

Situation-specific Business Model Development Methods for Mobile App Developers*

Sebastian Gottschalk, Enes Yigitbas, Alexander Nowosad, and Gregor Engels

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

Abstract. The development of effective business models is an essential task in highly competitive markets like mobile ecosystems. Existing development methods for these business models do not specifically bring into focus that the development process profoundly depends on the situation (e.g., market size, regulations) of the mobile app developer. Here, a mismatch between method and situation can lead to poor resource management and longer development cycles. In software engineering, software projects use situational method engineering to configure a development method out of a method repository based on the project situation. Analogously, we support creating situation-specific business model development methods with a method base and new user roles. Here, the method engineer obtains the domain expert’s knowledge and stores it in the method base as elements, building blocks, and patterns. We derive the expert knowledge from a grey literature review on mobile development processes. After this, the method engineer constructs the development method based on the described situation of the business developer. We provide an open-source tool and evaluate it by constructing a local event platform’s business model development method.

Keywords: Business Model Development, Situational Method Engineering, Mobile App, Business Model Development Tools

1 Introduction

The development of effective business models, defined by Osterwalder et al. as “the rationale of how the organization creates, delivers, and captures value” [23], is an essential task for a company to stay competitive. This is one of the results of the GE Innovation Barometer 2018 [14], 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” [14]. An important reason for this is that customers expect solutions for perceived

* This work was partially supported by the German Research Foundation (DFG) within the CRC “On-The-Fly Computing” (CRC 901, Project Number: 160364472SFB901) and the German Federal Ministry of Education and Research (BMBF) through Software Campus grant (Project Number: 01IS17046).

Preprint, cite this paper as:

Gottschalk S., Yigitbas E., Nowosad A., Engels G. (2021) Situation-Specific Business Model Development Methods for Mobile App Developers. In: Augusto A., Gill A., Nurcan S., Reinhartz-Berger I., Schmidt R., Zdravkovic J. (eds) Enterprise, Business-Process and Information Systems Modeling. BPMDS 2021, EMMSAD 2021. Lecture Notes in Business Information Processing, vol 421. Springer, Cham. https://doi.org/10.1007/978-3-030-79186-5_17

The final authenticated version is available online at:
https://doi.org/10.1007/978-3-030-79186-5_17

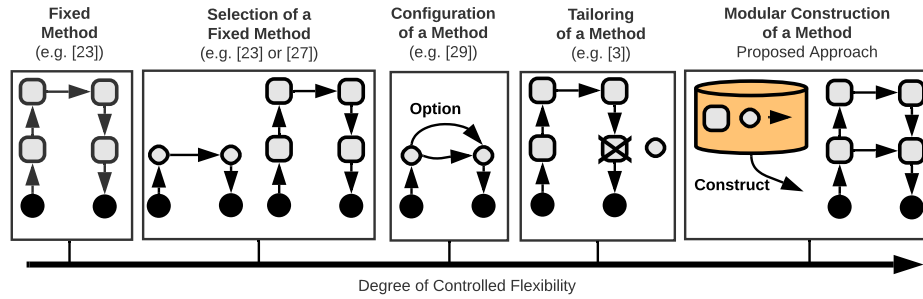


Fig. 1. Categorization of *Business Model Development Methods* (adapted from [20])

needs rather than just products [30]. These perceived needs correspond to the potential effect that the business model can be often more important than the latest technology of the product [6]. Attractive markets for companies are mobile ecosystems. As highlighted by the AppAnnie’s State in Mobile 2021 study [1], these ecosystems provided 218 billion app downloads that led to 142 billion dollar revenue just in 2020. Nevertheless, app developers compete with their app against millions of other apps over the users’ usage time. Therefore, effective business models are essential for staying successful in these markets.

The process of business model development is a creative task that often requires the collaboration of different stakeholders within a company [7, 8]. The process needs a deep understanding of the market, the existing competitors, and potential customers [30]. From this analysis, different possible business models have to be created in this process, and the corresponding assumptions have to be validated within the market [26]. This validation, for example, can be done by conducting experiments with the potential customers of the product [22]. To support the business model development, different domain experts propose various methods to develop such business models in the form of development processes (e.g., [22, 9]) and method repositories (e.g., [3, 27]). However, as shown in Fig. 1, most approaches (e.g. [22, 26]) provide *Fixed Methods* that do not focus on the mobile app developer’s situation. This can lead to poor resource management and longer development cycles. Some approaches, see Fig. 1, try to cover this lack of *Controlled Flexibility* by providing different *Configurations of a Method* [28] or basic *Tailoring of a Method* [3]. Nevertheless, these one-size-fits-all methods are not capable of covering all relevant contextual factors. Examples for the factors could be the own company (e.g., low vs. high financial resources), the market (e.g., mass vs. niche market), the competitors (e.g., high vs. low number of competitors), or the potential customers (e.g., adults vs. children).

