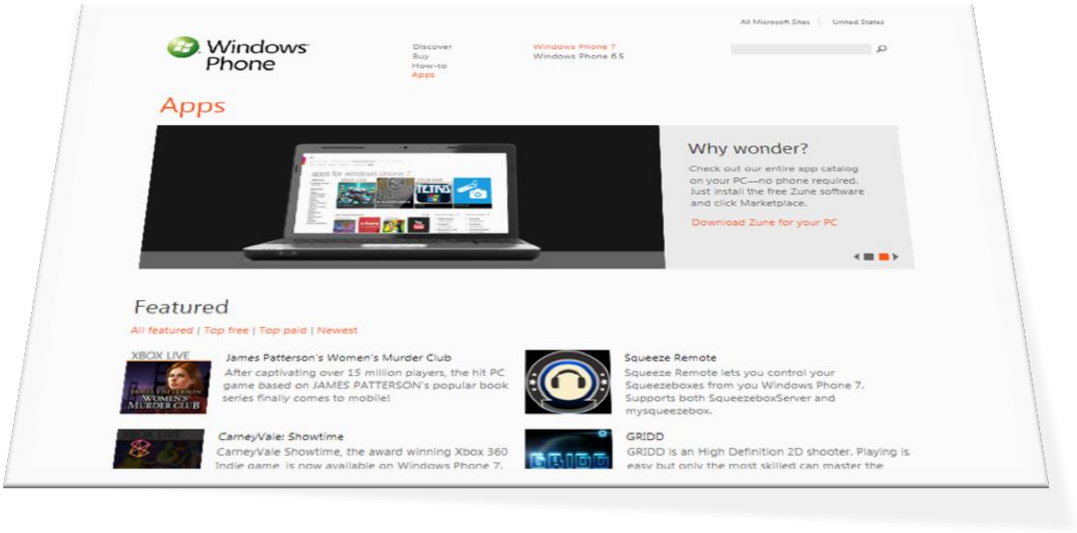
App stores for owners:
a multiple-case study of app stores



Why wonder?

Check out our entire app catalog on your PC—no phone required. Just install the free Zune software and click Marketplace.

Download Zune for your PC

James Patterson's Women's Murder Club

After captivating over 15 million players, the hit PC game based on JAMES PATTERSON's popular book series finally comes to mobile!

Squeeze Remote

Squeeze Remote lets you control your Squeezeboxes from you Windows Phone 7. Supports both SqueezeboxServer and mysqueezebox.

CarneyVale: Showtime

CarneyVale Showtime, the award winning Xbox 360 Indie game, is now available on Windows Phone 7.

GRIDD

GRIDD is an High Definition 2D shooter. Playing is easy but only the most skilled can master the

Ewoud Bloemendal – webloem@gmail.com
August 28th , 2012

Utrecht University
Master Business Informatics
Department of Information and Computing Sciences

Supervisors:	Dr. Slinger Jansen	(Utrecht University)
	Dr. Marco Spruit	(Utrecht University)

Abstract

This research aims to help app store owners translate their app store strategy into app store features and policies. At first a definition of app stores is given to scope the research. After that literature is studied to identify app store strategies. This resulted in an app store strategy model that models the app store strategy based on four dimensions. A multiple case study was performed on six app stores: Google Play, SlideMe, Apple Appstore, Binpress, Amazon appstore for Android and Intel AppUp. This study aimed to identify the features and policies used by app stores. The six sequential case studies resulted in an app store classification model with 67 features and 24 policies. The classification model is categorized based on the main influenced app store characteristic. In the last part of this research a mapping is made between the dimensions of the app store strategic model and the features and policies.

Preface

This master thesis document marks the end of my time studying for the Master Business Informatics at Utrecht University. It is the result of many hours of hard work spanning over eighteen months' time. During these months I have found that its true what they say: it is not about the destination, it is about the journey. And what a journey it was! Since the start of my graduation research I lost twenty-five pounds of bodyweight but gained five of it back later, I went on seven holiday trips to get my mind back in shape, I lost many of the convictions I held dear but found new ones along the way, I experimented with countless productivity tools until settling on a simple kitchen timer and I created over forty different versions of my thesis resulting in the one in front of you. And even though one might consider this master thesis to be a rather insignificant contribution in the search for universal truths and the meaning of life, I am sure that looking back I will keep valuing the journey that led up to it. Even if the balance of my students loans might trick you into believing otherwise.

In here I also take the opportunity to thank several people who helped shaping this master thesis. First and foremost I would like to thank my first supervisor, Dr. Slinger Jansen, for his advice, patience and motivation. After every single meeting we had I left his office with new ideas and renewed energy, again aiming for the highest result. Also I would like to thank Dr. Marco Spruit for agreeing to be the second supervisor of this project and especially for his flexibility in last minute scheduling. Great thanks goes to Frans Gemen, Henno Poesse, Erika Wortman and Martijn Knevel from IDYN B.V. who agreed to act as the project sponsor. I kindly recall the great collegial atmosphere of the many hours spend at their offices in Castle Dussen.

I would especially like to thank my dad, Henk Bloemendal, for his never ending interest in the project and support on the more difficult moments. Finally thanks to all my friends and family for their support and motivation.

Ewoud Bloemendal

Utrecht, August 28th 2012

Contents

1. Introduction

The product software business is a fast changing business in which trends come and go on regular basis. One of the latest trends that can be identified is the rise of the app stores. An app store is a curated marketplace for applications available for instant download and usually features categorization and user ratings of the applications sold. The concept of app stores is currently most successful in mobile platforms with well-known examples like the Apple App Store and the Android Marketplace by Google. App stores for other platforms include Steam for pc-games and Xbox Live Indie Games for Microsoft's' Xbox. There are also solid rumors that Microsoft is working on an app store for their new operating system version, Windows 8 (Warren, 2010).

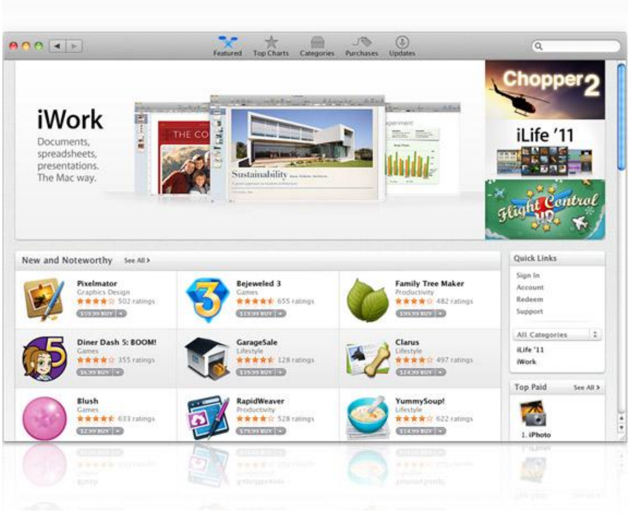


Figure 1 - The Mac App Store by Apple

The app store can be seen as a catalyst in a software ecosystem. They allow developers to sell their software and bring consumers useful functionality to their software platform. In this way external developers can not only be part of the software ecosystem but also part of the economical ecosystem that comes with it. A successful app store can possibly be beneficial to the success of a software ecosystem which consequently can be beneficial to the company that owns it. For example according to a Deutsche Bank analyst Apples App Store is the "the competitive moat which competing handset vendors cannot replicate" (Elmer-DeWitt, 2010).

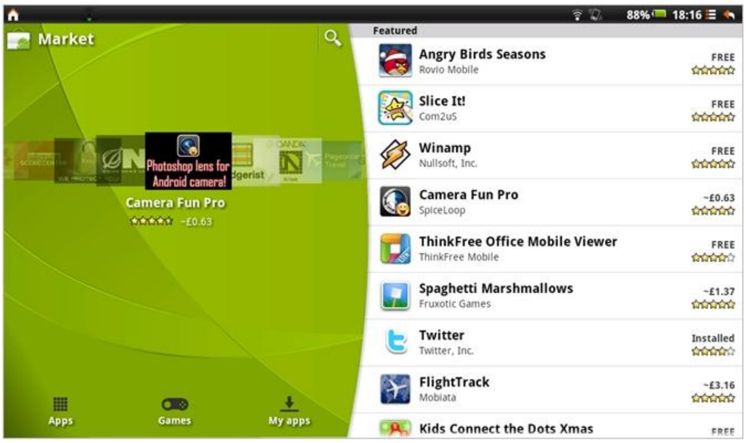


Figure 2 - The Android Market application by Google

2. Background

In order to better understand this research it can be useful to explain the origin of the concepts used. This chapter aims to establish a common ground and to relate this research to existing literature. Many of the concepts introduced here are first seen through the wide scope of scholars in economics, management or strategy and innovation. Then the scope is narrowed to the software industry to allow for a more specific and detailed description or explanation. In the following paragraphs the terms platform and ecosystems frequently occur. Several authors use different often overlapping definitions for these terms. The general rule used here is that platforms represent a perspective from products, technology or services where ecosystems represent a perspective from organizations, businesses, users or players.

2.1. App stores

2.1.1. Emergence of app stores

App stores are a very recent phenomenon, the first system using the name app store started in 2008. The concept of an app store is however not entirely new and the roots of these systems can be traced back to earlier concepts. At least three earlier phenomena that led to the emergence of app stores can be identified.

The first phenomenon were the existing content stores for mobile phones. Short after the introduction of custom ringtones services emerged selling custom ringtones to owners of mobile phones which would be charged on their regular telephone bill. When in later years mobile phones became more powerful and were able to show wallpapers and run java games the mobile content providers followed suit and started selling these too. As will be shown later in this thesis many of the currently existing (smaller) app stores for mobile devices have their roots in the sales of ringtones, wallpapers and games for mobile phones.

Another type of system that might be seen as a predecessor of app stores are extension galleries for software products. Since long software developers have used modular designs in order to increase the extensibility of their products. From the beginning of the internet many of these modules, plugins or extensions have been distributed over the internet, first ad hoc but later using dedicated websites. These systems were called extension galleries or plugin galleries. Especially in the open source community extensions galleries are a common phenomenon. Open source content management systems such as WordPress, Joomla and Drupal have large communities of extension developers who publish their work there. These galleries offer users of the extensions a curated environment where they can read reviews and ratings before trying out themselves.

A type of sites commonly used for getting software well before app stores became popular are the general download sites such as Tucows and download.com. Software programs were readily available almost everywhere on the internet but downloading software from unknown sources introduced undesirable security risks. The large download sites offered curation and reviewed software that was checked for potential security threads thus enabling users to acquire software more safely. For developers this offered a platform to market and promote their software to large groups of users.

Another example of one of the earlier digital distribution platforms is Steam, a platform for games. Founded in 2003 by Valve as a better way to distribute patches for their popular Counterstrike franchise it soon allowed users to buy games online without the need for any physical distribution. From 2005 third party developers were introduced on the store as well, bypassing the traditional brick and mortar game stores altogether. Systems like Steam and it's EA counterpart Origin are quite similar to app stores however they tend to have very high entry barriers for developers.

All the predecessors taken into account combined with the general rise of familiarity with online shopping and digital distribution allowed Apple to introduce the first system that used the name app store in 2008. Based in their already popular iTunes digital music distribution system they introduced a system that allowed users of the iPhone to download and buy apps that expanded the functionality of their phones. For developers it offered a complete new ecosystem with a relatively low entry barrier to start creating and selling iPhone apps.

It is interesting to see that in the mobile market the original focus on software is changing back to its roots of content providers. The apple app store already integrates with iTunes selling music, movies and books. Google recently rebranded their Android Market to Google Play now also offering music, TV, books, magazines and other content. In a nice touch this closes the circle all the way back to the original ringtone providers. Another recent phenomenon that has the potential of changing the app store market is the gaining support for HTML5 applications. HTML5 support is available on more and more devices and has the potential to bring the several mobile software ecosystems together with the existing web app ecosystem. How the existing app stores would cope with the potential convergence of software ecosystems is an interesting question that can only be answered in the future.

2.1.2. Naming

The term “app store” is derived from the Apple App Store for iOS devices, the most well-known app store to date. Apple is currently actively trying to establish the term “app store” as an Apple Corp. trademark by for example suing Amazon (Apple Inc. v. Amazon.com Inc., 2011) for using the name Amazon Appstore. Apple’s trademark is however disputed by other parties, most notably Microsoft (William O. Ferron, 2011) on grounds of it being too generic. They argue that the word app is an abbreviation of the very common word application and thus that an app store is much to generic to be considered a trademark.

Other possible names include variations on the word “market” (Android Market, Google App Marketplace, Adobe AIR Market), variation on “store” (Opencart Extension store, LG Application Store) or “exchange” (Salesforce App Exchange) or even only “apps” (Samsung apps). In literature other names covering slightly similar concepts include “app store-like platform business” (Kim, Kim, & Lee, 2010), “online distributed applications” or “digital distribution platform”.

2.1.3. Definition

A proper definition for an app store could not be found in current literature which is why a new definition was created. So in this research the following definition is used:

> *An app store is an online curated marketplace serving on or more software ecosystems that allows software developers to sell and distribute their products to users of a software platform.*

The different elements of this definition will be broken down and explained further below.

The first element of the definition is “online curated marketplace”. A marketplace is a common word describing a location where goods and services are exchanged. The adjective “curated” introduces the concept of a curating party that organizes and selects the collection of apps in an app store, a task generally performed by the owner. The word online of this element was added to distinguish the app store from a brick and mortar stores. Software ecosystems have been introduced earlier in this document and are made part of the definition to emphasize the relation between an app store and its ecosystem. It the following part two groups of users are mentioned: software developers and users of a software platform. The existence of these two groups are typical for an app store and thus part of the definition.

