Intertwined Development of Business Model and Product Functions for Mobile Applications: A Twin Peak Feature Modeling Approach*

Sebastian Gottschalk, Florian Rittmeier, and Gregor Engels

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

Abstract. Mobile app stores like Apple's AppStore or Google's PlayStore are highly competitive markets for third-party developers wanting to develop successful applications. During the development process, many developers focus on the multitude of product functions but neglect the business model as an equally important part. As a result, developers often fail to meet customer needs, leading to unnecessary development costs and poor market penetration. This, in turn, raises the question of how we intertwine the business model and product functions during the development process to ensure a better alignment between the two.
In this paper, we show this intertwined development by adapting the concept of Twin Peaks to the business model and product functions. Based on feature modeling as an abstraction layer, we introduce the concept of a Business Model Decision Line (BMDL) to structure the business model decisions and their relation to product functions structured in a Software Product Line (SPL). The basis of our feature models is the analysis of top listed applications in the app stores of Apple and Google. To create and modify both models, we provide an incremental feature structuring and iterative feature selection process. This combination of abstraction layer and development process supports third-party developers to build successful applications both from a business and a product perspective.

Keywords: Intertwined Development · Twin Peaks · Feature Model · Business Model · Product Functions

1 Introduction

Mobile app stores are highly competitive markets for third-party developers. The analytics company AppAnnie [2] reports for 2018 that 194 billion apps are just downloaded from Apple's AppStore and Google's PlayStore which lead to revenues of $101 billion for paid apps and in-app purchases. Over 70% of this revenue is paid out to the third-party developers. With additional revenue from

* 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. (2019) Intertwined Development of Business Model and Product Functions for Mobile Applications: A Twin Peak Feature Modeling Approach. In: Hyrynsalmi S., Suoranta M., Nguyen-Duc A., Tyrväinen P., Abrahamsson P. (eds) Software Business. ICSOB 2019. Lecture Notes in Business Information Processing, vol 370. Springer, Cham

The final authentificated version is available online at:
https://doi.org/10.1007/978-3-030-33742-1_16

transactions outside the store and advertisements, the monetarization potential becomes even larger. In contrast to that, there are over 2 million apps in these stores and the average end-user uses less than 40 of them within a month. Moreover, Gartner [6] has predicted that in 2018 less than 0.01% apps would becoming financially successful, while 90% of the applications are downloaded less than 500 times (study not validated until September 2019). In order to develop a successful app, developers must consider both the business model and product functions [3]. For this intertwined development, a common abstraction layer is required, which is researched less due to the different application areas of business and product modeling.

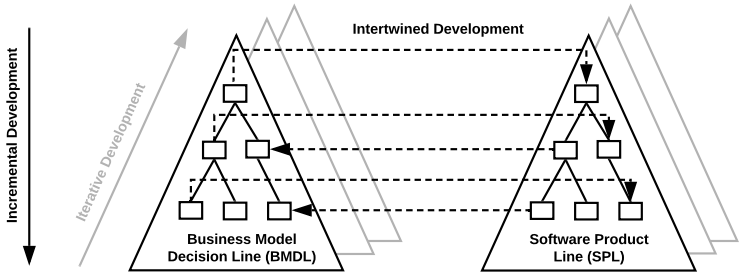

Fig. 1. Twin Peaks of BMDL-based Business Model and SPL-based Product Functions

In this paper, we show this intertwined development by adapting the concept of Tweak Peaks, which originally "intertwines software requirements and architectures to achieve incremental development and speedy delivery" [21]. Instead of the software requirements and the architectures, we intertwine the development of the business model and product functions, as seen in Fig. 1, by defining a structure and a development process. To abstract the business model and product functions due to the separation of concerns, we are using feature models as a structure. The corresponding development process is twofold: At the beginning, we create an initial structure using Incremental Development. After that, we update the structure with an Iterative Development based on customer needs.

The feature modeling of product functions can be done with the existing concept of a Software Product Line (SPL) which is a "set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" [4]. We adopt the concept of SPL to the structuring of the business model by creating a Business Model Decision Line (BMDL), where each feature represents a business model decision. The structure of the BMDL is based on the nine building blocks (Customer Segments, Value Proposition, Channels, Customer Relationships, Key Activities, Key Resources, Key Partners, Revenue Streams, Cost Structure) of the widely-adopted Business Model Canvas [22] and is refined with a domain engineering of literature and top listed mobile applications. The domain engineering provides an initial set of

features which can be extended by the third-party developer for his applications. We show the validity of our approach by providing concrete instances of our BMDL and SPL together with the development process based on a case study of streaming applications.

In the following, Section 2 describes our research approach to derive the BMDL and SPL. Section 3 shows both feature models by focussing on the BMDL as a new concept. The validity of both feature models is shown on concrete examples of streaming applications in Section 4. After that, in Section 5, we introduce the intertwined development based on Twin Peaks. Section 6 considers the related work. Finally, we give a conclusion in Section 7.