This *Modular Construction of a Method* to the situation, in turn, is possible with Situational Method Engineering (SME). SME is a subfield of method engineering and creates new methods out of existing methods based on the situational context where the method is applied [21]. To consider the situation in the business model development methods, we apply SME to business mod-

eling with the roles of a domain expert, a method engineer, and a business developer. Here, the method engineer uses the method knowledge of the domain expert to create a method base. This method base consists of method elements (e.g., stakeholders, tasks) that are composed into method building blocks (e.g., conduct customer interview as a task with the customer as a stakeholder) and structured according to method patterns (e.g., conduction of customer interviews before the development of prototype). The method engineer can then use this method base to construct development methods according to the business developer’s described situation. To apply our approach for mobile app developers, we construct a method base by conducting a grey literature review [12] of articles for developing business models of mobile apps. We extract the explicit and implicit steps of the development process and map them to their method elements, method building blocks, and method patterns. For using this method base in practice, we develop an open-source tool for the creation of situation-specific business model development methods. Moreover, we evaluate the whole approach by a case study of constructing a method of an event platform app.

2 Background and Related Work

2.1 Business Model Development

Business model development (BMD) is a continuous and challenging task, which often uses creativity and collaboration between different stakeholders [8]. It consists of several phases (i.e., initiation, ideation, integration, implementation in the BMI Magic Triangle [10]) where different possible business models have to be created, and the corresponding assumptions have to be validated within the market [26]. This validation, in turn, can be done by conducting experiments with the potential customers of the product/service [22]. Here, a method repository with different experiment sequences based on the type of business is introduced to provide flexibility in the method construction [3]. Moreover, the flexibility can be supported by alternative choices for process steps inside the method [28]. Nevertheless, those approaches focus on high abstraction levels and one-size-fits-all methods that cannot cover all relevant contextual factors. Artifacts and tools can support those approaches. One artifact is the Business Model Canvas [23], which divides the business model into nine building blocks where each building block consists of different elements. Moreover, manual tools like pattern databases [13] or software-based tools [29] can be used.

These software-based tools are often called Business Model Development Tools (BMDT) and provide different guidance levels to develop new and improve existing business models [29]. Here, earlier examples of these tools in the literature focus on the visualization of the business model [11] or simple financial assessments [16]. An analysis of business modeling tools in practice [29] shows that those tools focus on the design of business modeling but not on the actual decision support. Nevertheless, a shift from simple design support of business modeling to real decision support by these tools needs to be done [24]. BMDTs already introduce possible parts of decision support in research. Virtual Business

Model Innovation [7] introduced a concept of a BMDT with the four phases of analysis, design, implementation, and management. The Framework for Analysis of Business Model Management does a similar division into phases [31]. To provide decision support in different phases, the Business Model Developer [4] supports the configuration of the business model together with validation in terms of static analysis. Moreover, a domain-specific modeling method for configuring the business models and generating business plans is introduced [32]. Nevertheless, the proposed tools focus on a high abstraction level to propose a generic solution for all phases or lower abstraction of single phases or specific application domains. Moreover, none of them provides the flexibility of a method repository to construct the used method.

2.2 Situational Method Engineering

A method is a description of how to perform a procedure systematically [5]. For that, a method can be decomposed into different parts. Here, a method fragment is a reusable atomic block of a method that can have a process (called work unit), a product (called work product), or a producer focus [5]. This focus can further refined (e.g., product into activity) due to the application of the whole method. A method component consists of inputs and outputs of work products together with a process to transform the input into the output [21]. We will use the naming method elements for a method fragment and method building block for a method component to stick to the business model terminology. Moreover, we use the term of method pattern to note sequences of method building blocks. Method Engineering, in turn, is a research discipline to design, construct, and adopt these methods for the development of information systems [5]. If the method is developed for a specific project and depends on the project's situational context, it is called Situational Method Engineering (SME) [21].

SME has its origin in creating software development methods and typically consists of the two roles of a method engineer and a project manager. Here, the method engineer analyzes various methods and stores them in a method base. After that, the method engineer identifies the project's situational factors and constructs a suitable method of the method base. This method, in turn, is then enacted by the project manager to manage the project. While most of the existing approaches focus on developing software products, some also include business-related parts to their methods base. Here, a study [25] investigates how the agility of different methods can be used in method engineering. A case study [2] identifies different situational factors for the business, the customer (e.g., number of customers), market characteristics, product characteristics, and stakeholder involvement for phases in product management. An SME approach of IoT development methods [15] also includes business-related (e.g., regulations) and customer-related (e.g., domain experience) situational factors together with business-related (e.g., IoT Canvas) work products. Nevertheless, those approaches cover the business aspect as one side aspect of the product development process. They do not consider the BMD as a separate continuous process with its characteristics like hypothesis-driven experimentation [3].

3 Research Approach

This paper introduces situation-specific business model development methods. We apply the approach for mobile app developers by providing a method base on practical knowledge outside the research world. To gather such knowledge systematically and repeatably, grey literature reviews (GLRs) have been introduced as a promising approach in the last years [12]. Here, we follow the guidelines according to Garousi et al. [12], who structure the GLR in the three phases of (1) *Planning the Review*, (2) *Conducting the Review*, and (3) *Reporting the Review*. While this section considers just the most essential aspects, the complete review with sources and results is provided in our technical report [19].

In (1) *Planning the Reviews*, the need for a GLR needs to be motivated together with the explicit formulation of the research question the study aims to answer. Out of the research question, the search string and related inclusion and exclusion criteria are determined. To motivate the need for the GLR, we have used the checklist of Garousi et al. [12]. Here, we concluded the need for a GLR based on the subject’s complexity and the lack of practical experience in the formal literature. For filling the method base, we have defined the following research question: *What are the main business model development steps that need to be done by a mobile app developer?* To answer this question and by testing different search terms with the terminology in business model development, we have defined the following search string: *app AND (business model OR idea) AND (test OR validate OR develop)*. We include articles in English where the URL is accessible and directly connected to the research question. We exclude articles that provide no business model development process, not relate to mobile apps, are posted in forums, or are presented in the non-textual form.