Using this definition a list of requirements can be made to limit the scope of what is considered to be an app store in this research. To be considered an app store a system should:

- be available using the internet
- be curated by an owner
- allow for the selling and buying of software products
- take care of the financial transactions involved in selling the software products
- have two distinct user groups: developers and users
- be serving one or more software ecosystem
- implement a platform that takes care of the distribution of the software products

2.1.4. App stores in literature

In general the literature on app stores is very limited, possibly due to the relative short existence of the phenomenon. The most well-known app stores come from the domain of mobile phone platforms, thus it is no surprise that these app stores have got the most attention in literature. The success of Apple’s iTunes App Store has led to a fair amount of publications, for example (West & Mace, 2010) and (Eaton, Elaluf-Calderwood, Sørensen, & Yoo, 2011) and (Ghazawneh & Henfridsson, 2010) but most of it is related to application development for the iOS platform which is outside the scope of this research.

In their 2011 paper on the mobile application market Holzer and Ondrus take a developer perspective on the significant structural changes in their market (Holzer & Ondrus, 2011). They introduce eight propositions on how the mobile software market changes for developers. Their first proposition is that portal centralization (their way of describing the introduction of an app store) makes access to customers easier. They also propose that portal centralization lowers distribution costs but also limits the freedom of developers. With regard to mobile ecosystem based on open technologies they propose that open technology will lower the development costs of applications and offer more job opportunities for developers. When app stores choose to support a larger number of devices the authors propose that this higher device variety increases freedom for developers but also increases customization cost. Finally Holzer and Ondrus propose that a fully integrated platform facilitates the flow through the distribution process.

2.1.5. The role of app stores in a software ecosystem

So how do app stores relate to the literature discussed in this chapter? In a way an app store can be seen as the epitome of platform strategy and the place where all threads come together, it is a very physical realization of a platform strategy.

The app store can firstly be seen as part of the platform, a set of services offered by the platform owner. When for example an app store offers in-app transactions it will also become programmatically connected to the product created on the platform. App stores act as a very direct economic catalyst and they facilitate immediate network effects. They perform the tasks of matchmaking, building audiences and minimizing the costs of running a community (Evans & Schmalensee). App stores are an enabler for having the keystone advantage; they help creating value and more importantly sharing value (Iansiti & Levien). From an app store owner perspective the app store is a good way to monetize the ecosystem as it allows for non-traditional pricing strategies.

3. Research Design

3.1. Problem statement and research objective

Since app stores are a relatively recent phenomenon there is not much scientific literature available on this specific topic. App store developers have currently no complete picture of the current state of existing app stores. Also there is not much known on how the characteristics of app stores relate to the business strategy of the app store owner.

Therefore the objective of this research is to gain insight in the current state of app stores by performing a multiple case study on current examples of app stores and their characteristics. Secondly it studies the relation between app store features and policies on one hand and the business strategies used by app store owner on the other.

3.2. Research questions

The main research question for this thesis is the following:

> How can app store owners translate their app store strategy into app store *features and policies*?

The following research sub questions are defined to be used in this research:

RQ1: What are app store strategies and how can they be identified?

RQ2: What are the features and policies of an app store and how can they be identified within app stores?

RQ3: What is the relation between app store features and policies and platform business strategies?

3.3. Research process and methods

The research process of this master thesis research is modeled using the Process Deliverable Diagram (PDD) modeling technique as introduced by van de Weerd and Brinkkemper. (Weerd & Brinkkemper, 2008). The rounded boxes on the left side of the model represent the process activities and the arrows the workflow between them. The boxes with sharp edges on the right side represent the deliverables resulting from the processes and the arrows depict the relations between them. The striped arrows link a process to a deliverable. For sake of simplicity several details are left out of the PDD including the list explaining the deliverables.

In Figure 3 a high level overview of the research process is given. It starts with the research design resulting in a thesis work plan. The following three activities correspond with the three research questions of this master thesis. The first activity will be identifying platform business strategies, the second will be identifying app store characteristics and the third will try to relate those two concepts. Each will result in one or more chapters of the thesis. Those three activities are listed with a white box behind them meaning it are open activities that consist of sub activities. These sub activities are more explained and further detailed in the next paragraphs. The last step is the finalization of the thesis.

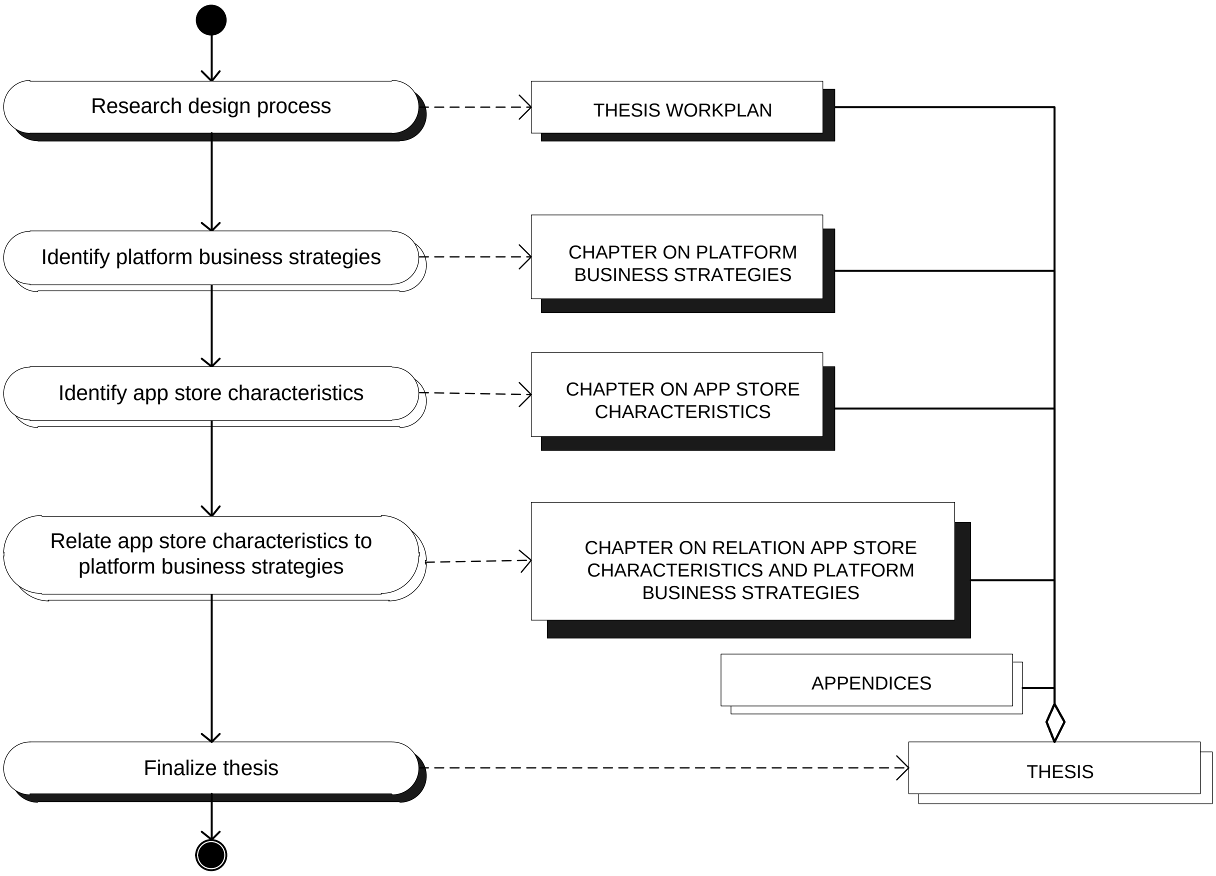

Research design process
THESIS WORKPLAN
Identify platform business strategies
CHAPTER ON PLATFORM BUSINESS STRATEGIES
Identify app store characteristics
CHAPTER ON APP STORE CHARACTERISTICS
Relate app store characteristics to platform business strategies
CHAPTER ON RELATION APP STORE CHARACTERISTICS AND PLATFORM BUSINESS STRATEGIES
APPENDICES
Finalize thesis
THESIS

Figure 3 - Overall research process modeled using a Process-Deliverable Diagram (Weerd & Brinkkemper, 2008)

3.3.1. Identifying platform business strategies

This first part of the research aims to answer the first research question (RQ1): What are platform business strategies and how can they be identified? This question does not try to find a holistic explanation of the concept of platform business strategies but rather aims to find a definition and explanation that is useful in the context of app stores and this research. Therefore existing scientific literature on this and related topics is studied to develop a suitable concept. The theory found will be used to define platform business strategies that can be related to app store characteristics. Figure 4 depicts the process using a PDD.

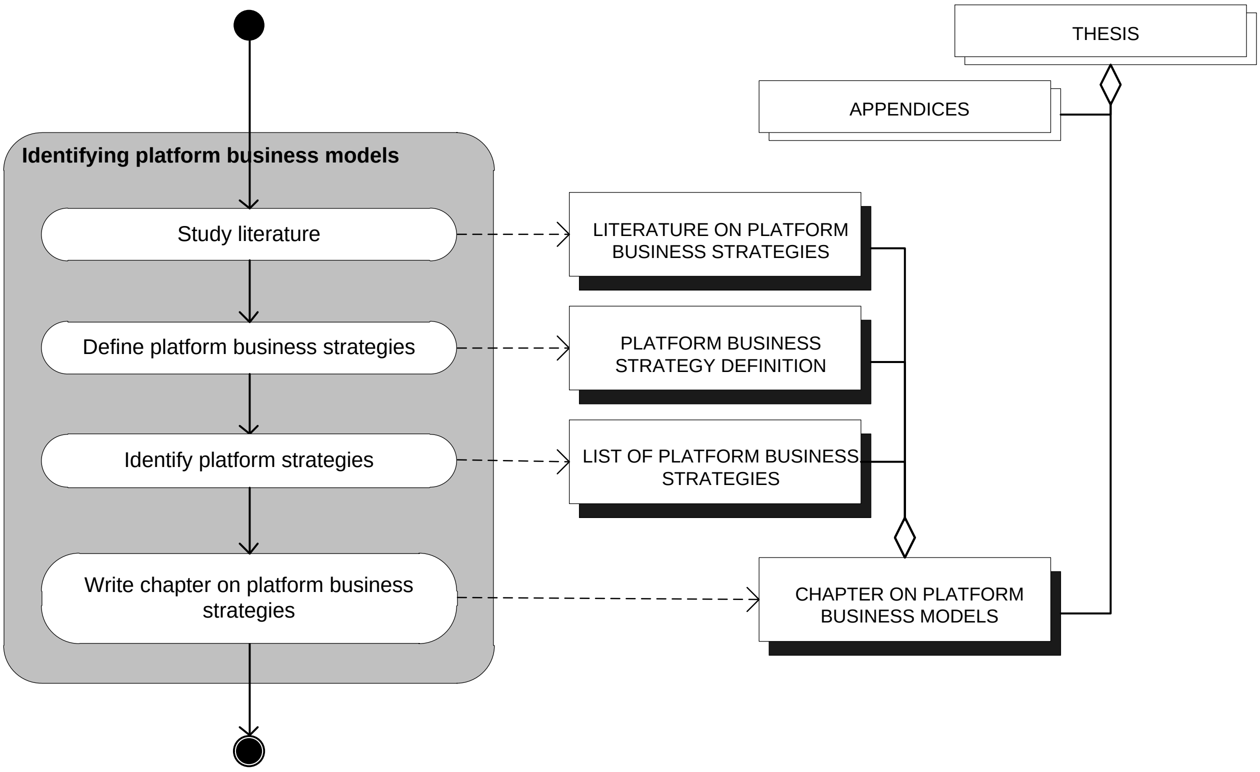

THESIS
APPENDICES
Identifying platform business models
Study literature
LITERATURE ON PLATFORM BUSINESS STRATEGIES
Define platform business strategies
PLATFORM BUSINESS STRATEGY DEFINITION
Identify platform strategies
LIST OF PLATFORM BUSINESS STRATEGIES
Write chapter on platform business strategies
CHAPTER ON PLATFORM BUSINESS MODELS

Figure 4 - Process overview of literature study

The first activity is a literature study. A more structured approach where several search keywords were systematically reviewed has been considered but due to the commonness of the keywords ("platform", "business" and "strategy" return countless papers not relevant to the subject) this did not yield the desired results. An ad hoc approach was used instead. Several papers and books recommended by the thesis supervisor were taken as a starting point. From there the search continued by exploring the relevant parts of the referenced literature. This resulted in even more literature and gave insight in important conferences concerning the topic and influential researchers in the field. Specific searches for the publications of these authors and the proceedings of these conferences added to the literature database. The found literature has been organized in groups with related topics and stored in a tree structure using mind map software XMind.

3.3.2. Identifying app store characteristics

The research question of the second part of the research is RQ2: What are app store characteristics and how can they be identified within app stores? In order to propose an answer this research question exploratory multiple-case study research will be performed on a number of existing app stores. The results of the different case studies will be combined and characteristics model will be described.

To guide the design of this part of the research the research method from a book by Robert K. Yin titled "Case Study Research – Design and methods" (Yin, 2009) is extensively used. Additionally a paper titled "Applied multi-case research in a mixed-method research project: Customer configuration updating improvement" by Jansen en Brinkkemper (Jansen & Brinkkemper, 2008) and the book "Case study methodology in business research" by Dul and Hak (Dul & Hak, 2008) are also used.

A more extensive description of the research method and some notes on how it is used can be found in paragraph 0 Figure 5 depicts the research process of this part using a PDD.