2 Research Approach

In the paper, we show the development of the business model and product functions based on feature models as an abstraction layer. For the feature models, we need to perform a domain engineering to collect the main features of mobile applications. This initial comprehensive set of features can be extended by the third-party developer to customize the feature models for his applications.

For domain engineering, we are using a 3-step extraction method based on a taxonomy development method by Nickerson et al. [20]. The method of Nickerson can be used to classify objects based on their common characteristics. We model each business model decision and product function as a characteristic of a mobile application. To use the method, we need to define meta-characteristics and ending conditions together with empirical-to-conceptual and conceptional-to-empirical iteration steps. The meta-characteristics are the most comprehensive characteristics that can be used as the basis for the choices in the taxonomy. Based on this meta-characteristics, we are running combinations of empirical-to-conceptional and conceptual-to-empirical iterations. After each iteration, the taxonomy is checked against objective and subjective ending conditions. While this section just briefly introduces the research approach, the intermediate results can be looked up in our technical report [9].

The creation process of the feature models consists of the initialization of the process, followed by three execution steps and ends with deriving of the feature models and the creation of the dependencies between them.

At the beginning of the process, we need to define the overall meta-characteristics together with the ending conditions. To model the business model decisions we are using the nine building blocks of the Business Model Canvas [22] as the most-comprehensive characteristics. We refine these blocks by the categories of the book Business Model Generation [22] to support the information extraction process. The objective ending conditions are the examination of all selected applications and papers for the corresponding execution step. As subjective conditions, we want to create an appropriate and cross-application usable model that can be easily extended by the third-party developer.

1. **Study Existing Material:** In the first step, we get an overview of different types of apps and their business models. Within the conceptual-to-empirical

iteration, we analyze selected literature [5, 7, 10, 15, 18, 19, 23, 26] from a literature search by Jazayeri et al. [13]. In the empirical-to-conceptual iteration, we look at the information of 150 apps[1] from the top lists of mobile ecosystems. Based on our updated taxonomy and different app categories, we select a comprehensive subset of the 150 applications to conduct a deeper analysis.
2. **Analyse Existing Applications:** In the second step, we conduct a deeper analysis of the product functions of the selected apps and their business model. In the conceptual-to-empirical iteration, we analyze business model decisions and product functions based on literature (e.g. analyses, news articles), which we obtain using Google Search. Within the empirical-to-conceptual iteration, we execute the apps and analyze their business model.
3. **Abstract Existing Features:** In the third step, we abstract the business model decisions and product functions to create a domain model for our taxonomy. This abstraction is especially relevant for the value propositions, which depend highly on the respective product functions. Moreover, we refine the naming and granularity of the features.

At the end of the process, we derive the feature models of the business model decisions and the product functions. Based on that, we create dependencies between these models. The result of the process is the BMDL and the corresponding SPL for the domain of mobile applications.

3 Business Model and Product Functions

In this section, we present the Business Model Decision Line (BMDL) together with the Software Product Line (SPL). While the construction and feature analysis for SPLs is well-studied in the literature [27], we focus on the BMDL. Based on the concept of Domain Engineering [27], we create a generic feature model for the construction of different business models. The model is based on an extractive product line approach, which is flexible enough to add new business model decisions in a reactive way [14].

For both feature models, we are using basic methods of hierarchical feature modeling (see Fig. 4 for a legend). Features can be mandatory or optional for the model instances. Moreover, there can be Or (at least one sub-feature is selected) and Alternate (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.

3.1 Business Model Decision Line

In this section, we present the Business Model Decision Line as the result of our analysis. In the beginning, we present the business model decisions by using the Business Model Canvas. After describing the translation from the canvas representation to a feature model, we describe important dependencies inside the feature model.

[1] Top 25 in Free, Paid and Grossing for Apple's App Store and Google Play Store

Canvas Representation The canvas representation of the business model decisions can be seen in Fig. 2. As a structure, we are using the Business Model Canvas, which consists of nine building blocks. Due to the impact to the customer needs, we are focusing on the Value Propositions, Customer Segments, Customer Relationships, Channels and Revenue Streams in this paper. Nevertheless, the Key Partners, Key Activities, Key Resources and Cost Structures are described in our technical report [9].