In (2) *Conducting the Review*, the review needs to be conducted by considering the search process, the source selection, the quality assessment, the data extraction, and the data synthesis. In the search process, we applied the search string to the Google search engine on January 19th, 2021, and anonymized our browser data for a maximum of objective results. We exported the first 50 results of the search results and manually scanned the inclusion and exclusion criteria that result in 38 articles. For the quality assessment, the essential criteria were the understandability of the processes. Moreover, the method base will contain links to the articles so that mobile app developers can convince themselves of the quality. After analyzing the first results, we found the division between atomic blocks (i.e., method elements), combined steps (i.e., method building blocks), and their different orderings (i.e., method patterns), which we choosed as initial attributes. We were iteratively refining our initial attribute method elements into the atomic parts of tasks, types, stakeholders, situational factors, artifacts, and tools. Based on these atomic elements, we extracted all method elements in the data extraction. Finally, in the data synthesis, which is shown in the technical report, we combined these elements into building blocks and patterns.

In (3) *Reporting the Review*, we documented the review results. We do this by publishing the highlights of our results in this research paper and providing access to the whole method base in our technical report [19].

4 Creation of Business Model Development Methods

In this section, we describe the creation of situation-specific business model development methods. As shown in Fig. 2, our approach adds the two roles of the *Domain Expert*, who provides the method knowledge, and the *Method Engineer*, who structures the method knowledge, in addition to the *Business Developer*, who describes his situation. Moreover, we have the two stages of (1) *Creation of Method Base* and (2) *Construction of Development Method*. After giving an overview of both stages, we will explain them in more detail and introduce a tool-support for the method development.

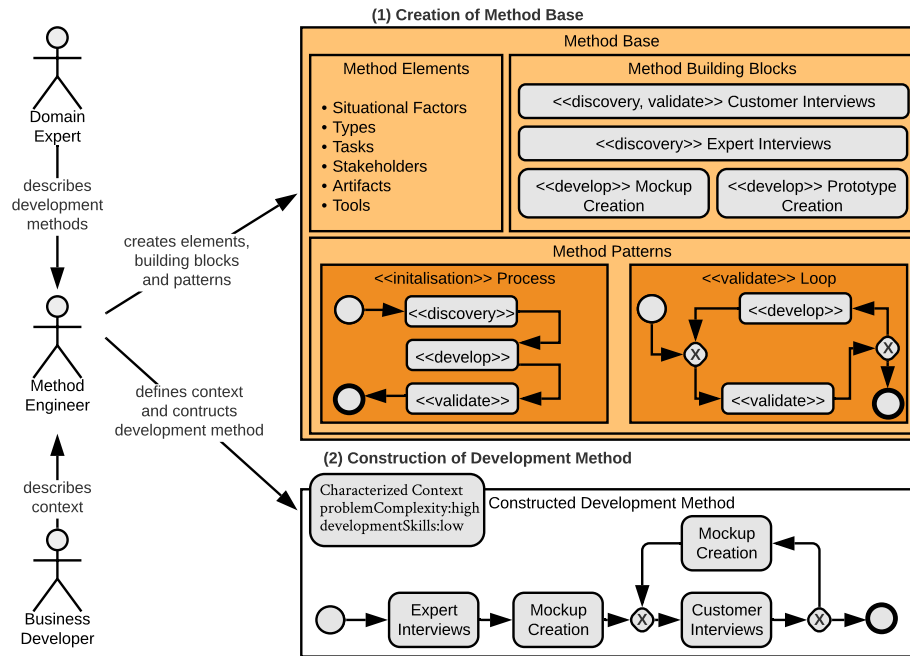


Fig. 2. Exemplary construction of a development method for the *Business Developer* by the *Method Engineer* based on the method knowledge of the *Domain Expert*

In the (1) *Creation of Method Base*, we have the *Method Base* consisting of *Method Elements*, *Method Building Blocks*, and *Method Patterns*. *Method Elements* are atomic parts of the method consisting of the possible *Situational Factors*, different *Types* of methods, performed *Tasks*, involved *Stakeholders* created *Artifacts* and used *Tools*. Multiple elements are combined to *Method Building Blocks* and structured to methods using *Method Patterns*. The content of the *Method Base* is created by the *Method Engineer* based on the knowledge of the *Domain Expert*. For example, the *Method Base* in Fig. 2 consists of the building blocks of *Expert Interviews* and *Mockup Creation*.

In the (2) *Construction of Development Method*, the *Constructed Development Method* is created out of the *Method Base*. For this, the *Business Developer* describes the current context. The *Method Engineer* uses this information to define the context according to the *Situational Factors* and constructs the method out of the combined *Method Patterns*. Moreover, the *Method Patterns* are filled with the *Method Building Blocks*. Here, the *Constructed Development Method* is characterized by a high *problemComplexity* and low *developmentSkills*. Therefore, *Expert Interviews* and *Mockup Creation* are selected from the *Method Base*.