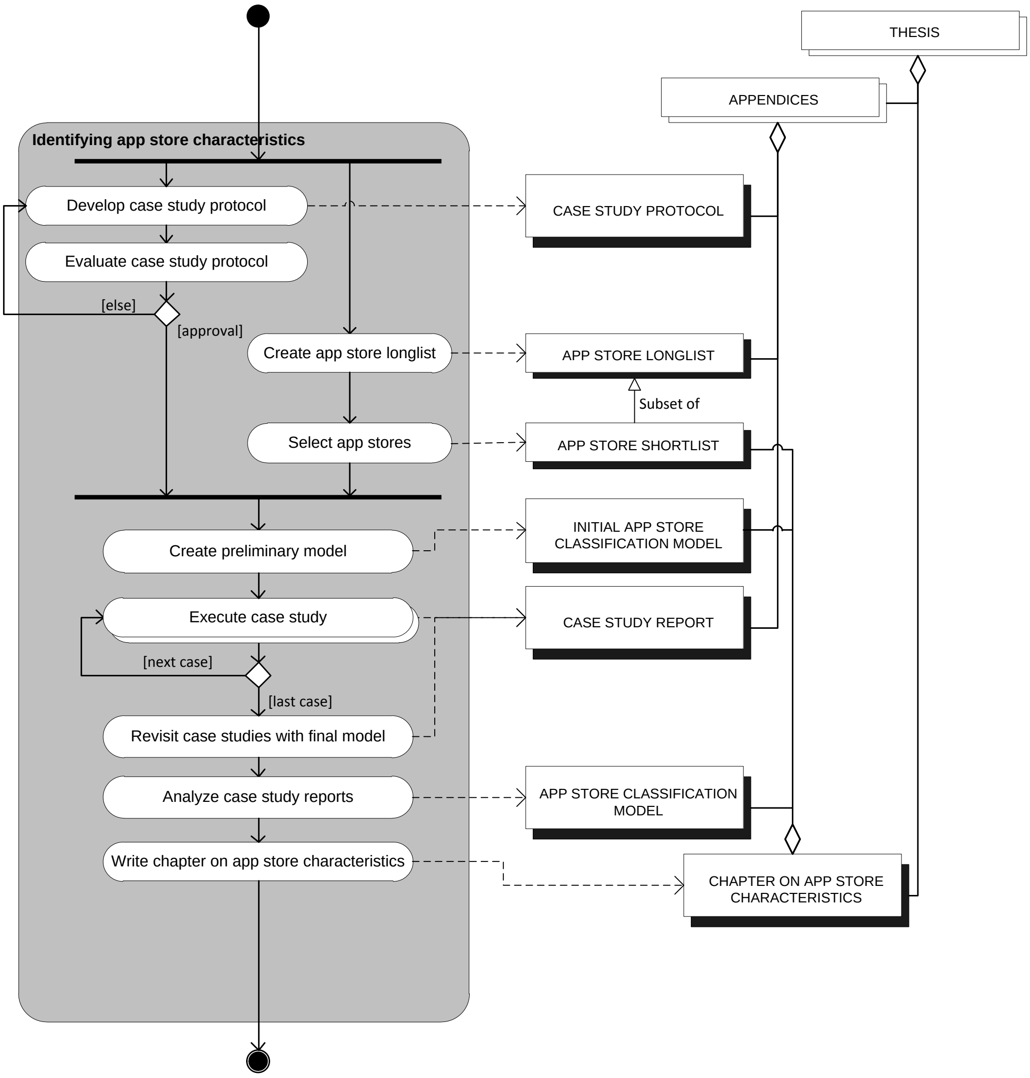

THESIS
APPENDICES
Identifying app store characteristics
Develop case study protocol
Evaluate case study protocol
[else]
[approval]
Create app store longlist
Select app stores
Create preliminary model
Execute case study
[next case]
[last case]
Revisit case studies with final model
Analyze case study reports
Write chapter on app store characteristics
CASE STUDY PROTOCOL
APP STORE LONGLIST
Subset of
APP STORE SHORTLIST
INITIAL APP STORE CLASSIFICATION MODEL
CASE STUDY REPORT
APP STORE CLASSIFICATION MODEL
CHAPTER ON APP STORE CHARACTERISTICS

Figure 5 - The case study research approach

The first activities are done parallel to each other: the development and evaluation of the case study protocol and the creation of the app store long list and the following selection of app stores from the long list. The case study protocol is suggested by Yin as a way to ensure the similar execution of multiple cases. The protocol is then evaluated together with the supervisor of this thesis research. The next step is the creation of a preliminary app store characteristics model. This model is then refined in a series of case studies, a step further detailed in Case study protocol. Each of these case studies result in a case study report. When the model is refined the earlier cases are revisited to make sure they eventually all use the same characterization. The next step is the analysis of the resulting case study reports and the formulation of the result: the final app store characterization model. The last step is creating a written report of the steps taken and the accompanying results into a master thesis chapter. The case study reports, case study protocols and app store long list are added to the master thesis as appendices.

3.3.3. Relating app store characteristics to platform business strategies

Relating app store characteristics to platform business strategies

RQ3: What is the relationship between app- store characteristics and platform business strategies?

The third activity is trying to relate the app store characteristics with platform business strategy characteristics. Since it would be hard to get any empirical evidence on this relation expert interviews are used. These experts should ideally have firsthand experience in developing and running an app store. After an interview plan is created the experts are selected and subsequently interviewed.

3.4. Strategy levels

The first thing to conceptually model are the different levels of strategy considered in this thesis. In Figure 6 the three well known strategy levels are shown: strategic, tactical and operational. For each level the corresponding concept in this thesis is modeled. The highest strategy level used is the platform strategy. The platform strategy is concerned with the question why a business would want to create a platform or ecosystem or join a platform or ecosystem. Within the context of an app store owner this strategy would also contain answers to the question why they are building or have built an app store.

Now that the platform strategy has prescribed why the company wanted to create an app store on a strategic level the app store strategy can be derived. The app store strategy is concerned with subjects on how to reach the goals outlined in the platform strategy with the use of an app store. Questions answered by the app store strategy are for example 'what revenue models would fit best with our intended developers' or 'how can we make the app store attractive to end users'. The app store strategy now can be translated into the lowest operational level, namely the features and policies. So the lowest level of abstraction used in this thesis to describe an app store will be using a set of features and a set of policies governing the use of these features.

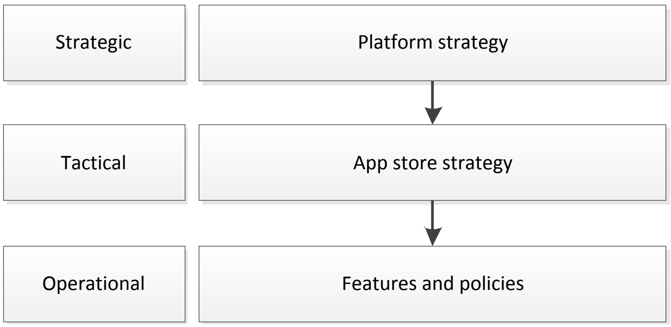


Figure 6 - Strategy levels

3.5. App store conceptual model

In order to create a better understanding of how the app store acts within its software ecosystem a conceptual model of app store mechanics was created. This model can be found in Figure 7.

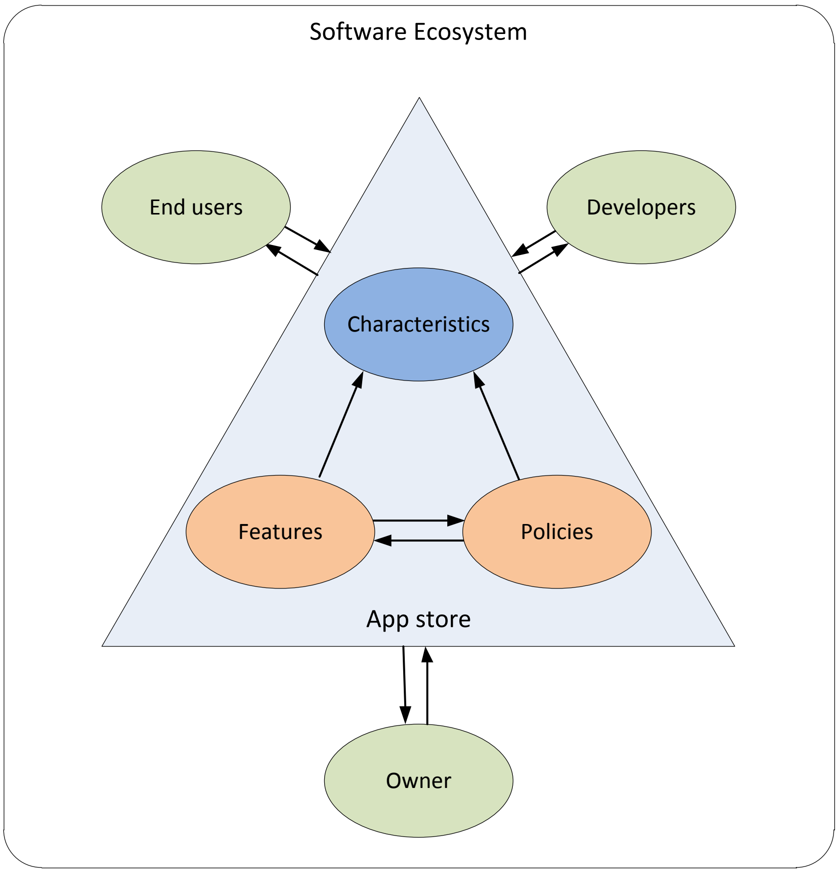


Figure 7 - Conceptual model of app store mechanics

In green the different actors within the ecosystem are modeled. The first one on the bottom is the owner, in this case not necessarily the owner of the ecosystem but rather the owner of the app store. The other two green ellipses represent a set of actors rather than one actor. The first set of actors are the end users, in this thesis also often referred to as just ‘users’. The second set of actors are the developers. The triangular shape represents the app store, with each edge facing one of the actors in the ecosystem. The app store functions as a marketplace bringing users and developers together. Developers can publish their apps using the app store, end users can search for apps and buy them from the developers using the app store. The app store is created and governed by the owner who generally takes a share of the generated revenues as compensation for this work. These relations are modeled by the arrows from and to the appstore triangle.

Within the triangle three concepts are depicted: two orange ellipses features and policies and in blue an ellipse for characteristics. Features are the most physical part of the app store. They represent the actual software and systems that the actors can interact with. Generally these consist of webservers serving a website or a set of web services. Policies represent the rules, regulations and governing processes that limit the functional reach of the features. For example an app store will most likely have a feature that allows users to write a review of an app. In most cases this will come with the accompanying policy that only users who have bought an app can write reviews about them. Features and policies together form the part of an app store that the owner can directly influence. Since he is the owner he can developer new features or change existing features and write new policies or instate new processes governing the app store.

The characteristics eclipse represents a set of app store characteristics that, in contrast with features and policies, cannot be directly influenced by the app store owner. A good example of a such characteristic would be the total number of apps available in an app store. Since the app store owner obviously cannot write all apps themselves this is something he cannot directly influence. One can image however that it is quite an important characteristic that influences the perceived value of the app store to the end users. Other examples of these characteristics are the number of developers, the number of end users, the quality of the apps, the number of paid versus free apps or the average spending of an end user. As the examples show most of these characteristics are represented using a number or statistic. Characteristics will generally be used to determine the success of an appstore: e.g. the number and quality of apps and the amount of money made using the app store.

The big assumption in this thesis is represented by the arrows going from the features and policies to the characteristics. App store owners try to indirectly influence the characteristics by implementing a certain set of features and policies. Together the features, policies and characteristics form the concept of an app store that is exposed to the end users and developers. These three combined also determine the value a certain app store represents to its users and developers. It should be noted however that this app store value is only a part of the larger overall value of the software ecosystem to a user or developer.

A more formal conceptual model can be found in Figure 8. The five rectangles each represent one of the concepts as introduced before. The first assumption A1 made in this research is that the app store strategy is result from the platform strategy. The more important assumptions are A2 and A3 which say that app store owners try to influence their app store characteristics using their feature and policy configuration. The relation that is studied in this thesis have been marked R1 and R2: how the app store strategy can be translated into a matching set of features and policies.

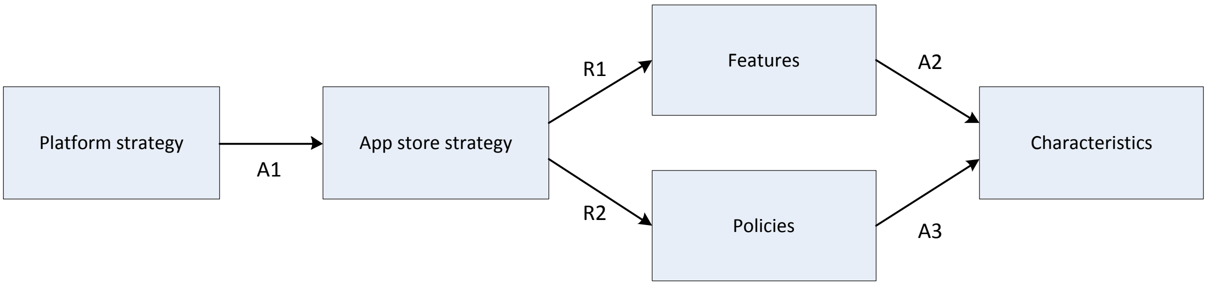


Figure 8 – Conceptual model

3.6. Case study research

Yin gives the following technical definition of a case study (Yin, 2009, p. 18): ”

> *A case study is an empirical inquiry that*
>
> - *Investigates a contemporary phenomenon in depth and within its real-life context, especially when*
> - *The boundaries between phenomenon and context are not clearly evident.*
>
> *The case study inquiry*
>
> - *Copes with the technically distinctive situations in which there will be many more variables of interest than data points, and as one result*
> - *Relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result*
> - *Benefits from the prior development of theoretical propositions to guide data collection and analysis.”*

Yin uses a “who,” “what,” ”where,” ”how,” and “why” categorization of research questions to investigate the applicability of a case study research method versus other methods such as experiment or survey method. The research question of this part of the research “What are app store characteristics and how can they be identified within app stores?” contains both the word “what” and “how”. Yin distinguishes two types of “what” questions. The first type are exploratory questions that aim to develop pertinent hypotheses and propositions for further inquiry, for example questions that start with “What can be learned from..”. The second type of “what” questions are a form of “how many” or “how much” inquiries such as “What have been the ways that..”. “How” questions (together with “why”) are more explanatory and deal with operational links needing to be traced over time instead of frequencies or incidence.