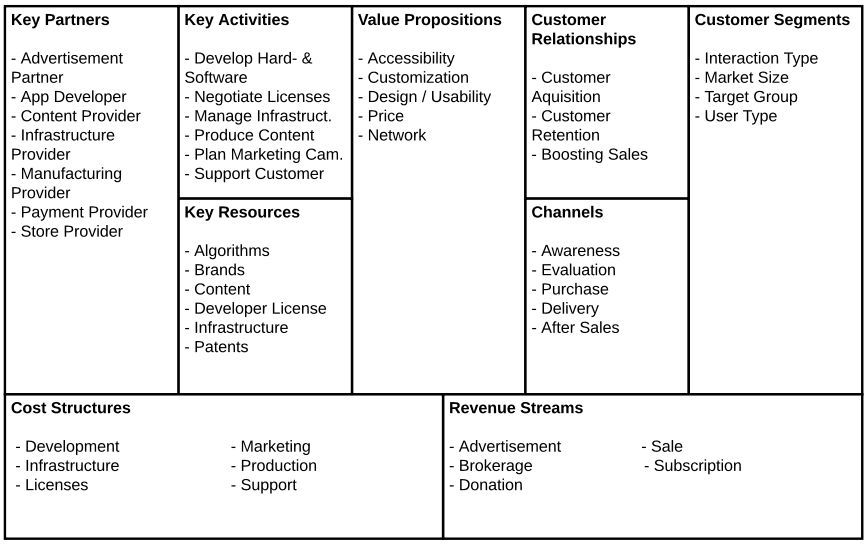

Fig. 2. Business Model Decisions for the Third-Party Developer

The **Value Propositions** are the promise of the third-party developer to a certain customer segment. Here, the *Accessibility* relates to the access strategy of the app which can be for example anonymous access, the simplified usage of single-sign-on services or the accessibility from different devices. To get a personalized experience the developer can use the concept of *Customization*. Examples of this customization are the usage of personalized recommendations or changeable user interfaces [18]. This user interfaces is also important for the *Design / Usability* decisions. To propose good usability, the developer can reduce the execution steps or use design patterns from existing applications. Part of the value proposition can also be a *Price* promise. Examples here are a low-price strategy [10] or a money-back guarantee. The last point is the *Network* aspect, which plays a role if multiple customers are connected through an application. Here, the quantity and quality of other customers can be proposed.

The **Customer Segments** are a distinct customer group which a developer wants to reach in the mobile app store. The *Interaction Type* describes the

interaction of a customer with other customers of the app. A customer can use the app only himself (called Single-User), interacting with the same type of customer (called Single-Sided-Market) or with customers of another type (called Multi-Sided-Market). Moreover, the *Market-Size* of a customer segment can be classified as a Niche- or a Mass-Market. Another point in the customer segment is the *Target Group*. The target group can be described by different characteristics like gender (e.g. Male), interests (e.g. Gamer) or relationship (e.g. Singles). The last point, we found out, is the *User Type* which relates to the decision if the customer is a private or professional one.

The **Customer Relationships** are relationships the developer wants to establish and maintain with each customer. The first step is the establishment of a relationship called *Customer Aquisition*. Examples of this acquisition step are the usage of advertisements or the implementing of a friend invitation system. After this step, the relationship is maintained within the *Customer Retention*. For the retention features like Locked-In [7], gamification or good customer support can be provided. To increase the revenue from existing customers there can be *Sales Boosting* techniques implemented. An example is the usage of Forced-Stops in games when the customer is not willing to spend money.

Inside the **Channels** the different phases of the value creation process are described. The *Awareness* is the first step to attract attention to their own application. Examples for the attraction are distribution via Word-Of-Mouth or a good store placement [10]. After creating this attraction, the customer needs an *Evaluation* of the benefits of the application. Here, the developer can use a Freemium model [19] or improve the rating and reviews in the store. This step is followed by the *Purchase* and *Delivery* of the applications. Depending on the mobile ecosystem, the payment for and the download of the application can be provided within the ecosystem or via an external system. The last step is called *After Sales*, where the customer receives value after the purchase process. Examples here are regular application and content updates.

Within the **Revenue Streams** different types of income can be generated. The most common way of generating income is the placing of *Advertisements* inside the app for example with In-App-Ads [19]. Moreover, the developer can also provide a *Brokerage* service between different customers and receives a transaction fee. In non-commercial applications sometimes also the *Donation* for the service is possible. Another possible option is the one-time *Sale* of the app or the usage of In-App-Payments for additional functions. To generate recurring revenue the developer can also use a *Subscription* model.

Feature Representation The canvas representation can be translated directly to the feature representation as seen in Fig. 3. After the translation of the model, the mandatory features have to be chosen. From the developer perspective, only the development of the application and the publishing and, if needed, the access to infrastructure, are mandatory. From a business perspective, there should be at least sales and marketing be considered. For sales, there should be at least one Revenue Stream and, if needed, the corresponding Channel to Purchase

used. For marketing, there should be strategies for Customer Aquisition and Customer Retention chosen, which can lead to marketing costs. The rest of the mandatory features, especially the Value Propositions, depend highly on the specific application.