4.1 Creation of Method Base

The first stage of our approach is the creation of the *Method Base*. This is done by the *Method Engineer* who is formalizing the *Method Elements*, *Method Building Blocks*, and *Method Patterns* out of the knowledge of the *Domain Expert*. In our case, we use the results of the grey literature review as domain knowledge. The usage of the grey literature review, in turn, allows us to accumulate knowledge from the practice of different experts in the field of business model development. While in this paper, we cover just the most important findings of our review, the full source of 234 method elements, 57 method building blocks, and 28 method patterns is available in our report [19]. Moreover, the whole *Method Base* can be used in the tool. An excerpt of this *Method Base* is shown in Fig. 3 and consists of *Method Elements*, *Method Building Blocks*, and *Method Patterns*.

The *Method Elements* are atomic parts of the methods that can be divided into tasks, types, stakeholders, situational factors, artifacts, and tools. *Tasks* are the main activities that need to be performed during the business model development. Here, various tasks on different granularity levels (e.g., *Conduct Interviews*, *Create Prototype*) are presented in the literature. *Types* are used to structure the building blocks and patterns. They can be divided into functional types, structure types, and method types. *Functional Types* are used to add additional logic to the method patterns. Here, the type *initialisation* can be used to represent the start patterns of the approach, and the type *generic* can be used to specify patterns that can be inserted into any other pattern, regardless of the type. Next, *Structure Types* are used to create a structure between different patterns. Finally, *Method Types* are used to classify different types of building blocks. *Stakeholders* are the persons who are involved in the building blocks. Here, we can divide between stakeholders of the own *Company* (e.g., *Business Developer*), existing *Partners* (e.g., *Development Agency*), and potential *Users* (e.g., *Customer*). *Situational Factors* are used to classify in which context a building block or a pattern can be applied. Here, we found the different categories of the own *Company* (e.g., *developmentSkills*), the identified *Problem* (e.g., *problemComplexity*), the potential *Customer* (e.g., *customerGroup*), the targeted *Market* (e.g., *marketSize*), the developed *Product* (e.g., *productComplexity*), and the state of the *Phase* (e.g., *productValidity*) for structuring the factors. Moreover, the values of the factors can be nominal variables (e.g., *mass* or *niche* for *marketSize*) or ordinal scales (e.g., *low* < *medium* < *high* for *developmentSkills*). *Artifacts* are working products that are created and modified