The first part of the research question used is of the first type of “what” and exploratory and the second part can be characterized as more explanatory. According to Yin using a case study as method has distinct advantages over other research methods when “a ‘how’ or ‘why’ question is being asked about a contemporary set of events over which the investigator has little or no control” (Yin, 2009, p. 13). Additionally Yin does state that many methods (including case studies) can be used to answer a “what” question (Yin, 2009, p. 10). Therefore it can be concluded that a case study research method can be used and might even be advantageous to use when trying to answer the research question. Since the emphasize of this research question is on the first part it can be characterized as an exploratory multiple case study.

3.6.1. Case study type

Yin discusses four types of case study designs based on a 2x2 matrix. On the horizontal dimension single- versus multiple-case designs are placed and on the vertical dimension embedded versus holistic analyses. A holistic analysis uses the whole case study as single unit of analysis whether the embedded analysis studies multiple units within a case. In his terms this research can be characterized as a multiple-case holistic case study design.

Dul and Hak distinguish three types of case studies (Dul & Hak, 2008). The first two are theory oriented case studies that are meant as a contribution to theory. These can be further divided as either theory-testing or theory-building case studies. The third type of case study either describes the design, implementation and evaluation of an intervention or illustrate the usefulness of a theory or approach in a specific situation. These case studies are typed as practice-oriented case studies. Using the types of Dul and Hak this case study design can be classified as a theory-building case study since it aims to contribute to the theory on app stores.

3.6.2. Multiple-case study

Both Yin and Dul and Hak are supporters of the use of replication logic with case studies. Replication logic can be seen as opposed to sample logic generally used in survey type research. The goal of both types of logic are to improve the external validity of the case study. In quantitative survey type research statistical factors such as the sample size often dictates to what extend a finding can be generalized to a larger domain. Case study research on the other hand uses qualitative analysis methods where the sample is often very small. In order to increase the confidence in the findings of a case study according to the authors a case study should be replicated. This can be seen as analogue to experiment type of research where a certain experiment is replicated multiple of times to increase the confidence in the findings. Yin advises to use replication logic where a case study should be replicated amongst several carefully selected cases that (based on theoretical propositions) are expected to yield a similar result. A more complicated study could use a “two-tail” design which consist of cases that have been chosen to yield results from both extremes of the expected outcomes. Yin calls this type of case study a multiple-case study and it is his preferred type of case study design. Dul and Hak do agree with Yin that using replication logic can greatly benefit a case study design but also note that in (business research) practise it is not often used. The reason is that it best suited for theory testing case studies whereas in practices most case studies are theory building or practise oriented.

Dul and Hak distinguish three types of multiple case study designs (Dul & Hak, 2008, p. 45): a comparative case study, a parallel single case study and serial single case study. A comparative case study is a multiple case study design that does not use replication logic whereas both parallel and serial single case studies do use replication logic. The serial type differs from the parallel type that it performs the case studies one after each other and takes the results of the previous case studies into account, as opposed to the parallel case study where all case studies are executed at the same time testing the same proposition. Yin’s proposed case study design would be a serial single case study in terms of Dul and Hak.

This overall research question used is focussed on theory building research and as Dul and Hak state this is less suited for using replication logic. This research however does also use the results of earlier cases in the following case study and thus which comes close to the replication logic of theory-testing research. In terms of Dul and Hak this would be a serial single case study. In the paragraph on case selection it can be found that with the selection of cases also some sample logic is used. As input for the first case study and a preliminary app store characteristics model is created. This model is based on an ad hoc literature review supplemented with input from a search through grey literature. An online search using Google with keywords such as “app store comparison” and “app store description” can offer a variety of app store characterizations that can be used as input for the preliminary model.

3.6.3. Data collection

The case studies performed in this research are all done using the internet without having any control over the case at all. Yin acknowledges that this can be viable way of doing research: “You could even do a valid and high-quality case study without leaving the telephone or Internet, depending upon the topic being studied.” (Yin, 2009, p. 15).

Yin introduces six most commonly used sources of evidence in a case studies: documentation, archival records, interviews, direct observations, participant-observation and physical artifacts. Three of them are relevant for this research and are further explained.

The first source of evidence is documentation. With documentation Yin means all forms of documentary information such as for example letters, e-mail correspondence, agendas, announcements, administrative documents and news clippings. These documents are often available through the internet. The most important use of documentation for case studies according to Yin is to corroborate and augment evidence from other sources. In general they are regarded as useful but they might not be accurate and lack in bias. A systematic search for relevant documentation are an important part of the data collection strategy of a case study. In the context of app stores documentation can be in the form of legal documents such as terms of service and agreements, articles from grey literature or developers documentation.

The second source of evidence is direct observation. Direct observations are possible when a case study is done in the natural setting of a case. Data collection from these observations can vary in formality from formal to casual. With a case study in a social context a researcher can for instance observe certain behaviors of people. According to Yin observational evidence can be useful in providing additional information about the topic being studied. In this research on app stores, the app store is a new technology that is directly observed within its actual context. Evidence from direct observation in this research will be gathered by visiting the actual app store over the internet or other required medium. This will in some cases cross the boundary with another source of evidence Yin mentions: participant-observation. Participant-observation is a mode of observation in which the researcher takes actively part in the case study. In the context of app stores this might be the case when for example for documenting purposes a transaction is made in an app store. This gives the opportunity to document a greater part of an app store. Yin identifies some problems with this type of evidence that might produce biased results but in the context of this research they are not relevant.

The third source of evidence are physical artifacts. According to Yin this can be a technological device, a tool or instrument, a work of art, or some other physical evidence. The relevance of artifacts is often limited however. In the context of this research physical artifacts could be screenshots of app stores provided by other sources. This type of evidence however converges with direct observations and documentation.

3.6.4. Quality

Yin lists four tests that are relevant to judge the quality of case study research. They are the following: construct validity, internal validity, external validity and reliability. Each of them will be further explained.

The construct validity tests whether the concepts being studied are correctly operationalized and measured. According to Yin this is especially challenging for case study research. In his definition of case study research he states that case study research is often used "especially when the *boundaries between phenomenon and context are not clearly evident*". This makes it hard to operationalize the concepts that are studied. Yin gives three tactics that can help to improve the construct validity of a case study. The first tactic is to use multiple sources of evidence when collecting data and use them in a convergent manner. The second tactic is to create a chain of evidence during data collection. The third tactic is to get the case study report reviewed by key informants.

Internal validity tests whether the research correctly establishes causal relationships. According to Yin this only concerns explanatory case studies and since this research is exploratory by nature it is not a relevant test for this research.

The third test is external validity, which describes the domain to which the findings of the case study can be generalized. In survey research this is often related to the sample size of the dataset. This analogy is according to Yin however not valid when dealing with case studies because they rely on analytical generalization rather than statistical generalization. Yin suggests to use replication logic instead, which replicates the test of a theory over multiple cases in a broader domain to improve the external validity. To accommodate the external validity replication logic is used in the case study design. In addition to that some sample logic is used when selecting the case studies from the long list.

The final test is reliability or empirical reliability. The objective of this quality criterion is that when the case study is done all over again by another researcher using the same procedure, it should give the same results. This foremost requires that the case and the methods used are sufficiently documented. Yin advices two specific tactics to improve the reliability of case study research: the use of a case study protocol and the development of a case study database. Both are used in this research.

3.6.5. Scope

The scope of the characterization is determined by the overall goal of this research: to help app store owners when they are creating an app store. The characterization will therefore be limited to that parts of an app store that can be directly influenced by the owner. For example the quality of the applications listed in a certain app store will be beyond the scope of this research. The quality guidelines and policies the owner however has implemented is most certainly part of the characterization.

The scope of this part of the research is secondly determined by the definition of an app store that is used in this research. This has for example limited the selected case studies. Even though the number of app stores seems to be growing rapidly it is not needed to limit the number of app stores under consideration by for example adding a geographical constraint. This would also contradict the fact that most app stores have a worldwide scope themselves.

3.6.6. Creating a case study protocol

Yin promotes the development of a case study protocol before starting case study research. In fact, Yin sees the development of a case study protocol as essential in case of a multiple-case study. The protocol is used to guide the researcher when collecting data from a case. The most important goal of the protocol is to increase the reliability of the case study. In general a case study protocol should contain the following sections: an overview of the case study project, field procedures, case study questions and a guide for the case study report.

The protocol itself is targeted at a single data point, in this research a single case. Since this is a multiple-case study the protocol is reused for each case, with only minor changes to the case study overview to reflect the specificities of the case. The reuse of the same protocol is essential to ensure that data is measured in a consistent and objective way. This improves the construct validity of the research.

The case study protocol used in this research is shown in paragraph 3.7.

3.7. Case study protocol

The main focus of this protocol is to provide a structured and repeatable way of doing the case study research. Since the cooperation of app store owners with this research is unlikely it does not aim to communicate the research design to the stakeholders.

The case description is the only part of this case study protocol that varies between the cases. It is therefore made part of the case study database rather than the case study protocol. The main research question for each case study is RQ2: What are app store features and policies and how can they be identified within app stores?

3.7.1. Field procedures

Each case study will be performed using the procedures modelled in Figure 9. The research is done in four steps: initialization, data collection, analysis and finalization.

The initialization starts with asking the case owner if they would like to participate in this research by giving access to their app store. It is expected that most app store owners will choose not to cooperate in the research or not to react at all to the request. In that case other means of access will be investigated. This can be done by obtaining a client device and using the app store from the client perspective and by creating or borrowing a developer account. If direct access to the app store is not or only partially possible either the case has to be dropped or it should be considered whether it is possible to use other sources of evidence to document the app store characteristics.

The data collection step starts with creating a description of the app store. This will at least include a short textual description of the shop, its owner organization, its software ecosystem and a link to where it can be found. The second step will be to determine the usage data of the app store. Usage data can be an important factor in determining the value of the case for the eventual characterization model that is produced. Usage data is often not made available by the owner organization of the app store, so this data will have to rely heavily of data from other sources. This can either be news coverage of press releases by the owner organization or gathered from other parties that keep statistics of app stores.

Studying app store features and gathering documents surrounding the app store are the next steps that are simultaneously executed. Direct observations of the features of an app store will be done by the researcher and documented using screenshots. Other documents will be searched systematically within the information provided by the app store owner or if needed by other relevant sources. Both the feature reports and documents found will be added to the case study database.

The next step is to analyse the data that is collected. Except for the first case study each case study will use the model provided by the previous study to create an app store characterization. If the collected data suggests improvements of the model these will be applied to the model as the following sub activity. This results in a refined app store characteristics model. The results of the analysis and a description of process will be added to the case study report.

The last activity is the finalization of the case study. The report for the case study will be written and the case study will be published online. If applicable the owner of the app store will be thanked for its cooperation and given a copy of the case study report.

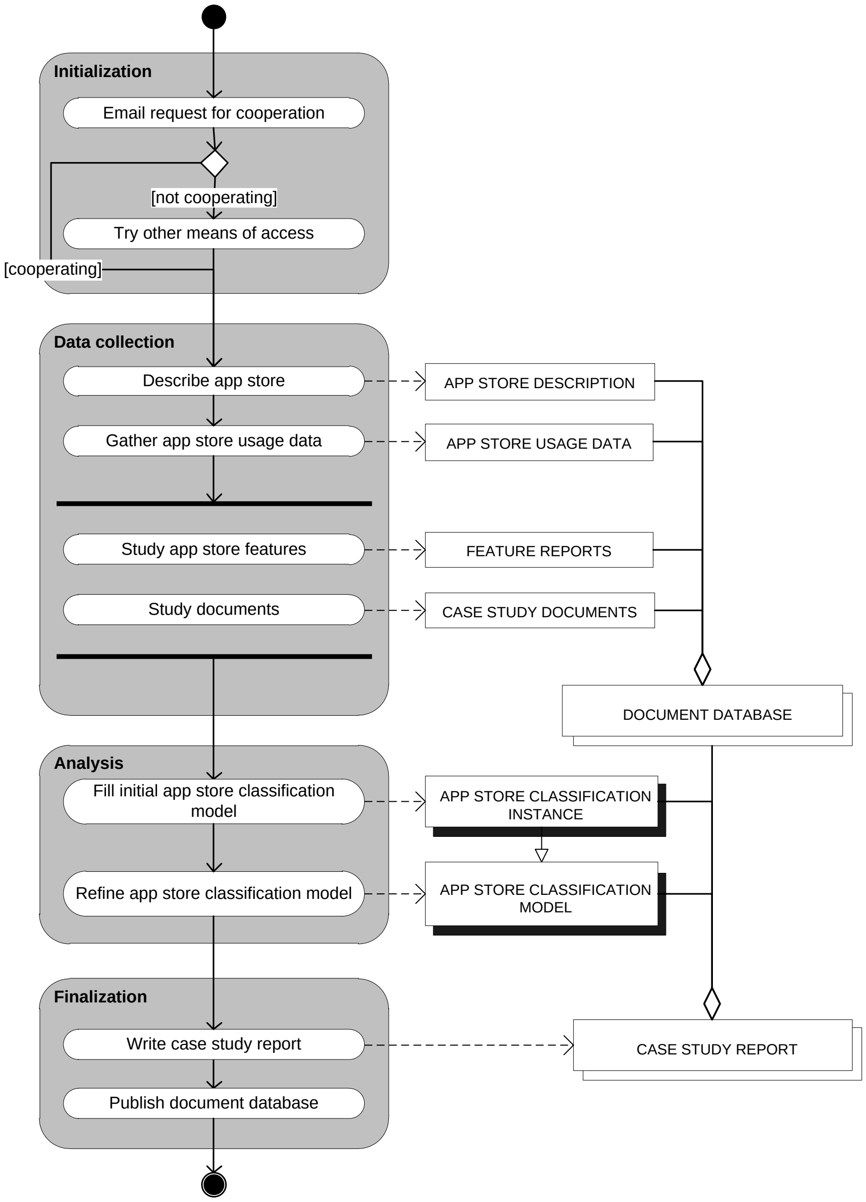


Figure 9 - Field procedures

3.7.2. Document database

The document database entirely consist of digital materials. In its basic form it is a folder containing the case description document, the app store usage data, feature reports with accompanying screenshots and other case study documents. To keep an index on the case study database an overview will be created using Microsoft Excel. Each document will be added to the index using a title, a year of publication and the source of the data. The sources can be either literature, observations, documentation from the app store owner or documentation from other sources on the internet. For sources from the internet the URL is stored in the index together with a screenshot of the page in order to provide reference to the document if it would not be available online anymore. Each document in the document database will have a unique identifier that will be used throughout the case study to refer to the document.