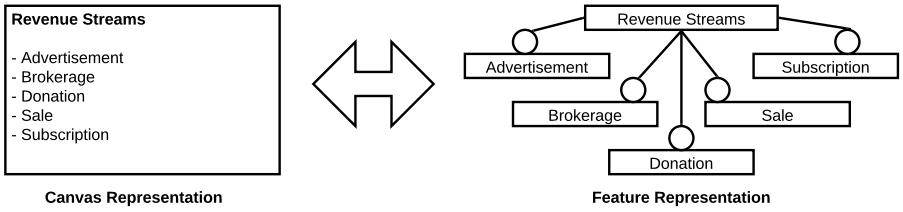

Fig. 3. Canvas Representation vs. Feature Representation

Dependency Management The structure of the BMDL can be refined by using dependencies. These dependencies can be divided into mandatory and optional dependencies.

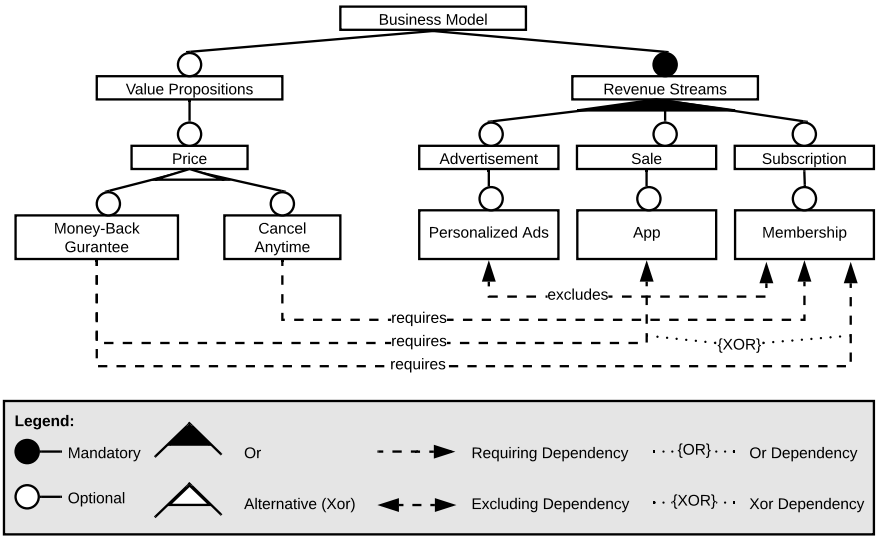

Fig. 4. Feature Dependencies of the Business Model Decision Line

The mandatory dependencies are defined mostly on the third hierarchy level of the BMDL. Here the child features of Key Activities, Key Partners and Key

Resources require specific child features in the Cost Structures. Moreover, the child features of Channels, Customer Relationships, Value Propositions, and Revenue Streams require specific Customer Segments. The optional dependencies, which are flexible choices of the developers, are defined mostly on the fourth and lower levels of the hierarchy.

An example of the dependency management can be seen in Fig. 4. Here, the usage of Personalized Ads and a Membership are excluded from each other and the Value Proposition to Cancel Anytime requires a Membership. Moreover, for a Money-Back Guarantee, there has to be used at least one payment model (i.e. Sale, Subscription).

3.2 Software Product Line

The SPL of the product functions can be seen in Fig. 5. It consists of three feature groups of General Functions (Home Screen, Settings), User (Management, Interaction) and Item (List, Consumption, Provision).

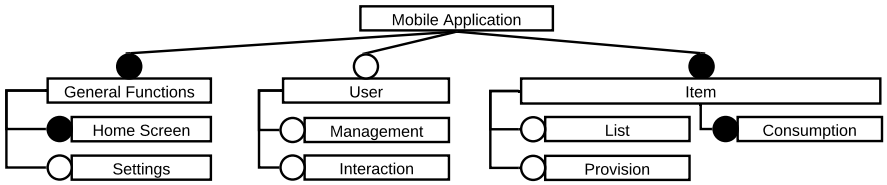

Fig. 5. Product Functions of Mobile Applications

The *General Functions* are the most common features, which are used within an application. In our analyzed application these where a home screen with some starting information and the settings for the application. If a customer can register to the application and used an account, a *User Management* needs to be implemented. In Single-Sided- and Multi-Sided-Markets there is often used some kind of *User Interaction*. Here the different users can edit their profiles, establish friendships with each other or send messages. In nearly every app there are some items (e.g. Movies, Songs, Products, Weather Information) which are displayed and processed. The *Item List* provides different parts to structure these items (e.g. Categories, Search). Within the *Item Comsumption* it is possible to interact with these items (e.g. Play, Comment, Rate). The last feature group is the *Item Provision* where content can be provided (e.g. Create Content, Upload Videos).