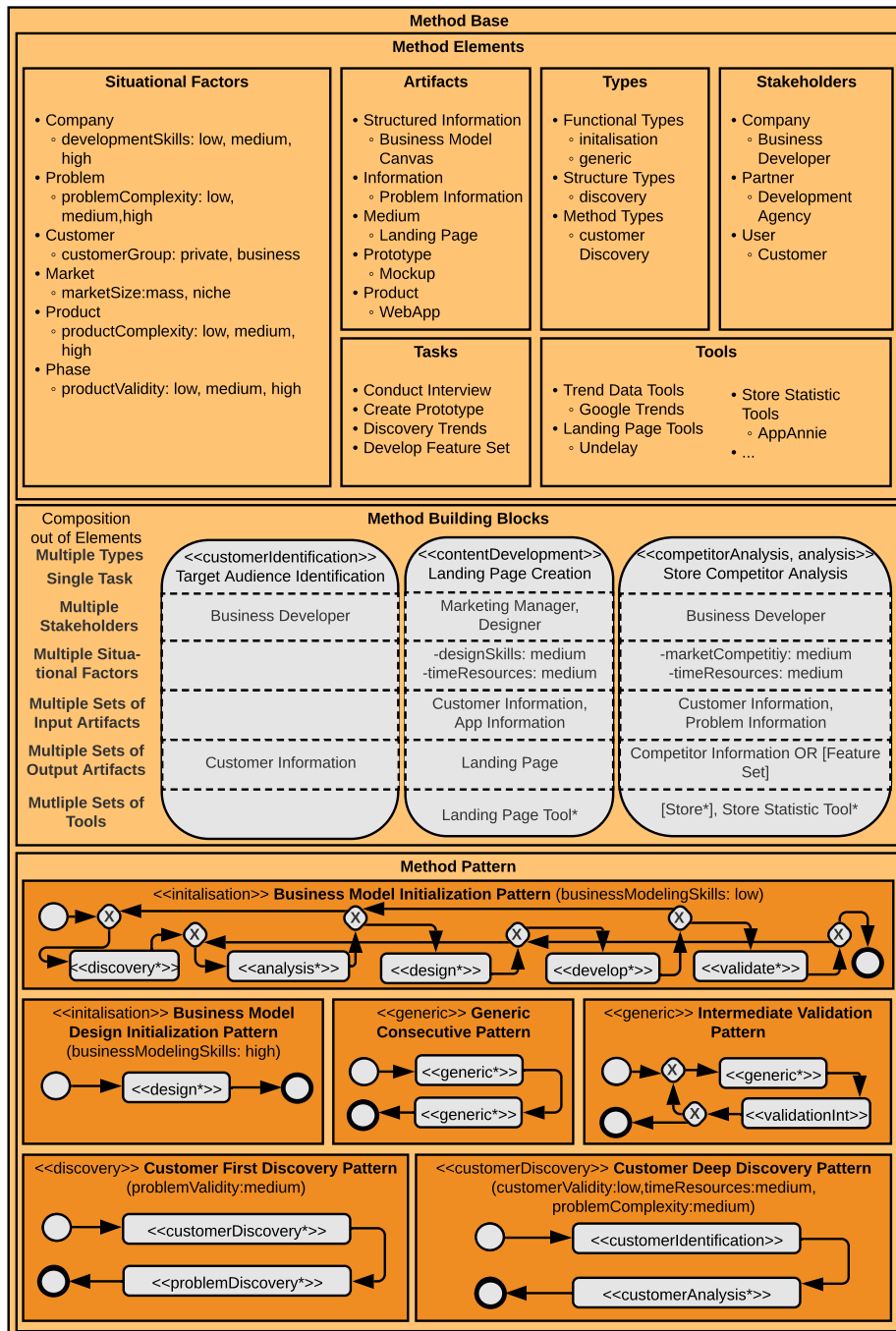


Fig. 3. Excerpt of the Method Base created by the Grey Literature Review

during the business model development process. Here, we found the categories of *Structured Information* (e.g., *Business Model Canvas*), unstructured *Information* (e.g., *Problem Information*), used *Mediums* (e.g., *Landing Page*), created *Prototypes* (e.g., *Mockups*), and developed *Products* (e.g., *WebApp*). *Tools* are used to support the performing of activities. For this, there can be tools for supporting the tasks or creating and modifying artifacts.

The *Method Building Blocks* are combined out of the different elements. Here, a single *Task* is included in the method block together with multiple *Types* and multiple *Situational Factors*. Moreover, for the *Artifacts* we allow the modeling of *Input Artifacts* and *Output Artifacts* so that *Method Building Block* can be interpreted as the transformer of the *Artifact*. For the *Stakeholders*, *Artifacts* and *Tools*, we allow the definition of multiple sets, where the concrete set needs to be selected during the construction of the method (e.g., set with single elements of *Expert* or *Customer* as *Stakeholder* for *Conduct Interview* as a *Task*). Here it is also possible to define just a category of the element so that any element of this category can be selected during the construction (e.g., category of *User* as *Stakeholder* for *Conduct Interview*). This, in turn, allows us to reduce the manual modeling effort in the *Method Base* while keeping the method space flexible in a targeted manner. Last, we provide a list function, which means that multiple elements can be used within the building block (e.g., multiple *Feature Sets* out of a *Competitor Analysis*). With *Target Audience Identification*, *Landing Page Creation* and *Store Competitor Analysis*, three example building blocks are shown in Fig. 3. While the *Target Audience Identification* is completely fixed, the *Landing Page Creation* allows the choice of a tool at the construction (marked with *). Moreover, the *Store Competitor Analysis* allows the output artifacts of competitor information for multiple artifacts of a *Feature Set* (marked with []).

The *Method Patterns* are used to assemble the building blocks in a structured way. For this, each pattern has a type, a name, additional situational factors, and a process part in the form of a BPMN model. Here, the *initialisation* type refers to patterns that can be used at the beginning of the construction. In our example, this is the construction of a process with all extracted phases (e.g., *Business Model Initialization Pattern*) for beginners in business model development (e.g., *businessModelingSkills* are *low*) or single phases (e.g., *Business Model Design Initialization Pattern*) for experts in business model development (e.g., *businessModelingSkills* are *high*). Moreover, the *generic* type can be made to classify patterns that can be used in every other pattern to increase the number of steps (e.g., *Generic Consecutive Pattern*) or provide additional validation to a single step (e.g., *Intermediate Validation Pattern*). Finally, the placeholders in the pattern can be filled with method building blocks (marked with << *type* >>) or a choice of building blocks and patterns (marked with << *type** >>).

4.2 Construction of Development Method

The second stage, as shown in Fig. 2, is the construction of the development method. This is done by the *Method Engineer* who is defining the context and constructing the method out of the description of the *Business Developer*.

For the definition of the *Context*, a subset of the *Situational Factors* of the *Method Elements* has to be selected with corresponding values of the situation. These provided values are used to give recommendations for *Method Building Blocks* and *Method Patterns* by ordering both according to the distance between the provided factors of the context and the needed factors of the buildings blocks and patterns. Thus, the weighted distance of all factors selected for the context and provided by the building blocks and pattern is calculated. While nominal variables for factors need concrete matching (e.g., *mass* vs. *niche* for *marketType* has distance 1), ordinal scales are weighted according to the scale (e.g., provided *developmentSkills: medium* and required *developmentSkills: high* has distance 1/2). Moreover, we automatically matching values in the including direction (e.g., provided *developmentSkills: medium* and needed *developmentSkills: low* has distance 0) to cover building blocks and patterns outperform the context.

For the construction of the *Development Method*, a structured selection of building blocks and patterns based on the weighted distance is made. In the beginning, all patterns with the *initialisation* type out of the *Method Base* are recommended by the approach. Next, the placeholders inside this pattern need to be filled with *Method Building Blocks* of the required or the generic type (marked as `<< type >>` and `<< generic >>`). During this filling, also abstract building blocks have to be instantiated with concrete elements from the *Method Base*. Moreover, it is possible to fill the placeholders with a mixture of *Method Building Blocks* and *Method Patterns* (marked as `<< type* >>` and `<< generic* >>`). This process is repeated until all placeholders in the method are filled out with building blocks or patterns. To allow a continuous expansion of the method, it is also possible to extend the first pattern by another pattern of type *initialisation*. After the complete method has been constructed, the last step is to check the quality of the method. To ensure the quality of the method, the approach checks if every abstract building block is filled out, and for every input artifact, there is an output artifact created before in the method.