3.7.3. Case study report outline

The following topics has to be discussed in each case study report: a case description, usage data on the app store, means and date of access, the initial model used, the filled in model resulting from the case and changes to the model between those two versions of the model resulting in a refined model.

4. Results literature study

In this chapter the results of the literature study on app store strategies is listed. The topics discussed come mostly from three angles: literature on software ecosystems, literature on platforms and economic literature on two-sided markets. First ecosystems and platforms are discussed as they form the basis in which app stores operate. After that some reasons for creating an appstore are discussed as well as some mechanisms and governance strategies that are relevant to app stores. Finally a app store strategy model is proposed combining the most relevant factors resulting from this literature research.

4.1. Ecosystems

Companies nowadays are not stand-alone entities with a single group of clients and suppliers anymore. In management literature this shift can also be seen: the classic value chain by Porter has evolved into extended value chains and value networks. Companies and their products and services cannot be considered on their own but should be seen in context of all their connections to other companies, products and services. Especially in the software industry companies heavily rely on products other companies to create their value propositions. Microsoft's Windows for example would be a less useful without the enormous amount of applications that you can run on it. In this paragraph the literature on ecosystems is introduced and further explained in the context of this research.

4.1.1. Business ecosystems

In 1993 James Moore first introduced the ecological metaphor of a company as part of an ecosystem in an article in the Harvard Business Review (Moore, 1993). In his article he argues that successful businesses are those that are rapidly evolving, since out-innovating its competition is the only sustainable advantage a company can have. He borrows the concept of an ecosystem from biology where the evolution of a certain species is influenced by its surrounding ecosystem and then compares it to the business world: "*In a business ecosystem, companies coevolve capabilities around a new innovation: they work cooperatively and competitively to support new products, satisfy customer needs, and eventually incorporate the next round of innovations.*" (p. 76).

In 1996 Moore wrote a book on this topic in which he gives a more extended definition of a business ecosystem and which will be used throughout this master thesis:

> *"An economic community supported by a foundation of interacting organizations and individuals—the organisms of the business world. The economic community produces goods and services of value to customers, who are themselves members of the ecosystem. The member organisms also include suppliers, lead producers, competitors, and other stakeholders. Over time, they coevolve their capabilities and roles, and tend to align themselves with the directions set by one or more central companies. Those companies holding leadership roles may change over time, but the function of ecosystem leader is valued by the community because it enables members to move toward shared visions to align their investments, and to find mutually supportive roles"* (Moore, 1996, p. 26)

4.1.2. Business ecosystem roles

Iansiti and Levien elaborate on the concept of business ecosystems in their book “The keystone advantage” and distinguish three significant types of member strategies within a business ecosystem: the keystone, the physical dominator and the niche player (Iansiti & Levien, 2004, p. 68). A keystone member fundamentally aims to improve the overall health of its ecosystem by providing a stable and predictable set of common assets that other organizations use to build their own offerings. Keystones have generally a small physical size compared to their impact in the ecosystem. A physical dominator tries to integrate vertically or horizontally to own and manage a large proportion of the ecosystem directly and then it will try to exploit its position. This is an aggressive strategy that creates and captures a lot of the value of the ecosystem. The niche player aims to develop specialized capabilities that differentiate it from other companies in the ecosystem. Although being relatively small on their own they are high on numbers and so niche players account for the biggest part of the value created in the network (Iansiti & Levien, 2004). Iansiti and Levien provide numerous examples of these strategies and generally promote the keystone strategy over the physical dominator strategy. One example they provide is the success of the Wal-Mart retailing company. The main driver behind this success is that it figured out how to create, manage and evolve its vast network of business partners. They use their consumer information to coordinate and optimize their network of suppliers. Using technology, capabilities and policies they made it function as a big collective providing customers with the products they needed in time, at a low price.

4.1.3. Software Ecosystems

In the software industry the shift in perspective towards business ecosystems can also be found. When the concept of a business ecosystem is applied to the software industry the term software ecosystems arises, first coined in 2009 by Bosch (Bosch, 2009). Software vendors are not creating a product just by themselves anymore and selling it to their own customers anymore but rather function as a part of a larger ecosystem of software companies and products. The software they make is most likely based on the platform of another company, using the tools of yet another company and they might be selling it to consumers through a marketplace owned by a third company. For software vendors this means a shift in perspective towards software ecosystems and a need for strategies that are aware of these ecosystems.

Jansen, Finkelstein and Brinkkemper have published a research agenda for software ecosystems in which they give a definition of software ecosystems:

> *“[A software ecosystem is] a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts.”* (Jansen, Finkelstein, & Brinkkemper, 2009).

In the same paper Jansen et al. introduce three different perspectives or scopes on a software ecosystem, based on earlier work by Brinkkemper et al. on Software Supply Networks (Brinkkemper, van Soest, & Jansen, 2009). Their modeling of the perspectives is shown in Figure 10 with the independent software vendor (ISV) in the center. The software vendor level shows the software vendor and all products and services supplied by the vendor. The software supply network level considers all buyers and suppliers who have direct contact with the ISV. The broadest perspective level is the software ecosystem level which shows all related software organizations in the software ecosystem.

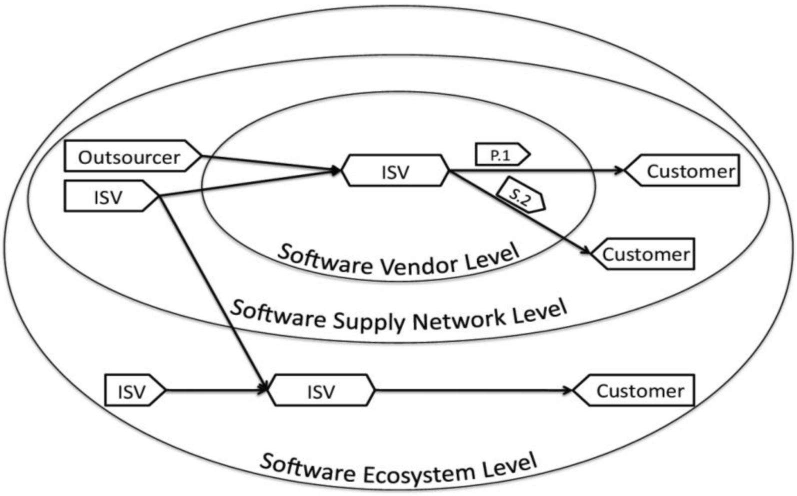
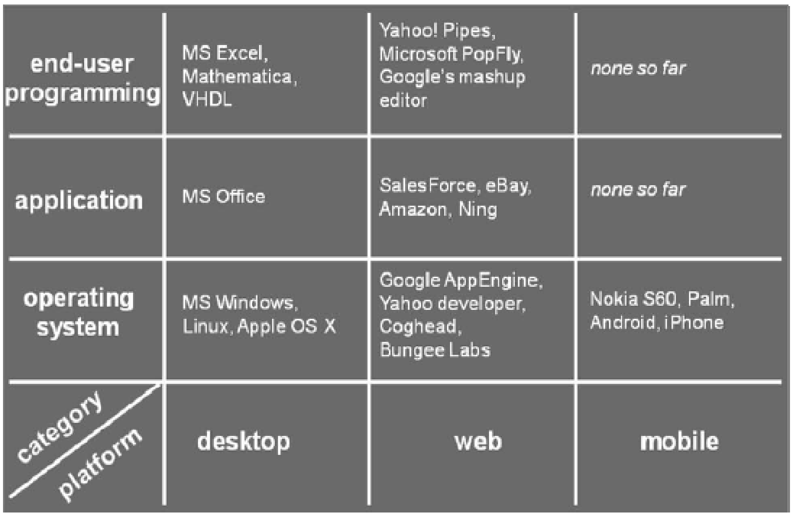


Figure 10 - Software Ecosystem Perspectives (Jansen, Finkelstein, & Brinkkemper, 2009)

Bosch has also introduced his Software Ecosystem Taxonomy. This taxonomy organizes software ecosystems in a two dimensional space according to the level of abstraction and the dominant computing platform. Figure 11 – Software Ecosystem Taxonomy by Bosch shows this taxonomy and gives some examples for each of the blocks.



end-user programming	MS Excel, Mathematica, VHDL	Yahoo! Pipes, Microsoft PopFly, Google's mashup editor	*none so far*
application	MS Office	SalesForce, eBay, Amazon, Ning	*none so far*
operating system	MS Windows, Linux, Apple OS X	Google AppEngine, Yahoo developer, Coghead, Bungee Labs	Nokia S60, Palm, Android, iPhone
category **platform**	**desktop**	**web**	**mobile**

Figure 11 – Software Ecosystem Taxonomy by Bosch (Bosch, 2009)

4.1.4. Software ecosystem modeling

Boucharas, Jansen and Brinkkemper proposed a way to model software ecosystems (Boucharas, Jansen, & Brinkkemper, 2009). They identified the need for a formal modeling standard since software vendors had trouble to distinguishing the specific software ecosystems in which they are active and how to use these ecosystems to their strategic advantage. The modeling standard they provided aimed to enable software vendors to communicate about relationships in their software supply network, theorize about weak spots or weak links in their business model and to anticipate upcoming changes in the software ecosystem.

The Software Ecosystem Modeling (SEM) technique they propose consists of two models: the Product Deployment Context (PDC) and Software Supply Network diagrams. The PDC provides a quick overview of the architecture and dependencies of a software product in its running environment (see Figure 12 for an example). The Software Supply Network diagrams are based on earlier work (Jansen, Brinkkemper, & Finkelstein, Providing transparency in the business of software: a modeling technique for software supply networks, 2007) and depict a series of linked software, hardware, and service organizations cooperating to satisfy market demands (see Figure 13 for an example). An SSN model shows the dependencies and flows between firms and thus allows reasoning about the business model of a software firm.

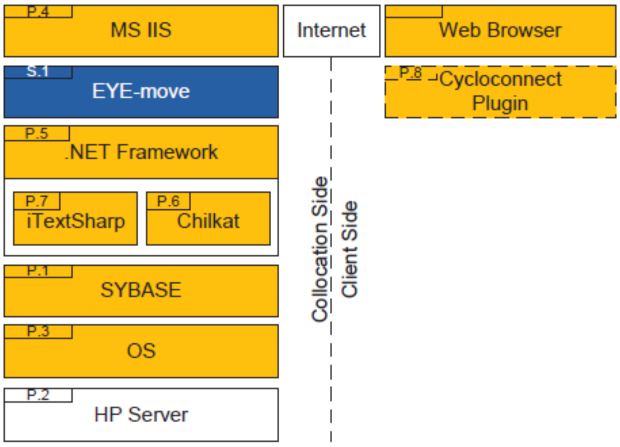


Figure 12 - Example of a Product Deployment Context (Boucharas, Jansen, & Brinkkemper, 2009)

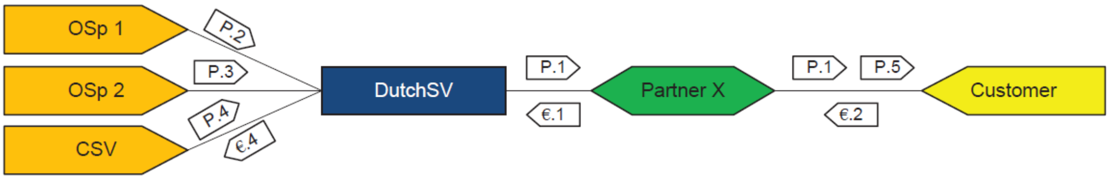


Figure 13 - Example of a Software Supply Network (Boucharas, Jansen, & Brinkkemper, 2009)

4.1.5. Ecosystem roles

As mentioned before Iansiti and Levien distinguish three significant member strategies a business ecosystem: the keystone, the physical dominator and the niche player. Software ecosystems have a smaller scope which allows for a more precise division of ecosystem roles. Messerschmitt and Szyperski provide such a partitioning with eight business functions forming a software value chain from software creation to software use, shown in Figure 14. (Messerschmitt & Szyperski, 2005).

