypotheses modeling and mapping approach HypoMoMap by introducing a structure and an engineering process.

3.1 Structure of the Approach

The structure of the approach can be seen in Fig. 2. It consists of the two components of a *Hypotheses Lake* and an *Evaluation Island*. In the **Hypotheses Lake**, different elements of hypotheses (*Validated Hypotheses*, *Disapproved Hypotheses*, *Untested Hypotheses*) can be modeled with corresponding evidence score (e.g. *Kids* with a score of 2) and a priority (e.g. *Kids* with a priority of 2). The scores and priorities can have a value from 1 (lowest) to 5 (highest). The hypotheses interrelate to each other in a hierarchical order (e.g. *Customer* is decomposed into *Kids*). The hierarchy consists of *AND/OR-Relationships*. While with an *OR-Relationship* each hypothesis in a higher hierarchy level can be validated with a single lower interrelated hypothesis (e.g. *Customer* or *Kids*), the *AND-Relationship* provides only validation to a higher hypothesis by validating all lower interrelated hypotheses (e.g. *Kids* into *Age* and *Gender*). Moreover, each hypothesis is mapped to the experiments which can be used to validate or

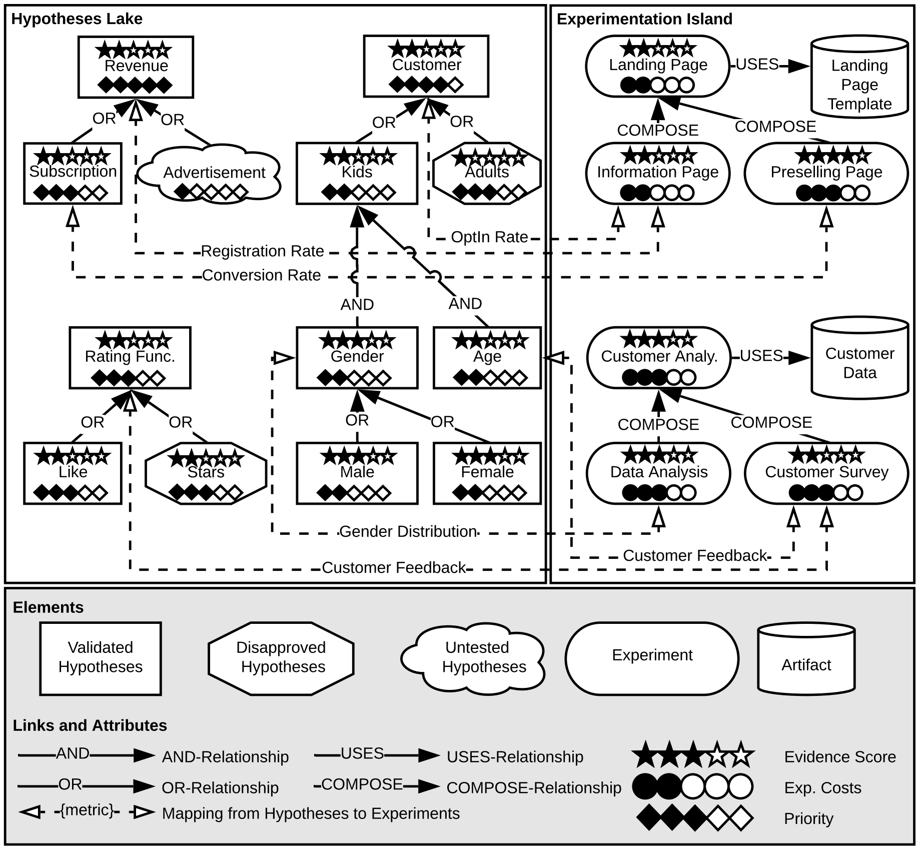

Fig. 2. Visual Notation of HypoMoMap

disprove the hypothesis (e.g. validate *Gender* with *Data Analysis*). For the mapping, we explicitly model the metric, which is used to validate the hypothesis.