4.3 Tool-Support for Method Development

To use our approach in practice, we have developed a corresponding tool-support. The open-source tool BMDL Method Modeler, which is based on our BMDL Feature Modeler [18], can be accessed online¹ or downloaded². The BMDL Feature Modeler, in turn, has been developed to derive business models out of an existing business domain knowledge. In the past, we have also extended the tool to model the knowledge of different experts [17]. Based on Angular and the BPMN.io Framework by Camunda, we have created a method development tool for the *Method Engineer*. The tool can be divided into the (1) *Creation of Method Base* and the (2) *Construction of Development Method*.

The (1) *Creation of Method Base* is shown in Fig. 4. Here, the *Method Engineer* can model the *Situational Factors* (see Fig. 4 (a)), the *Stakeholders*, the

¹ Use the Tool: <https://sebastiangtts.github.io/bmdl-method-modeler/>

² Download the Tool: <https://github.com/SebastianGTTS/bmdl-method-modeler>

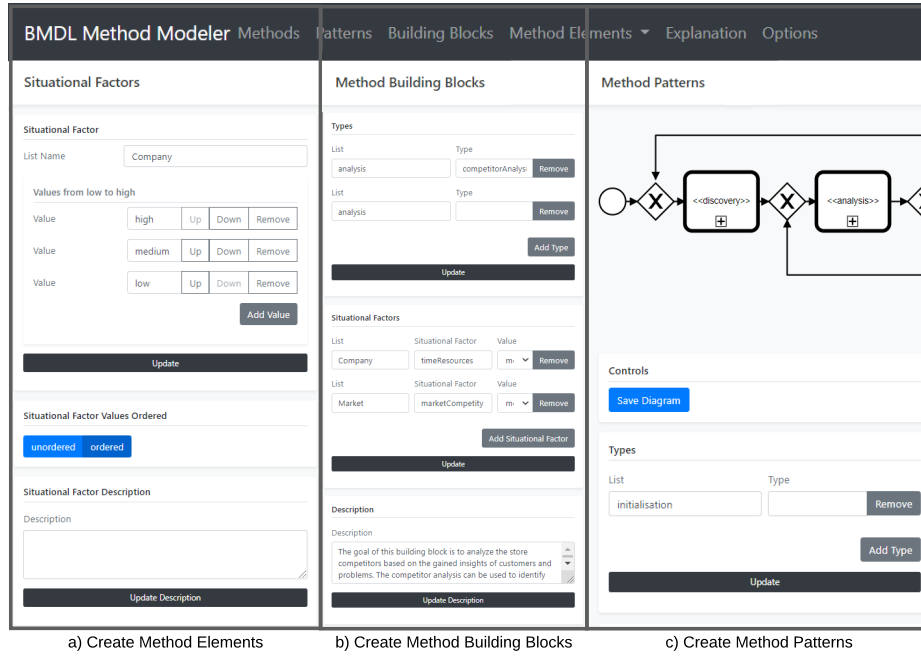


Fig. 4. Creation of the *Method Base*

Types, the *Artifacts* and the *Tools* as *Method Elements*. After that, he can create *Method Building Blocks* by choosing corresponding elements for a *Task* (see Fig. 4 (b)). Moreover, he can define *Method Patterns* with corresponding placeholders for *Types* and recommended *Situational Factors* (see Fig. 4 (c)).

The (2) *Construction of Development Method* is shown in Fig. 5. Here, the *Method Engineer* can characterize the context of the mobile app developer (see Fig. 5 (a)). Based on this situation, he receives recommendations for *Method Patterns* and *Method Building Blocks*. During the construction of the method, he received live feedback on quality problems (see missing building block for `<< develop >>` in Fig. 5 (b)).

5 Case Study on Local Event Platform

In this section, we evaluate our approach by conducting a case study on OWL Live. OWL Live is a local event platform created in the OWL culture portal's research project³. This research project aims to establish a local area event platform that the project partners should sustainably operate. The value of the platform is to aggregate event information from different sources based on machine learning algorithms. OWL Live is a two-sided market between event

³ Project Website: <https://www.sicp.de/en/projekte/owlkultur-plattform>

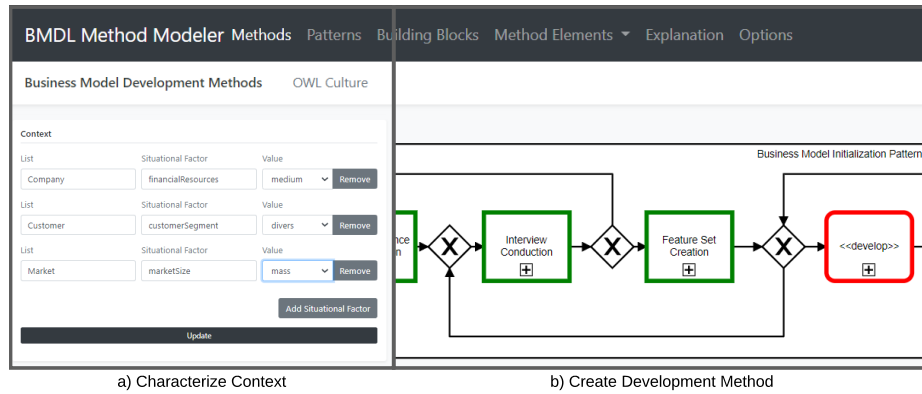


Fig. 5. Construction of the *Development Method*

providers and event visitors that both have to be considered during the business model development. Therefore, an effective business model development method that fits the project partners' situation is essential for the platform.