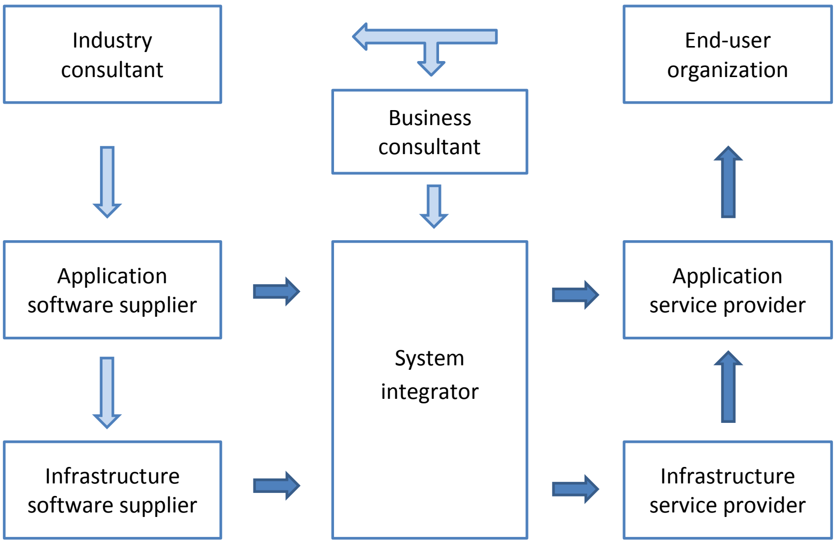

Industry consultant
Business consultant
End-user organization
Application software supplier
System integrator
Application service provider
Infrastructure software supplier
Infrastructure service provider

Figure 14 - Natural business partitioning of the value chain (Messerschmitt & Szyperski, 2005, p. 174)

Examples of the three roles of Iansiti and Levien can easily be mapped to compositions of the business functions of Messerschmitt and Szyperski as is shown in Figure 18. Both authors refer to an example from the mainframe era, such as IBM. IBM in the 1960's would produce every technological part of its mainframes itself and provide virtually every service that its customers would need. As a typical physical dominator they were having a large physical presence in the ecosystem (shown in blue). The Windows OS platform by Microsoft is given as an example of a keystone player (shown in red) and maps to business function Infrastructure software supplier. Any software company building applications on the Windows platform would be a niche player (shown in green) and maps to another business function Application software supplier.

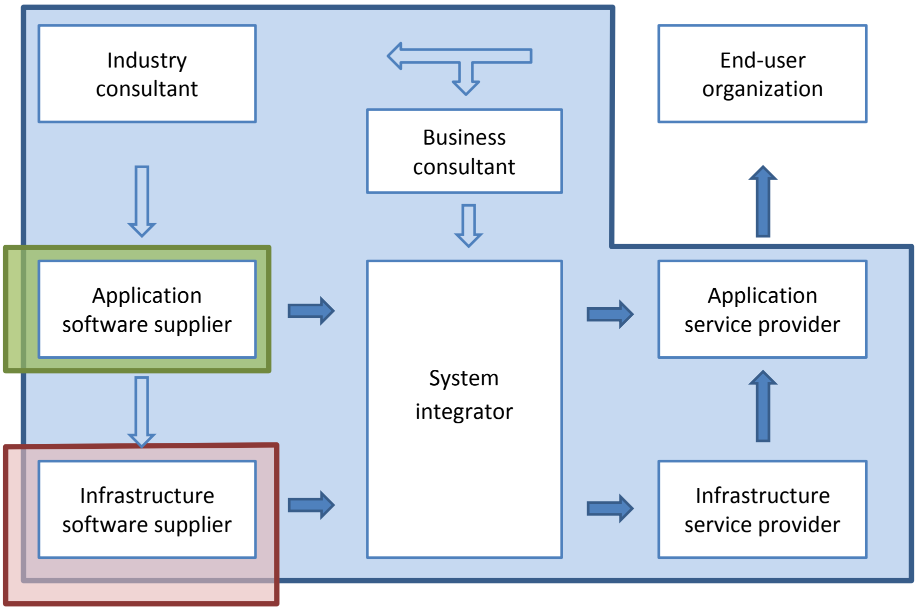

Industry consultant
Business consultant
End-user organization
Application software supplier
System integrator
Application service provider
Infrastructure software supplier
Infrastructure service provider

Figure 15 - Mapping of an example of a keystone (red), niche player (green) and a physical dominator (blue).

4.2. Platforms

Where ecosystems are mostly concerned with the relations between companies, users or actors in general in the ecosystem, platforms focus more on the products and services of those companies. Traditionally product development literature on platforms is focused on product platforms within a company. The two mayor subjects in this literature are component reuse and modular architecture. A typical example of literature on component reuse would be Utterback and Meyer who propose creating a product family based on a common platform (or core) as a way to develop more products in less time (Utterback & Meyer, 1993).

In contrast with platforms within a company Iansiti and Levien give a more abstract level definition of a platform: "(a platform is a) set of solutions to problems that is made available to the members of the ecosystem through a set of access point or interfaces." (Iansiti & Levien, 2004, p. 148). This definition moves the platform in the context of ecosystem. According to them the platform is exactly what provides value to other members in the ecosystem and makes the platform owner a keystone player.

Baldwin and Woodard work towards a more unified view of platforms (Baldwin & Woodard, 2008). They see a central role for platform products and services in mediating the activities of or ecosystems of firms. Three platform types are distinguished by Baldwin and Woodard: platforms in a firm as product lines, platforms across multiple firms as multi-product systems, and platforms in the form of multi-sided markets. They argue that a platform has a platform architecture and corresponding design rules that governs the relations between components of the platform and allow them to interoperate. According to the authors this architecture shows a fundamental unity for each type of platform. They describe this unity in platform architectures as "*modularizations of complex systems in which certain components (the platform itself) remain stable, while others (the complements) are encouraged to vary in cross section or over time."* (p. 2). According to them the most stable element of a platform are the interfaces between the platform and its complements, even more stable than the core of the platform itself. Finally they argue that control over the interfaces of a platform means control over the platform and its evolution itself.

4.2.1. Software platforms

Now that product and industry platforms have been explained the scope can be narrowed to the software industry. Software platforms are not really different from platforms in other industries. They might be more obviously visible in software than in other products however. The concepts of component reuse and modular architecture are a commonplace in software architecture, for example in object-oriented programming.

A type of architecture often found in software is the layered or multi-tier architecture. The software is then divided in logically separated processes, a common example of which is an architecture consisting of a presentation-tier, a logic-tier and a data-tier. Each layer typically only interacts with the directly adjacent layers and does this generally using well defined interfaces. Each of these layers can be seen as a platform for the layer on top of it, and it is not uncommon that underlying layers are built and maintained by another firm.

4.2.2. Industry platforms

Instead of traditional product platforms Cusumano is more concerned with industry platforms or platform ecosystems. Cusumano describes two fundamental differences of industry platforms with product platforms (Cusumano, 2010). The first difference is that an industry platform is a foundation, service or core technology that holds little value to its users without complementary products or services. A typical example would be a computer operating system such as Microsoft Windows being fairly useless without accompanying applications. The second mayor difference Cusumano describes is the creation of network effects. These network effects means that the more external players adopt the ecosystem and create or use the complementary innovations the more valuable the platform and its complements becomes. This allows for positive feedback loops where the increase of users allows for the increase of creators of complementary products, which will attract more users etc. A schematic model of a platform ecosystem is shown in Figure 16. See for a more detailed explanation on network effects also paragraph 4.3.1. .

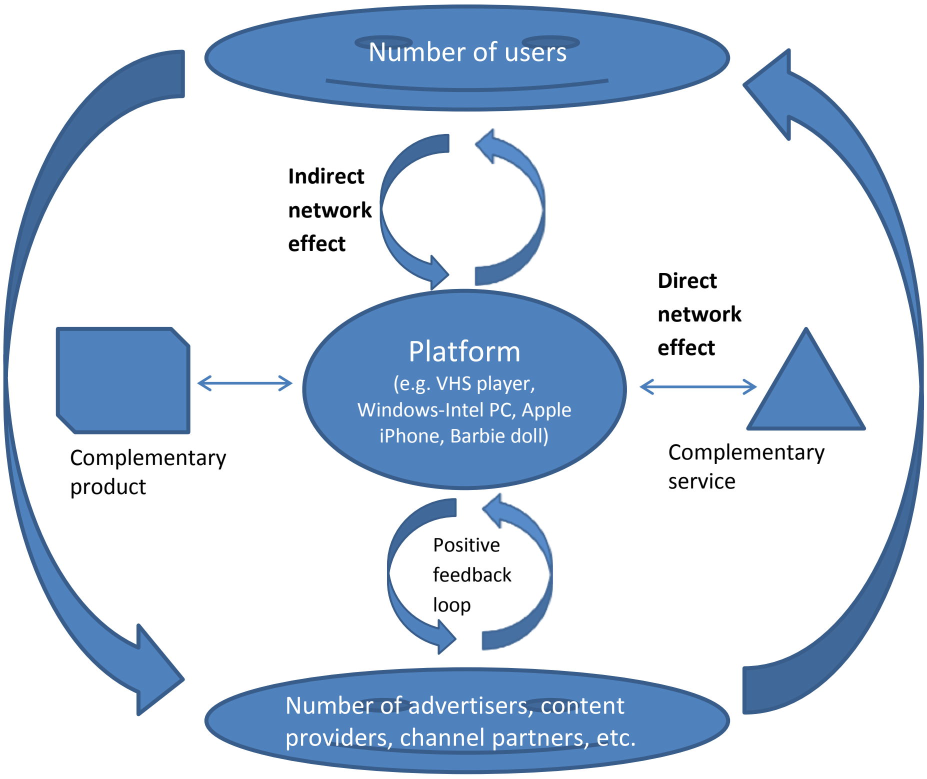


Figure 16 - The ecosystem of platforms, complements, and network effects (Cusumano, 2010, p. 25)

4.2.3. Software stack

Another approach that crosses the boundaries between a platforms and an ecosystem is one by Gao and Iyer. They describe a software stack using a layered view borrowed from software architecture that divides software activity into layers that are complementary to each other (Gao & Iyer, 2006). Generally companies operate in one or only a few layers. They argue that the layered structure has implications for the competition in the software industry. Competition mostly takes place within the layers where a software products compete with similar products for a greater use in products at higher layers. The software stack is depicted in Figure 17 accompanied by some examples for each layer.

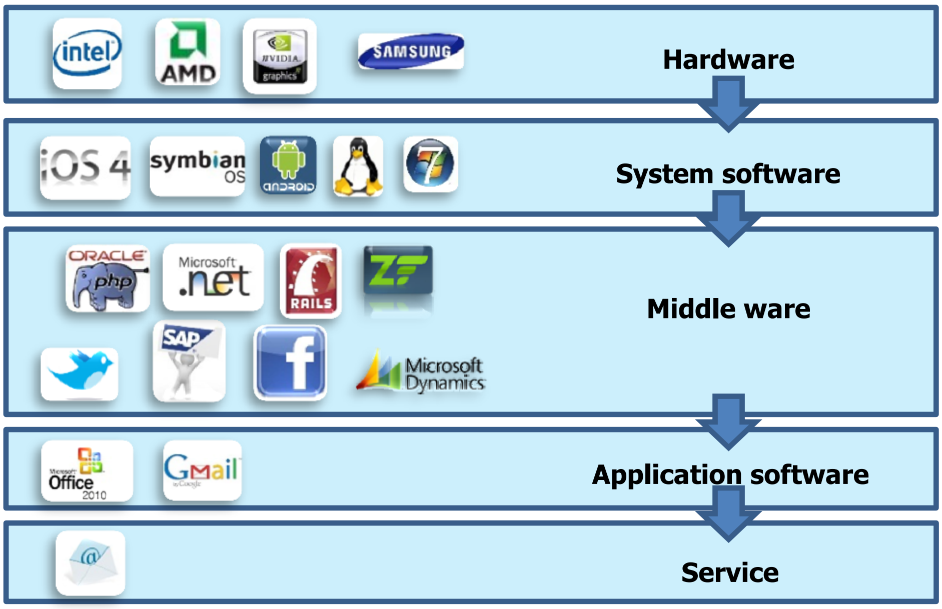


Figure 17 - The software stack with some examples (Gao & Iyer, 2006)

4.3. Platform mechanisms

In this section some textbook economics related to app stores are introduced. These theories will be used to get a better understanding of the reasons for app stores to exist and a deeper understanding of how app stores work economically. This is done by introducing the theory of “network effects” and based on this the more refined concept of “two-sided markets”.

The distinction between literature on two-sided markets and literature on network effects is not very clear. Rysman points out that literature on two-sided markets can technically be seen as a subset of the literature on network effects but in practice literature on two sided markets distinguish by their focus on the platform or intermediary and its actions, in particular with regard to pricing choices. Literature on network effects has more emphasis on adoption by users and optimal network size (Rysman, 2009).

4.3.1. Network effects

In economic and business literature a network effect is the effect that one user of a good or service has on the value of that product to other people. A celebrated example of a product with a network effect is a telephone. When one user has a phone it is quite useless. When more people however start using a phone the value of that one phone increases dramatically. If the value of a product or service is dependent on the number of other users of that product or service then a network effect is present.

In economic literature this effect is often referred to as network externalities. Even though this is a very similar concept Liebowitz and Margolis have argued that network effects is a better choice of words and therefore I will use network effects (Liebowitz & Margolis, 1998). In the literature I reference this might however not be the case and there the terms are used interchangeable.

A number of different types of network effects can be distinguished and are further described here. The first type are direct network effects. The aforementioned phone industry is a classic example of direct network effects. Direct network effects are the simplest form of network effects: more users lead to more value. This form was first extensively studied in 1985 in literature by Katz and Shapiro (1985) and also by Farrell and Saloner (1985). A well-known example related to network effects is Metcalfe law. This law was formulated with regard to the use of Ethernet cards and states that value of a telecommunications network is proportional to the square of the number of connected users of the system (Metcalfe, 1995).