4 Describing existing Mobile Applications with Feature Modeling

To show the validity of our approach, we provide concrete instances of the BMDL and the SPL for the streaming applications of Netflix, YouTube, and Spotify.

For the BMDL, we focus in Table 2 on the Value Propositions (VP), Customer Segments (CS), Channels (Ch) and Revenue Streams (RS) as the most customer-related variability points. The instances of the Key Partners (KP), Key Activities (KA), Key Resources (KS) and Costs Structures (Co), which contain business-related variabilities, are described in our technical report [9]. The corresponding instances of the SPL can be seen in Table 1.

Table 1. Describing the Streaming Apps based on the SPL

Feature	Subfeature	Netflix	YouTube	Spotify
General	**Home Screen**	Home Screen		
General	**Settings**	Settings		
User	**Management**	Register, Password Lost, Login, Logout		
User	**Interaction**	-	Profiles, Friendships, Messages, Shared Playlists	Profiles, Shared Playlists
Item	**List**	Categories, Highlights, Search / Filter, Recommendations		
Item	**Consumption**	Stream, Rate, Download	Stream, Comment, Like, Download	Stream, Like, Download
Item	**Provision**	-	Upload, Update, Delete	-

5 Twin Peaks of Business Model and Product Functions

To intertwine the development of the business model and the product functions, we are using the concept of Twin Peaks [21]. In this concept, Nuseibeh discusses the general issue of the alignment of requirements and architecture within software development. Instead of considering the areas separately, both areas are developed at the same time. With this incremental development of both equally weighted areas (i.e. Twin Peaks), Nuseibeh improves the flexibility of the development process, which can adapt rapidly on changing requirements.

We adopt his concept by modeling the business model and product functions as Twin Peaks and using feature modeling as an abstraction layer (see Fig. 6). To create an initial feature model structure, we are using Incremental Development, while further changes are adopted using Iterative Development.

The *Incremental Development* provides an initial structure of the business model and product functions and consists of a Starting Step, an arbitrary number of Refinement Steps and an Ending Step.

1. **Starting Step:** In the first step, we are using the feature models of our predefined BMDL and SPL as the initial layer of our mobile application.

Table 2. Describing the Streaming Apps based on the BMDL

Block	Decision	Netflix	YouTube	Spotify
VP	**Access**	Paid Account	No Account, Free Account, Paid Account	Free Account, Paid Account
VP	**Customization**	Personalized Recommendations		
VP	**Design / Usabillity**	Responsible Design, Easy Usability		
VP	**Price**	Low-Price	Freemium	Equal-Price-Strategy
VP	**Network**	-	Quantity of other Market-Side, Share Content, Connect with other Users	Share Playlists, Connect with other Users
CS	**Interaction Type**	Single-User	Multi-Sided-Market	Single-Sided-Market
CS	**Market Size**	Mass-Market		
CS	**Target Group**	Content-Consumer	Content-Creator, Content-Consumer	Content-Consumer
CS	**User Type**	Private User		
Ch	**Awareness**	Advertisement, Word-of-Mouth, Store Position		
Ch	**Awareness**	-	3rd-Party-Integration	3rd-Party-Integration, Distributable Codes
Ch	**Evaluation**	Rating, Reviews		
Ch	**Evaluation**	Free Month	Free Month	-
Ch	**Purchase**	Homepage	Homepage, App	Homepage, App
Ch	**Delivery**	App-Store		
Ch	**After Sales**	App-Updates, Content-Updates, Push-Notifications		
CR	**Customer Aquisition**	Single-Sign-In, Invite Friends		
CR	**Customer Retention**	Content-Updates, Self-Service	Locked-In, Self-Service	Content-Quantity, Self-Service
CR	**Boosting Sales**	-		
RS	**Advertisement**	-	Advertisement without Account, Advertisement with Free Account	Advertisement with Free Account
RS	**Brokerage**	-	Money for Content Creators	-
RS	**Donation**	-		
RS	**Sale**	-	Sell Movies, Lend Movies	-
RS	**Subscription**	Subscription for Content	Subscription for Premium Content	Subscription for Premium Features

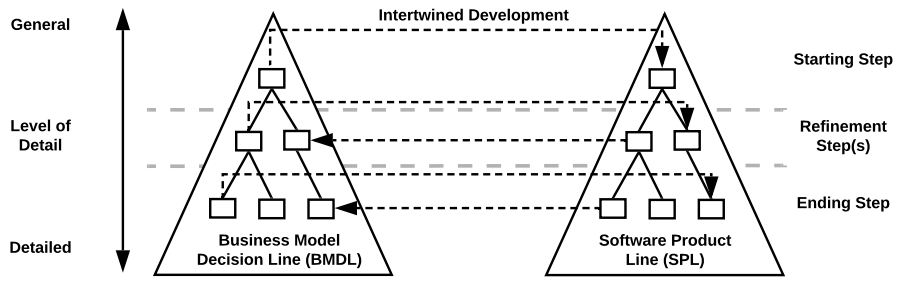

Fig. 6. Incremental Development of the Twin Peaks (based on Nuseibeh [21])

2. **Refinement Step(s):** In every refinement step, we select the features in the current layer of the mobile application and define a more detailed layer of features and dependencies within and between the business model and product functions.
3. **Ending Step:** In the last step, we select the features of the current layer of the mobile application and determine the business model and product functions.