On the **Experimentation Island**, different elements of *Experiments* (e.g. *Data Analysis*) and corresponding *Artifacts* (e.g. *Customer Data*) are modeled. Each experiment has a maximum score of evidence that can be achieved by conducting the experiment and costs that are generated by the conduction (e.g. *Data Analysis* with a score of 3 and costs of 3). Moreover, each experiment can use different provided artifacts (e.g. *Customer Analysis* uses *Customer Data*). Moreover, the experiments are decomposed to more accurately experiments.

3.2 Engineering Process of the Approach

The engineering process of the approach is depicted in Fig. 3. It consists of the two phases of the Initialisation and the Adaptation. In the **(1) Initialisation** phase, the business developer has to *(1.1) define hypotheses and experiments*

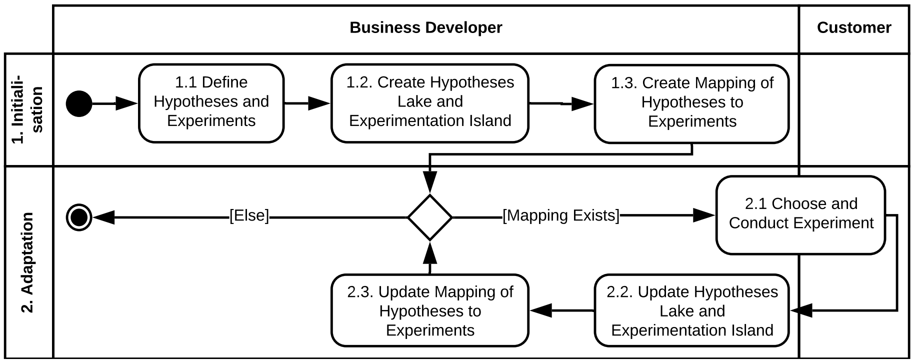
Business Developer
Customer
1. Initiali-sation
2. Adaptation
1.1 Define Hypotheses and Experiments
1.2. Create Hypotheses Lake and Experimentation Island
1.3. Create Mapping of Hypotheses to Experiments
[Else]
[Mapping Exists]
2.1 Choose and Conduct Experiment
2.2. Update Hypotheses Lake and Experimentation Island
2.3. Update Mapping of Hypotheses to Experiments

Fig. 3. Engineering Process of HypoMoMap

at the beginning. While the hypotheses of the business model and the product features can be derived from the business strategy and its product goals [11] or developed user stories [9], the experiments can be chosen from existing libraries [2] or own experiences. Based on this, the developer needs to *(1.2) create the Hypotheses Lake and Experimentation Island*. For this, the hypotheses and experiments are structured in interrelated hierarchies. Moreover, each hypothesis has to be decomposed in separately validatable units, which are connected with *AND/OR-Relationships* (e.g. *Rating Function* into *Like* or *Stars*). With the term validatable units, we mean to split the hypotheses in small assumptions, which can be validated with high evidence with an experiment (e.g. a *Stars* rating can be easier validated than *Rating Function*). Moreover, the validation of these small assumptions can support the validation/disapproval of different hypotheses. While at the beginning all hypotheses got the type *Untested* with no estimated evidence, the developer has to estimate the maximum score of evidence and generated costs of the experiments (e.g. *Landing Page* with a score of 2 and costs of 3). To support this process, the developer can use the experimentation catalog, which has proposed in [2]. Moreover, the developer has to prioritize the hypothesis to consider the most important hypotheses at the beginning. After that, the developer needs to *(1.3) create the mapping of hypotheses to experiments* where the hypotheses are mapped to experiments that can be used for testing the hypotheses. Here it is important to choose multiple experiments for the hypotheses to test these different hypotheses with the same experiment.