To create a business model development method, we interview the responsible project manager to gather the context and additional project information. Next, we constructed a development method together with the project manager by using the BMDL Method Modeler. While the situational factors (e.g., *customerSegment: uniform*, *marketSizeMass: mass*, *customerValidity: high*) could be easily used in the tool, the additional information (e.g., different customer groups due to platform business, the promised conduction of a closed and open beta phase) are harder to model by using the different patterns and building blocks. After the construction, we discussed the results with the project manager. Here, the major outcome was that the approach supports structuring the business model development and allows the discovery of new essential method parts (e.g., analysis of store trends, inbound marketing during the development).

An overview of the development method, which is fully displayed inside the tool, can be seen in Fig. 6. As the project has already defined and validated a first target group, they can use the *Target Audience Identification* without a deeper analysis. To get into the problem, we suggest running a *Market Problem Observation* to find additional pain points, and a *Store Trend Discovery*, to identify existing trends in similar apps. Moreover, we suggest analyzing the problem by a *Customer Interview Conduction* for event providers and *Social Media Survey Conduction* for event visitors. After that, a *Market Potential Analysis* should be the foundation for sustainable operation, and the *Store Competitor Analysis* and *Competitor Analysis* should allow getting an overview of other platforms. After that, the *Feature Set*, the *Value Proposition Development* and the *Business Model Development* should be conducted. Because of the different beta phases, it is important to do a *Competitive Advantage Analysis* and a subsequent *Feature Priorisation*. After that, the closed beta of the developed MVP with fewer users should be validated with group interviews in the first *Intermediate Validation*

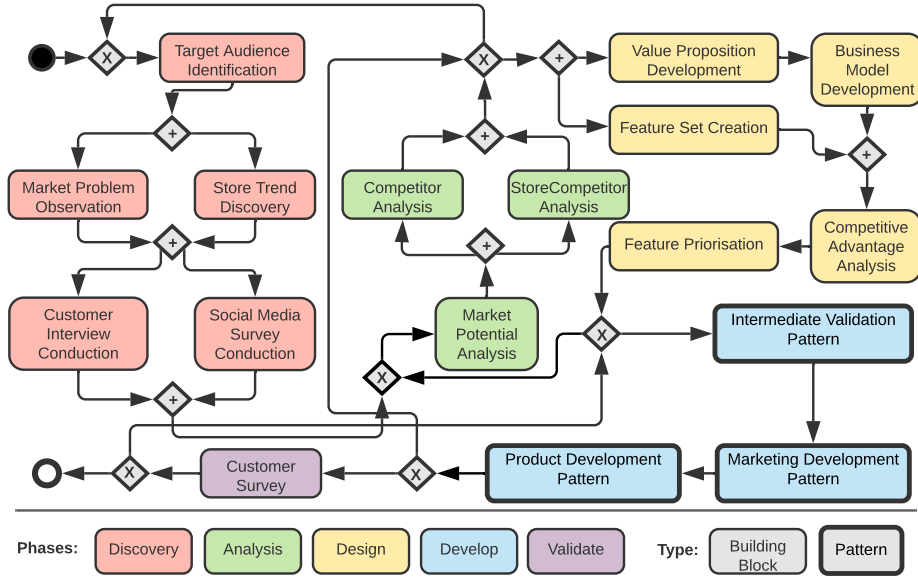


Fig. 6. Business Model Development Method of OWL Live

Pattern. After that, the open beta will be validated with customer surveys in the second *Marketing Development Pattern*. Moreover, during this open beta, the conduction of inbound marketing for getting new users is recommended. After that, the product will be developed with the *Product Development Pattern* and ongoing validated with a *Customer Survey*.

6 Conclusion

The development of effective business models is an essential task in mobile ecosystems. Here, existing development methods for these business models do not bring into focus that the development process profoundly depends on the mobile app developer’s situation. To solve this problem, we have introduced an approach for creating situation-specific business model development methods. For that, our approach consists of the two stages of creating a method base and the construction of a development method. We have implemented our approach as an open-source tool and evaluated it by conducting a case study to create a business model development method for a local event platform.

Our future work is threefold and deals with enhancing the method base, enacting the development method, and the light-weight structuring of method building blocks. First, by conducting our GLR, we comparatively quickly reach saturation of new knowledge. Therefore, we want to extend our approach so that multiple domain experts can enhance the method base with their domain knowledge (e.g., gaming apps) and improve the GLR with scientific literature. Second,

by analyzing the business model development domain, we saw that the used methods could change over time. Therefore, we want to support the enactment of the development method with an internal execution engine. This integration of construction and enaction of the method should allow a runtime adaptation of the method due to changing situations. Third, by analyzing our pattern structuring process, we investigate that it provides high guidance in structuring but also increases the modeling effort. Therefore, we want to examine a light-weight structuring of method building blocks using kanban boards. This structuring, in turn, should reduce the setup costs and complexity for smaller businesses.