An example of the incremental development based on streaming applications is illustrated in Fig. 7. In the *Starting Step*, we are modeling the Value Propositions (e.g. Price) and Revenue Streams (e.g. Advertisement, Subscription) as BMDL and how they are related to the product functions for the user (e.g. Management) and the item (e.g. Consumption, Provision) as SPL. For example, we can decide if we want to use an Advertisement or Subscription as an income model and notice that User Management is required for Subscription. In the *Refinement Step*, we select the Price as Value Proposition and the Subscription model and the required User Management. Moreover, we define new features for the Business Model (e.g. Cancel Anytime) and Product Functions (e.g. Upgrade) together with the creation of dependencies within and between the models (e.g. Cancel Anytime requires Membership, Adv. Features requires Upgrade). In the *Ending Step*, we choose that the user can Cancel Anytime with a corresponding Membership model. For the product functions, the user can Register, Play and Rate the existing items and Upload new items.

The *Iterative Development* provides to ability to rapidly change both models based on changing customer needs. The development can be divided into operations of Feature Selection Change and Feature Evolvement Change.

1. **Feature Selection Change:** A feature selection change is an activation and deactivation of features without changing the structure of the feature model. The change can be made directly in the model and verified with a consistency check. If consistency errors occur, the error needs to be resolved by returning to the specific layer in the incremental development and repeat the incremental development from this layer.