In the **(2) Adaptation** phase, the developer has to *(2.1) choose and conduct an experiment*. Because this choice is a critical part, we provide different techniques, which can be used for different situations because of their outcomes: With *Highest Priority*, a hypothesis with the highest priority is chosen. This setting is also used by other models [11, 4] and ensures that the most important assumptions are validated first. Nevertheless, it has the disadvantage that more iteration cycles are needed for the validation of the model. For *Best Estimated Ratio*, the developer looks at each experiment and chooses the maximum of hypotheses evidence gain which can be validated in a single execution of the experiment in comparison to the costs of the experiment. The hypotheses evidence gain is defined by cumulated evidence scores between the current evidence score and the estimated evidence score with the experiment of all selected hypotheses. This setting is used to maximize validated learning by considering the costs but has the disadvantage to not consider the most critical hypotheses at the beginning. With *Best Discounted Ratio*, we discount the hypotheses evidence gain of single hypotheses in the Best Estimated Ratio with their priority to focus the validated learning on the most important assumptions. Nevertheless, it has the disadvantage that the calculation is quite complex without tool support.

After that, the developer needs to *(2.2) update Hypotheses Lake and Experimentation Island*. For this, each tested hypothesis is typed as *Validated* or *Disapproved* with the evidence score of the experiment. Here it is also possible to use a lower evidence score if the experiment turned out as less effective as expected. Additionally, the evidence scores are propagated to the higher levels of the hierarchy. While with an *OR-Relationship* the higher hypothesis is set to the highest evidence score of the lower hypothesis, the *AND-Relationship* assumes the lowest evidence score of all lower hypotheses. Moreover, new hypotheses that are derived by the experiment can be added to the model. In the end, the developer needs to *(2.3) update the mapping of hypotheses to experiments*. For this, the developer removes all mappings between hypotheses and experiments which do not provide further evidence gain adds new mappings which are derived from the experiment. At every adaptation of the model, it is important to save the existing model together with the results of the experiments to provide traceability. The adaptation phase is repeated until no experiment that can improve the evidence in the hypotheses.

4 Instantiation

We illustrate the usefulness of our approach by testing the business model of a mobile to-do app. The main functionality of the app is to provide customers an organization of their daily tasks. For this, we structure the hypotheses according to the customer-side (Customer Segments, Value Propositions, Customer Relationships, Channels, Revenue Streams) of the Business Model Canvas and model the product features as different value propositions. Because of the multitude of hypotheses, we develop a preliminary version of a tool-support based on [6].

In the **(1) Initialisation**, we need to define the hypotheses we want to validate together with corresponding experiments. For developing a new business model, Teece [16] argues that a deep analysis of the market, the existing competitors and potential customers are needed. We apply this by analyzing the market of mobile applications, the existing competitors (e.g. Microsoft ToDo, Any.do, Todoist), and three selected customer segments (Business Improver, Fitness Improver, Life Improver). The full analysis is shown in [6]. Out of the analysis, we generate 96 hypotheses. We structure them into the five main hypotheses of the customer-side on the canvas (e.g. Customer Segments →Target Group

→Business Improver; Value Propositions →Automatic Task Scheduling, Workflow Tracking) so that the overall goal is to create a canvas where all building blocks are validated. While the features for the competitive advantage get a high priority to test them at the beginning, the existing features of competitors are already validated by them and therefore receive less priority. Moreover, we use the test catalog of Bland et al. [2] to create five experiments (Landing Page, Clickable Prototype, Single-Feature Minimum Viable Product, Split-Test, Customer Survey) with three artifacts (Facebook-Page, Landing Page, Mockup) to derive a mapping from the hypotheses (e.g. Customer Segment →Landing Page, Workflow Tracking →Clickable Prototype). In the **(2) Adaptation**, we need to test the hypotheses by conducting experiments. We use the strategy of *Best Discounted Ratio*, to maximize the validated learning by considering also the experimentation costs. Here, we saw that many hypotheses can be combined and test together in single experiments. An example of this is the Split-Test with a Facebook-Page to test hypotheses about the Customer Segments and Facebook as a potential channel. Moreover, a Landing Page with the corresponding artifact can be used to test hypotheses about the Revenue Streams and the Channels. By choosing the experiments identified with our modeling and mapping approach, we could efficiently reduce the number of adaptation loops to validate the model.