References

1. App Annie Inc: The State of Mobile 2021, <https://www.appannie.com/en/go/state-of-mobile-2021/>
2. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The Influence of Situational Factors in Software Product Management: An Empirical Study. In: International Workshop on Software Product Management. pp. 41–48. IEEE (2008)
3. Bland, D.J., Osterwalder, A.: Testing business ideas. John Wiley & Sons, Hoboken (2020)
4. Boßelmann, S., Margaria, T.: Guided Business Modeling and Analysis for Business Professionals. In: Service Business Model Innovation in Healthcare and Hospital Management, pp. 195–211. Springer (2017)
5. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. *Information and Software Technology* **38**(4), 275–280 (1996)
6. Chesbrough, H.: Business model innovation: it’s not just about technology anymore. *Strategy & Leadership* **35**(6), 12–17 (2007)
7. Ebel, P., Bretschneider, U., Leimeister, J.M.: Leveraging virtual business model innovation: a framework for designing business model development tools. *Information Systems Journal* **26**(5), 519–550 (2016)
8. Eppler, M.J., Hoffmann, F., Bresciani, S.: New business models through collaborative idea generation. *International Journal of Innovation Management* **15**(06), 1323–1341 (2011)
9. Fagerholm, F., Sanchez Guinea, A., Mäenpää, H., Münch, J.: The RIGHT model for Continuous Experimentation. *J. Syst. Softw.* **123**, 292–305 (2017)
10. Frankenberger, K., Weiblen, T., Csik, M., Gassmann, O.: The 4I-framework of business model innovation: a structured view on process phases and challenges. *International Journal of Product Development* **18**(3/4), 249 (2013)
11. Fritscher, B., Pigneur, Y.: Supporting Business Model Modelling: A Compromise between Creativity and Constraints. In: Task Models and Diagrams for User Interface Design, vol. 5963, pp. 28–43. Springer (2010)
12. Garousi, V., Felderer, M., Mäntylä, M.V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology* **106**, 101–121 (2019)
13. Gassmann, O., Frankenberger, K., Csik, M.: The business model navigator: 55 models that will revolutionise your business. Pearson, Harlow (2014)
14. General Electric Inc: GE Global Innovation Barometer 2018, <https://www.ge.com/reports/innovation-barometer-2018/>

15. Giray, G., Tekinerdogan, B.: Situational Method Engineering for Constructing Internet of Things Development Methods. In: Business Modeling and Software Design, vol. 319, pp. 221–239. Springer (2018)
16. Gordijn, J., Akkermans, H.: Designing and evaluating e-business models. *IEEE Intelligent Systems* **16**(4), 11–17 (2001)
17. Gottschalk, S., Kirchhoff, J., Engels, G.: Extending Business Model Development Tools with Consolidated Expert Knowledge (accepted). In: Business Modeling and Software Design. Springer (2021)
18. Gottschalk, S., Rittmeier, F., Engels, G.: Hypothesis-driven Adaptation of Business Models based on Product Line Engineering. In: International Conference on Business Informatics (CBI). IEEE (2020)
19. Gottschalk, S., Yigitbas, E., Nowosad, A., Engels, G.: Situation-specific Business Model Development Methods for Mobile App Developers: Technical Report, <https://cs.uni-paderborn.de/fileadmin/informatik/fg/dbis/SituationSpecificBMDM.pdf>
20. Grieger, M.: Model-Driven Software Modernization: Concept-Based Engineering of Situation-Specific Methods (2016)
21. Henderson-Sellers, B., Ralyté, J., Ågerfalk, P.J., Rossi, M.: Situational Method Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
22. McGrath, R.G.: Business Models: A Discovery Driven Approach. *Long Range Planning* (43), 247–261 (2010)
23. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken (2010)
24. Osterwalder, A., Pigneur, Y.: Designing Business Models and Similar Strategic Objects: The Contribution of IS. *Journal of the Association for Information Systems* **14**(5), 237–244 (2013)
25. Qumer, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology* **50**(4), 280–295 (2008)
26. Ries, E.: The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, USA (2014)
27. Shin, J., Park, Y.: On the creation and evaluation of e-business model variants: The case of auction. *Industrial Marketing Management* **38**(3), 324–337 (2009)
28. Simmert, B., Ebel, P.A., Peters, C., Bittner, E.A.C., Leimeister, J.M.: Conquering the Challenge of Continuous Business Model Improvement. *Business & Information Systems Engineering* **61**(4), 451–468 (2019)
29. Szopinski, D., Schoormann, T., John, T., Knackstedt, R., Kundisch, D.: Software tools for business model innovation: current state and future challenges. *Electronic Markets* **60**(11), 2794 (2019)
30. Teece, D.J.: Business Models, Business Strategy and Innovation. *Long Range Planning* **43**(2-3), 172–194 (2010)
31. Terrenghi, N., Schwarz, J., Legner, C., Eisert, U.: Business Model Management: Current Practices, Required Activities and IT Support. In: Proceedings of the Conference on Wirtschaftsinformatik, pp. 972–986 (2017)
32. Wieland, M., Fill, H.G.: A Domain-Specific Modeling Method for Supporting the Generation of Business Plans. In: Modellierung 2020, pp. 45–60 (2020)