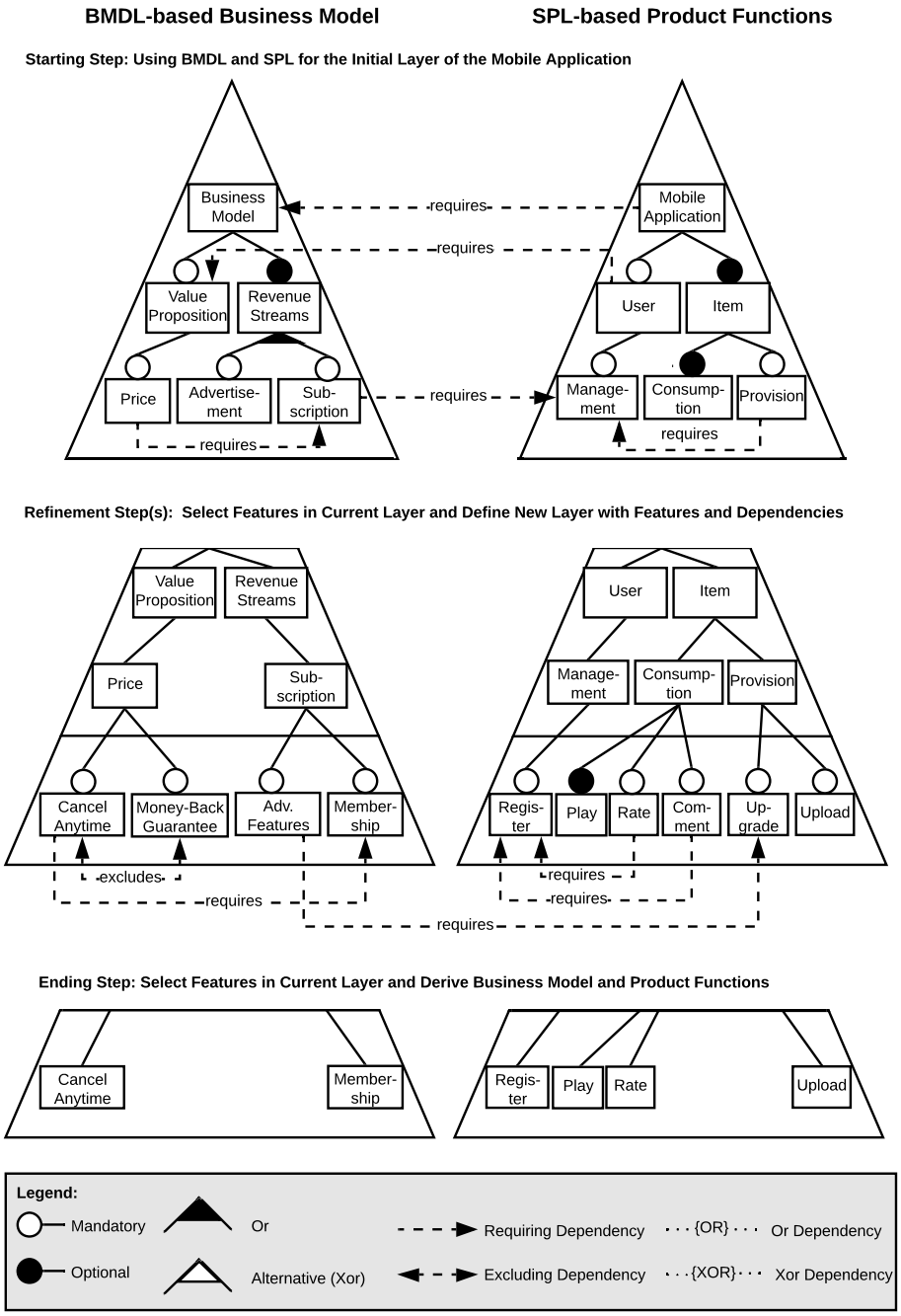

Fig. 7. Incremental Development Process of a Streaming App

2. **Feature Evolvement Change:** A feature evolvement change is adding or deleting of features in the structure of the model. The change is done by returning to the specific layer in the incremental development, add or delete the specific feature and repeat the incremental development from the layer.

An example of the iterative development can be given based on the streaming application in Fig. 7. As a *Feature Selection Change*, the developer could allow the users to comment on the application. As a *Feature Evolvement Change*, the developer could change his revenue stream from subscription to advertisement. Because the advertisement feature is not evolved in the feature model, he needs to return to the specific layer in the model (i.e. Starting Step) and starts the incremental development again based on the current structure.

6 Related Work

Integration of Business Aspects in SPL's McGregor [17] points out that changes in the business case propagated directly the architecture and components of a software product line which forces adjustments of the production and test plan. His work is based on the idea of Svahnberg et al. [24] to integrate the business unit into the requirements engineering process of an SPL. Ahmed et al. [1] perform an empirical study to figure out the most important key business factors for SPLs. Mannion and Savolainen [16] research on the aligning of business and technical strategies by arguing of feature model granularity based on the business aspects of Operational Excellence, Product Leadership and Customer Understanding.

Variability Modeling of Business Aspects Hyrynsalmi et al. [11] analyze the variability of revenue streams for third-party developers. Jansen et al. [12] propose different variation points for user-focused and developer-focused features based on app store case studies which can be interpreted as alignment between value propositions and product functions. Xu et al. [28] research on the relations of different business aspects which lead to app recommendations. Sze Wan et al. [25] analyze the value propositions of mobile messengers with a study on WeChat and WhatsApp. In [8], we introduce a Business Variability Model (BVM) to model the business model decisions of software ecosystems but not focus on the connection to the product functions.

7 Conclusion and Future Work

Mobile app stores like Apple's AppStore or Google's PlayStore are highly competitive markets for third-party developers wanting to develop successful applications. Because of the high amount of applications in these stores, the developer needs to consider the development of the business model and product functions both in app development. In this paper, we showed this intertwined development of business models and product functions using the Twin Peak concept based on feature models as an abstraction layer. The structure of the feature models

is based on the Business Model Canvas and a domain engineering of top-listed mobile applications. The development process is divided into incremental and iterative development. At the beginning of the process, we used an incremental development for the initial model, while the iterative development is used to update the model based on customer needs. This combination of abstraction layer and development process supports third-party developers to build successful applications both from a business and a product perspective.

While our current approach is made for mobile applications, it can be easily transferred to other domains. To do this the collected information in the domain engineering (i.e. Papers, Applications) needs to be exchanged with information about the new domain. This exchange will change the structure of BMDL and SPL, while the development process remains the same.

Our future work is twofold: First, we want to evaluate the structure and development process of our approach by conducting an empirical study with third-party developers. Second, we want to apply feature model mining to our approach so that the BMDL and SPL can be automatically derived from examples, which simplifies the domain engineering process.