We conduct a preliminary evaluation by analyzing the effectiveness and efficiency of our approach. The effectiveness is supported by the explicit modeling of hypotheses and experiments. With this modeling, it is possible to gather an overview of the currently tested hypotheses and whether important assumptions not tested. Moreover, by refining these hypotheses into small validatable units, the validation of fuzzy hypotheses can be replaced. The efficiency is supported by the explicit mapping of hypotheses to experiments. With this mapping, it is possible to validate different hypotheses at the same time by conducting a single experiment. Moreover, by analyzing old hypotheses and experiments, the selection of matching experiments for new hypotheses is supported. A potential limitation is the manual modeling complexity, which is time-consuming due to the multitude of hypotheses and corresponding experiments the developer can choose. To solve this limitation, we want to classify different hypotheses types with a literature review and map these types to suitable experiments. With this, it is possible to provide tool-support, which supports the modeling of hypotheses and experiments together with an automated mapping between them.

5 Conclusion

Hypothesis Engineering (HE) is used to build successful products by conducting experiments to test hypotheses about uncertain customer needs. We introduce a modeling and mapping support for HE, which provides traceability in terms of documentation of the adaptation of customer needs and identification of experiments which provides the most evidence gain in the customer needs. For this, we model the hypotheses with evidence scores in interrelated hierarchies and map them to corresponding experiments with maximum evidence and costs. By conducting the identified experiments based on the best ratio of increased evidence and estimated costs, we support the process of improving the evidence in the customer needs. In the future, we plan to classify different hypotheses types with a mapping to experiments to support a questionnaire-based engineering process for automated experiment derivation. Moreover, we want to evaluate the applicability of our approach with a student seminar on lean development.

References

1. Apel, S., Batory, D., Kästner, C., Saake, G.: Feature-Oriented Software Product Lines. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
2. Bland, D.J., Osterwalder, A.: Testing business ideas. John Wiley & Sons, Hoboken (2020)
3. Chesbrough, H.: Business Model Innovation: Opportunities and Barriers. Long Range Planning **43**(2-3), 354–363 (2010)
4. Fagerholm, F., Sanchez Guinea, A., Mäenpää, H., Münch, J.: The RIGHT model for Continuous Experimentation. J. Syst. Softw. **123**, 292–305 (2017)
5. Gottschalk, S., Rittmeier, F., Engels, G.: Intertwined Development of Business Model and Product Functions for Mobile Applications: A Twin Peak Feature Modeling Approach. In: Software Business, pp. 192–207. Springer (2019)
6. Gottschalk, S., Rittmeier, F., Engels, G.: Hypothesis-driven Adaptation of Business Models based on Product Line Engineering. In: International Conference on Business Informatics (CBI): (forthcoming). IEEE (2020)
7. Lindgren, E., Münch, J.: Raising the odds of success: the current state of experimentation in product development. Inf. Softw. Technol. **77**, 80–91 (2016)
8. McGrath, R.G.: Business Models: A Discovery Driven Approach. Long Range Planning (43), 247–261 (2010)
9. Melegati, J., Wang, X.: QUESt: new practices to represent hypotheses in experiment-driven software development. In: International Workshop on Software-intensive Business (IWSiB). pp. 13–18. ACM (2019)
10. Melegati, J., Wang, X., Abrahamsson, P.: Hypotheses Engineering: First Essential Steps of Experiment-Driven Software Development. In: RCoSE/DDrEE. pp. 16–19. IEEE (2019)
11. Olsson, H.H., Bosch, J.: The HYPEX Model: From Opinions to Data-Driven Software Development. In: Continuous Software Engineering, vol. 14, pp. 155–164. Springer (2014)
12. Olsson, H.H., Bosch, J.: Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation. In: Software Business, vol. 210, pp. 154–166. Springer (2015)
13. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. J Manag Inf Syst **24**(3), 45–77 (2007)
14. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, USA (2014)
15. Samavi, R., Yu, E., Topaloglou, T.: Strategic reasoning about business models: a conceptual modeling approach. Inf Syst E-Bus Manage **7**(2), 171–198 (2009)
16. Teece, D.J.: Business Models, Business Strategy and Innovation. Long Range Planning **43**(2-3), 172–194 (2010)
17. van Lamsweerde, A.: Goal-oriented requirements engineering: a guided tour. In: International Symposium on Requirements Engineering. pp. 249–262. IEEE (2001)