A second type of network effects are indirect. Indirect network effects occur when the increases in usage of one product or network spawn increases in the value of a complementary product or network, which can in turn increase the value of the original. An example can be found in the market of DVD-players. The mere fact that other users have a DVD-player does not increase the value for a user, but the indirect effect that there will most likely be more of the complementary good namely DVD's available does affect the user. A framework for the economics of indirect network effects was first introduced by Economides and Salop (1992)

When the users of a particular product can be divided in two (or in some cases more) distinct groups there can also occur two sided network effects. The increases in usage by one set of users increases the value of a complementary product to another distinct set of users and vice versa. These effects are further refined into the concept of two-sided markets and are further elaborated in section 4.3.2.

The last type of network effects distinguished are local network effects. Sometimes the value of a product or service to a consumer is not determined by the total amount of users but from the amount of users in the neighborhood of the consumer. An typical example can be found in social networks. These networks are almost useless when none of your friends participate and increase in value when more of your friends join. These local effects are described by Arun Sundarajan (2007).

4.3.2. Two-sided markets

The concept of two-sided markets was first introduced by Parker and Van Alstyne (2000) and in 2001 almost simultaneously by Rochet and Tirole (2003). A rough definition as given by Rochet and Tirole is as follows: "Two-sided markets are markets in which one or several platforms enable interactions between end-users, and try to get the two sides "on board" by appropriately charging each side. That is, platforms court each side while attempting to make, or at least not lose, money overall." (Rochet & Tirole, 2006)

Rysman identifies two main elements in his definition of a two sided market: firstly at least two sets of distinct users interact through an intermediary or platform and secondly the decisions of each set of users affect the outcomes of the other set of users (Rysman, 2009). The latter part of this definition is called cross-side network effects as opposed to same-side network effects where decisions have effect within the same set of users. The platform and its sides with the both types of network effects are schematically depicted in Figure 18.

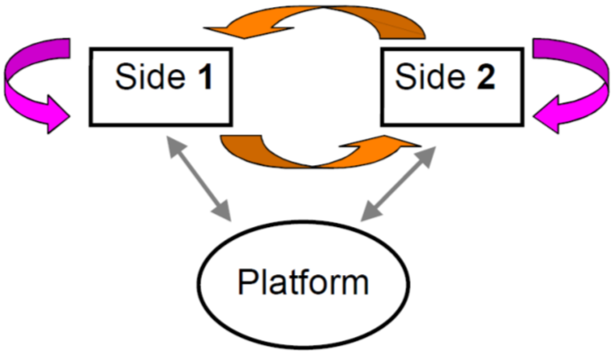


Figure 18 - Cross-Side and Same-Side Network Effects in a Two-Sided Network (Eisenmann T. R., 2006) [1]

Examples of two-sided markets can easily be found: credit card companies with merchants and card holders, operating systems with developers and end users, newspapers with readers and advertisers, recruiting websites with employers and job seekers.

4.4. Platform strategy

Now that the concepts of platforms and ecosystems are explained it is time to look at the question why a companies would consider following a platform strategy and create a platform or start a software ecosystem. The used literature generally takes the perspective of a keystone player (Iansiti and Levien) or infrastructure supplier (Messerschmitt and Szyperski) or, when looked at it from a platform perspective, the platform owner or creator.

Iansiti and Levien argue in their book "The keystone advantage" (Iansiti & Levien, 2004) that the health of business ecosystems can be measured by their productivity (in general and in terms of delivery of innovation), robustness and niche creation. Since current business ecosystems are of a highly interconnected nature, members of such an ecosystem share their fate with each other. Iansiti and Levien then introduce their keystone strategy as "an operating strategy that improves the overall health of the ecosystem and, in so doing, benefits the sustained performance of the firm" (p. 82) According to the authors, keystone players have a focus on managing external resources and maintaining ecosystem health. They do this by improving productivity and efficiency of their ecosystems in a variety of ways. Then they make sure that these efficiencies actually happen and share the impact with the ecosystem. The two core components of a keystone strategy are creating value in the ecosystem and sharing value within the ecosystem. Why this keystone strategy is advantageous to a company is then illustrated by the cases of Microsoft's operating system platforms and its growth over the last decades and by the case of Wal-Mart's integrated supply chain that made them the biggest retail company in the USA.

Cusumano agrees with Moore that innovation and the ability to adapt to market demands are the most important competitive advantage a company can have. For Cusumano the reasoning for following a platform strategy is thus the much greater potential for innovation and growth. Cusumano urges managers of industries affected by or network effects to compete on the basis of platforms or complement to the platform of another company (Cusumano, 2010).

---

[1] © Eisenmann, Parker, Van Alstyne. This file is licensed under the Creative Commons Attribution 2.5 Generic license.

In their book “Catalyst code: the strategies behind the world’s most dynamic companies“, Evans and Schmalensee introduce a different metaphor for the keystone player or platform owner: the economic catalyst (Evans & Schmalensee, 2007). They define an economic catalyst as “an entity that has (a) two or more groups of customers; (b) who need each other in some way; but (c) who can’t capture the value from their mutual attraction on their own; and (d) rely on the catalyst to facilitate value creating reactions between them. (p. 3)”. According to the authors the core functions of a catalyst are matchmaking, building audiences and minimizing the costs of running a community. They then carry on by explaining the more robust nature of a multi-sided business versus a single-sided business and introduce the existence of (indirect) network effects and the possibilities for positive feedback loops and thus fast growth. When a company would become a catalyst business and choose the right pricing structure this could become an engine of economic growth and profitability. Some of the concepts mentioned here are further explained in the next paragraph.

Hagel, Brown and Davison introduce their shaping strategy in an article in the Harvard Business Review in 2008 (Hagel, Brown, & Davison, 2008)**.** This strategy opposes the reactive stance of many companies which solely focus to keep up with relentless technological changes, try to protect their existing markets and try to improve their current performance. In contrast shaping companies are more proactive and use technology changes to create new business ecosystems that benefit themselves and other participants. The strategy consists of three important steps. The first step is to communicate a shaping view that highlights how a broad industry or market is changing and identifies the opportunities for a wide range of participants. The second step is to develop a shaping platform which according to the authors is *“a set of clearly defined standards and practices that helps organize and support the activities of many participants.”* (p. 7). This platform should make it easy and inexpensive for participants to develop and deliver their own products or services. The third step is to show shaper acts. This basically means to show the potential participants that a company has both the conviction and capability for success.

4.5. Platform governance

Where the preceding paragraph explained why having a platform is a good idea this paragraph focuses on the strategies or tactics used to govern a software ecosystem.

Eisenmann, Parker and Van Alstyne introduce three strategies for two-sided markets that can be applied to software ecosystem governance (Eisenmann, Parker, & Van Alstyne, 2006). The first one is concerned with pricing strategy. Their advice is firstly to consider subsidizing quality- and price-sensitive users (see section 4.5.3. on Pricing Strategy) and secondly to secure “marquee” users’ exclusive participation. Their second strategy explains how to cope with winner-take-all competition between competing ecosystems. As a third strategy they advise to cope with competing ecosystems by avoiding envelopment and being open for changes in the business model. The last two strategies are further explained in section 4.5.1. on Ecosystem competition.

4.5.1. Ecosystem competition

In many cases the positive feedback loop of two-sided network effects can lead to winner-take-all battles between software ecosystems. In the their paper Eisenmann, Parker and Van Alstyne list three conditions that indicate that a certain market is likely to be served by one platform: a) multi-homing costs are high for at least one user side, b) network effects are positive and strong - at least for the users on the side of the network with high multi-homing costs and c) neither side’s users have a strong preference for special features (Eisenmann, Parker, & Van Alstyne, 2006). When an market indeed is likely to be served by a single platform they advise platform owners to decide whether they want to share or fight for the propriety control. When a company decides to fight for control they at least need cost or differentiation advantages. Other helpful assets are deep pockets, a reputation for past prowess and preexisting relationships with prospective users.

Another strategy Eisenmann et al. suggest to deal with competition between ecosystems is to avoid envelopment. According to the authors platforms have overlapping user groups tempting some related platform providers to swallow others' users. They give the example of Real Networks, a company that specialized in streaming media software but was under pressure of Microsoft. They used their existing customer relations to start a new subscription music service called Rhapsody. So instead of being swallowed they chose to change their business model. Other options the authors suggest is finding a (preferably bigger) partner that can provide value to your ecosystem or to consider taking legal action based on the still underdeveloped anti-trust law for two-sided markets.

4.5.2. Innovation strategy

According to Eaton et al. there is a tension between control and generativity in service innovation for digital ecosystems. (Eaton, Elaluf-Calderwood, Sørensen, & Yoo, 2011) Creating an ecosystem allows a company to leverage external resources in generating service innovations. At the same time a company needs to exercise a some form of control on the ecosystem to be able consume the economic returns of the generativity. Too much control however stifles generativity as it can act as an economical, technical or psychological barrier for external resources to participate. According to the authors "The core challenge of service innovations in a digital ecosystem is to continuously engage in balancing control and generativity." (Eaton, Elaluf-Calderwood, Sørensen, & Yoo, 2011, p. 6).

Ghazawneh and Henfridsson also see this challenge and have made it the central question of a research paper (Ghazawneh & Henfridsson, 2010). They tried to find the governance process by which platform owners can simultaneously maintain platform control and stimulate third-party development through platform boundary resources . To do this they have done an empirical analysis on the use of boundary resources to govern the Apple iPhone innovation network. As a result the authors found three boundary resources that Apple had used: the software development kit (SDK), the APIs and the developers' agreement. The governance process that they derived from their research is depicted in Figure 2. It is a cyclical pattern that is driven by the owner of the platform and its changes in the boundary resources over time. The platform owner can create a new boundary resource design, secure its use by rewriting the agreements with other platform members, use its distribution channels to increase the knowledge heterogeneity and counteract other boundary resources that may pose a thread the ecosystem.

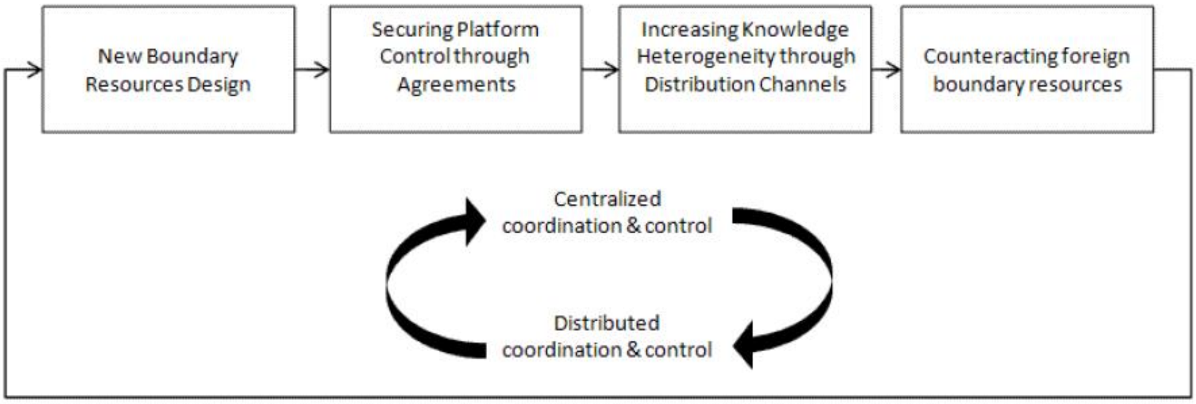

New Boundary Resources Design
Securing Platform Control through Agreements
Increasing Knowledge Heterogeneity through Distribution Channels
Counteracting foreign boundary resources
Centralized coordination & control
Distributed coordination & control

Figure 19 - A process perspective of governing third-party development

4.5.3. Pricing Strategy

Pricing strategy in two sided markets is one of the most extensively researched topics within two-sided market literature. Economists create formal models of two-sided markets and use them for example to prove why it makes sense for some companies to give away products for free. Since most of this literature is too technical within the context of this master thesis I will only introduce an example based on Eisenmann, Parker and Van Alstyne to illustrate this point (Eisenmann, Parker, & Van Alstyne, 2006).

Adobe is a software company that has created a portable document format (PDF) and builds reader software (Adobe Reader) for consumers and writer software (Adobe Acrobat) for document creators. Using traditional pricing logic Adobe would determine the price of their products by seeking the biggest revenue rectangle (price × quantity) under each demand curve, as depicted in Figure 20.