References

1. Ahmed, F., Capretz, L.F.: Managing the business of software product line: An empirical investigation of key business factors. Information and Software Technology **49**(2), 194–208 (2007)
2. App Annie Inc: The State of Mobile 2019, https://www.appannie.com/en/go/state-of-mobile-2019/
3. Chesbrough, H.: Business model innovation: it's not just about technology anymore. Strategy & Leadership **35**(6), 12–17 (2007)
4. Clements, P., Northrop, L.: Software product lines: Practices and patterns. Addison-Wesley, Boston, 7th edn. (2009)
5. Fontao, A.d.L., Santos, R.P.d., Dias-Neto, A.C.: Mobile Software Ecosystem (MSECO): A Systematic Mapping Study. In: Annual Computer Software and Applications Conference (COMSAC). pp. 653–658. IEEE (2015)
6. Gartner Inc: Predicts 2014: Mobile and Wireless, https://www.gartner.com/en/documents/2620815
7. Gonçalves, V., Walravens, N., Ballon, P.: "How about an App Store?" Enablers and Constraints in Platform Strategies for Mobile Network Operators. In: Ninth International Conference on Mobile Business and Ninth Global Mobility Roundtable (ICMB-GMR). pp. 66–73. IEEE (2010)
8. Gottschalk, S., Rittmeier, F., Engels, G.: Business Models of Store-Oriented Software Ecosystems: A Variability Modeling Approach. In: Ninth International Symposium on Business Modeling and Software Design (BMSD), vol. 356, pp. 153–169. Springer (2019)
9. Gottschalk, S., Rittmeier, F., Engels, G.: Intertwined Development of Business Model and Product Functions for Mobile Applications: A Twin Peak Feature Modeling Approach: Technical Report (2019), https://cs.uni-paderborn.de/fileadmin/informatik/fg/dbis/IntertwiningBMandPF.pdf
10. Holzer, A., Ondrus, J.: Mobile application market: A developer's perspective. Telematics and Informatics **28**(1), 22–31 (2011)

11. Hyrynsalmi, S., Suominen, A., Mäkilä, T., Järvi, A., Knuutila, T.: Revenue Models of Application Developers in Android Market Ecosystem. In: International Conference on Software Business (ICSOB), vol. 114, pp. 209–222. Springer (2012)
12. Jansen, S., Bloemendal, E.: Defining App Stores: The Role of Curated Marketplaces in Software Ecosystems. In: International Conference of Software Business (ICSOB), vol. 150, pp. 195–206. Springer (2013)
13. Jazayeri, B., Platenius, M.C., Engels, G., Kundisch, D.: Features of IT Service Markets: A Systematic Literature Review. In: International Conference on Service-Oriented Computing (ICSOC), vol. 9936, pp. 301–316. Springer (2016)
14. Krueger, C.: Easing the transition to software mass customization. In: Proceedings of the 4th International Workshop on Software Product-Family Engineering, pp. 282–293. Springer (2001)
15. Lee, S.M., Kim, N.R., Hong, S.G.: Key success factors for mobile app platform activation. Service Business **11**(1), 207–227 (2017)
16. Mannion, M., Savolainen, J.: Aligning product line business and technical strategies. In: 17th International Software Product Line Conference (SPLC). vol. 6287, pp. 406–419. ACM (2013)
17. McGregor, J.D.: The Evolution of Product Line Assets: TECHNICAL REPORT CMU/SEI-2003-TR-005 (2003)
18. Menychtas, A., Vogel, J., Giessmann, A., Gatzioura, A., Garcia Gomez, S., Moulos, V., Junker, F., Müller, M., Kyriazis, D., Stanoevska-Slabeva, K., Varvarigou, T.: 4CaaSt marketplace: An advanced business environment for trading cloud services. Future Generation Computer Systems **41**, 104–120 (2014)
19. Müller, R.M., Kijl, B., Martens, J.K.J.: A Comparison of Inter-Organizational Business Models of Mobile App Stores: There is more than Open vs. Closed. Journal of theoretical and applied electronic commerce research **6**(2), 13–14 (2011)
20. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. European Journal of Information Systems **22**(3), 336–359 (2013)
21. Nuseibeh, B.: Weaving together requirements and architectures. Computer **34**(3), 115–119 (2001)
22. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken (2010)
23. Roma, P., Ragaglia, D.: Revenue models, in-app purchase, and the app performance: Evidence from Apple's App Store and Google Play. Electronic Commerce Research and Applications **17**, 173–190 (2016)
24. Svahnberg, M., Bosch, J.: Evolution in software product lines: Two cases. Journal of Software Maintenance: Research and Practice (11), 391–422 (1999)
25. Sze Wan, W., Dartane, O., Mohd Satar, N.S., Ma'arif, M.Y.: What WeChat Can Learn From WhatsApp? Customer Value Proposition Development for Mobile Social Networking (MSN) Apps: A Case Study Approach. Journal of Theoretical and Applied Information Technology (97) (2019)
26. Tuunainen, V.K., Tuunanen, T., Piispanen, J.: Mobile Service Platforms: Comparing Nokia OVI and Apple App Store with the IISIn Model. In: International Conference on Mobile Business (ICMB). pp. 74–83. IEEE (2011)
27. van der Linden, F., Schmid, K., Rommes, E.: Software Product Lines in Action. Springer, Berlin, Heidelberg (2007)
28. Xu, C., Peak, D., Prybutok, V.: A customer value, satisfaction, and loyalty perspective of mobile application recommendations. Decision Support Systems **79**, 171–183 (2015)