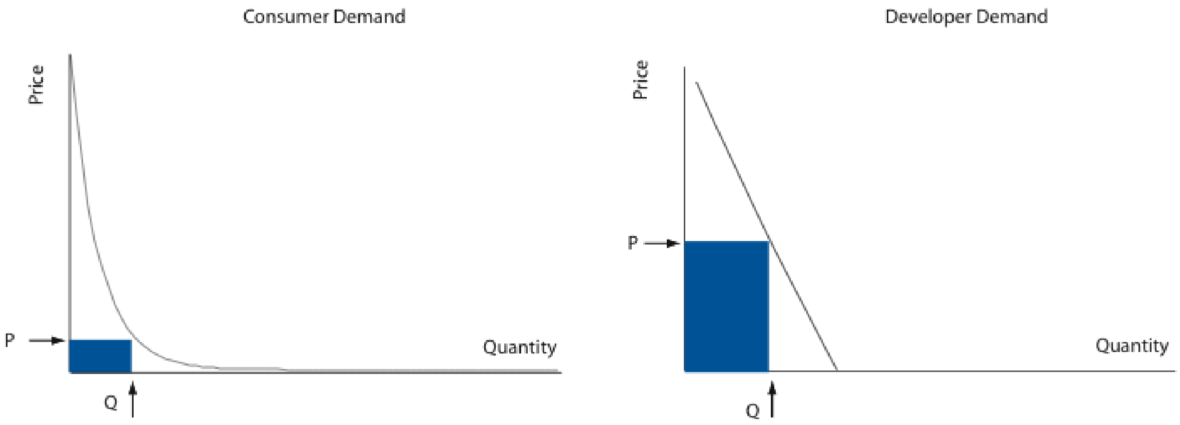


Figure 20 - Traditional pricing logic. Based on (Parker & Van Alstyne, 2005, p. 1498) [2]

In a two-side market however there can be cross-side network effects. When Adobe decided to give away there reader software for free this not only affected the reader side of the market but also affected the demand curve of the document creator's side. The enormous amount of people who can

---

[2] © Eisenmann, Parker, Van Alstyn. This work is licensed under the Creative Commons Attribution 2.5 License.

read PDF documents makes it much more appealing for document creators to use Adobe Acrobat software. In Figure 21 this more elaborate pricing logic is shown. So long as the revenue gained (red box) is higher than the revenue lost (light blue box), this strategy is profitable.

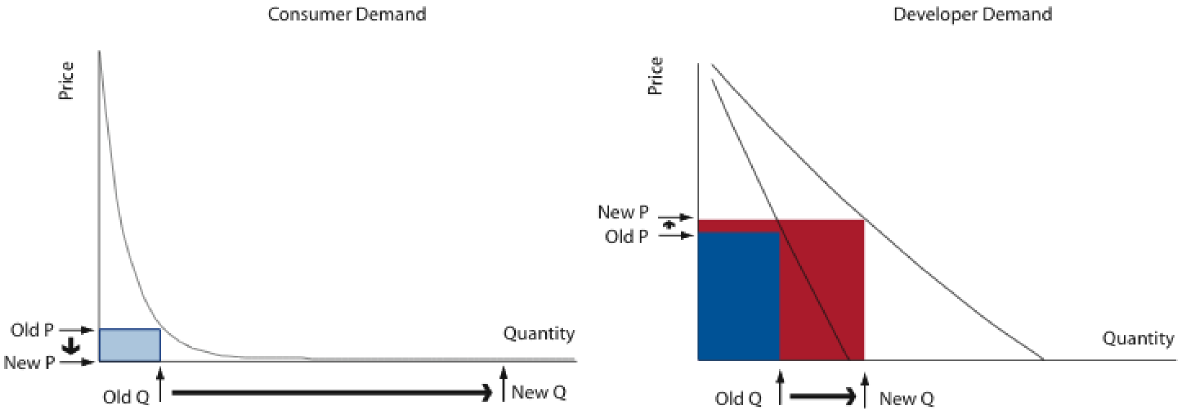


Figure 21 - The subsidy largely changes network size. Based on (Parker & Van Alstyne, 2005, p. 1498) [2]

The example has illustrated that subsidizing one side of a two-sided market can be a very profitable price strategy. It is however not trivial to choose which side of the market to subsidize. Eisenmann et al. suggest six factors that help determine which side to choose: The ability to capture cross-side network effects, the user sensitivity to price, the user sensitivity to quality, the output costs, the existence of same-side network effects and the users' brand value (Eisenmann, Parker, & Van Alstyne, 2006).

4.5.4. Openness

According to Rysman the openness strategy of a platform owner consists of two main issues (Rysman, 2009). The first one is the number of sides this platform is going to pursue. A company like Apple builds an operating system but also keeps close control over the hardware side in their platform. Microsoft however leaves this side wide open for others. In this example Apple's OS has a less open strategy and would be characterized as a two-sided platform between the consumers and software-providers whereas Microsoft's OS can be characterized as a three-sided platform between consumers, hardware manufacturers and software-providers.

The second issue of openness is according to Rysman the compatibility with competing platforms. The options for a platform are to be incompatible, to compatible or to integrate in some form. An example given by the author of openness in this sense is an ATM machine. Most ATM machines choose to integrate in some form with other platforms by allowing members of other banks to use their machine, often in a limited way.

4.6. App store strategy model

This paragraph will introduce a new app store strategy model that could be derived from the studied literature.

4.6.1. Identifying main concepts

From the literature three main aspects of a platform strategy can be derived. First of all the competitive advantage that comes with the higher level of innovation possible when leveraging the development power of the ecosystem. Both Moore and Cusumano as well as Iansiti & Levien identify this as one of the main drivers behind companies pursuing a platform strategy (Moore, 1996) (Iansiti & Levien, 2004) (Cusumano, 2010). This high level of innovation should however be balanced with the amount of control a platform owner exercises (Ghazawneh & Henfridsson, 2010). A second driver for a platform strategy is the existence of network effects that can provide enormous growth opportunities. Both Cusumano and the research on network effects and two-sided markets of (amongst others) Rysman show how cross-network effects allow the creation of positive feedback loops that can make markets grow very fast (Rysman, 2009). However according to the pricing strategy in order to start such a positive feedback loop the owner might typically have to subsidize one of the sides of the market (Eisenmann, Parker, & Van Alstyne, 2006). The third driver for adoption of a platform strategy is the opportunities of value sharing in the ecosystem. Iansiti and Levien introduced the concept of shared fate within the ecosystem, i.e. the notion that a healthy ecosystem benefits al actors within the ecosystem. They identified that the two core components of a keystone strategy are creating value in the ecosystem and sharing value between the members of the ecosystem.

Now that these three main aspects of the platform strategy has been identified they need to be made more specific to be used as an app store strategy. In order to do this we first have to consider the narrowing scope: from software ecosystem and platform to an app store. A platform strategy in literature generally assumes that a company is already the owner of the platform or will become the owner of a platform. However with regard to app stores this is not always the case. More often than not are app stores owned by another actor in the ecosystem than the platform owner, see for example the existence of many app stores for the android platform owned by Google. These app stores directly compete with Google's own Google Play. Since app stores can be regarded as a part of the platform services, app store owners could be considered partial platform owners by default. However since their control of the platform does not extend beyond the app store services this is strategically not very relevant. Of course the fact that an app store owner is not the owner of the software platform influences the strategy they will have to use in order to make their app store successful. So in order to refine a platform strategy into an app store strategy the position of a company within the ecosystem cannot be ignored.

4.6.2. App store strategy model

In order to classify app store strategies into a model the model in Figure 22 is proposed. The model uses four factors to describe an app store strategy. The first factor describes the position of the app store owner in the software ecosystem. It makes a distinction between app store owners that are also owner of the software platform and app store owners that rely on someone else's software platform. The second factor makes a distinction based on the strategic goal of the app store owner. It uses the concepts of value creation and value sharing derived from Iansiti and Levien. A value creation strategy aims to create value for the entire ecosystem. An app store strategy more focused on value sharing is more concerned with the monetization of the platform, not necessarily just for themselves but also for the ecosystem as a whole.

The third factor makes a distinction based on the balance between control and innovation. A governance style based on control will for example put more strict curation of the app store content, whereas an innovation style will be more welcoming to diversity and apps pushing the boundaries of the platform. The fourth factor in the model is concerned with the way the app store owner tries to initiate the positive feedback loop. It is based on the economics of pricing strategy but within the scope of app stores it is intended to go beyond the financial drivers. It reflects on the way that app store owners try to get a side to join their app store by subsidizing that side.

The app store strategy now can be represented by a positions on each of the axis of a factor.

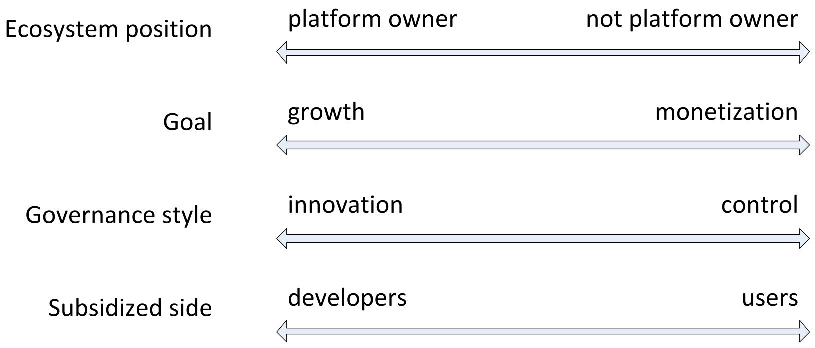


Figure 22 - App store strategy model

The subsidized side factor is now ranging from developers to users. It could be argued however that a company with deep enough pockets could very well subsidize both sides of the market. Therefore each axis has to represent the factor relative to the other side. So a position in the center of the subsidized side axis would represent subsidizing both sides equally rather than not subsidizing any.

It is also likely that the four factors are not completely independent of each other. The most obvious example would be the connection between ecosystem position and the strategic goal. An app store owner who is not a platform owner is unlikely to have a strict growth goal since that would very much be a philanthropic exercise. For a platform owner however it is very possible to have a strict growth focused strategy. Google for example has had a strict growth strategy for their Android platform in order to ensure a strong presence in the fast growing mobile market. This presence in the mobile market would ultimately result in a strong position in the mobile search market, a goal much closer to the core business of Google. Note that this example shows it is possible that seemingly inconsistent app store strategies are the result of the overall ecosystem strategy of an organization. Two other factors that are likely not completely independent are having a growth goal and an innovation governance style. An innovation governance style would most likely allow more apps to enter the app store than a more controlled governance style. Allowing more apps would also have an effect on the growth of the number of apps in the app store which would suggest some dependency between growth and innovation.

5. Results case study research

This chapter shows the result of the multiple case study. The goal of the six performed case studies was to create a model that can be used to classify existing app stores. Such a model should consist of a number of measures that can accurately describe the features and policies of an app store. In order to create this model for each of the six selected app stores data was gathered on its features and policies. This was done in a sequential order where each case study used the results of the previous case study as a starting point. This way each case study resulted in a more refined and more extended version of the classification model. In order to create a starting point for the first case study an initial app store classification model was created by performing some preliminary research. This preliminary research used existing literature comparing existing app stores to create an initial model. A more detailed description of the followed research method can be found in the field procedures part of the case study protocol in paragraph 3.7.

Note: all citations in block quotes (e.g. [7]) in the case studies refer to a document from the document database belonging to the respective case study. An index of this document database for each case study can be found in the case study's report in the appendices.

5.1. Case selection

In order to select the cases that will be researched first an long list was created. This long list should contain an overview of currently existing app stores, but since there is no authoritative source listing all app stores it is unfortunately not exhaustive. The long list has been created by reviewing the search results for keywords related to app stores. At first Google was searched using keywords "app store", "application store", "marketplace", "app store comparison", "app store launch". These search results however largely consisted of many similar articles describing only a small number of well-known app stores. So instead these searches were performed on a major technology blog that regularly publish about app stores TechCrunch. These searches allowed better filtering of the output and also searched Crunchbase, a business directory focusing on the technology sector. After reviewing 1500 search results from this source a long list of app stores was created. In order to extend this list even more two existing lists from Wikipedia and Distimo were used to add even more app stores. This resulted in a list of 81 currently existing app stores which can be found in Appendix D – Long list. Each app store is listed together with their platform type (mobile, web, desktop, other) inspired by the software ecosystem taxonomy by Bosch (Bosch, 2009). The other category features app stores for TVs and robots and code only app stores that do not adhere to only one other category. A breakdown of the number of app stores found in each category is given in Figure 23.

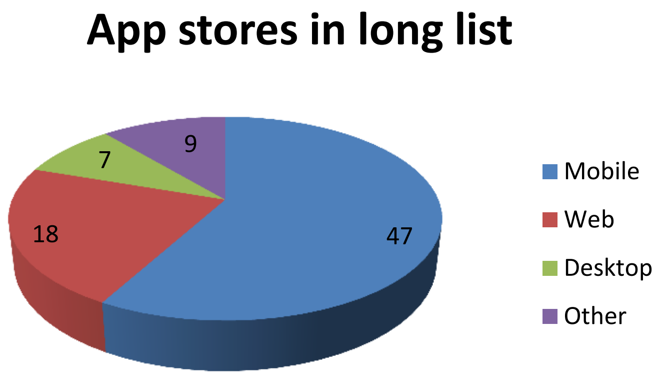


Figure 23 - App stores in long list

In order to select the app stores from the long list that were used for the actual case studies three sets of criteria were used. The first set is concerning the definition of an app store. The second set of criteria deals with the accessibility of the data. The last set of criteria aims to improve the generalization of this sample.

Criteria concerning the definition:

- It should be available using the internet.
- It should be curated by an owner.
- It should allow for the selling and buying of software products.
- It should take care of the financial transactions involved in selling the software products.
- It should have two distinct user groups: developers and users.
- It should be serving one or more software ecosystem.
- It should implement a platform that takes care of the distribution of the software products.

Criteria concerning accessibility:

- It should allow researcher to get a complete overview of the app store features and policies.
    - Access to the end user side of the app store
    - Access to the developer side of the app store
    - Access to a device running the apps / code available on the app store and
    - The ability to perform a download / transaction from the app store to the device.

Criteria concerning generalization:

- The dataset should include cases from each dimension of the software ecosystem taxonomy (Bosch, 2009).
- The dataset should include cases from multiple vendors.
- Only active app stores with over 1000 transactions are considered.
- It should at least be 6 months old.

This resulted in the following six app stores that were selected for the case study: **Google Play**, **SlideMe**, **Apple Appstore**, **Binpress**, **Amazon appstore for Android** and **Intel AppUp**. It must be noted that in most cases the accessibility was the most important selection factor, since it is for example very hard to study the inner workings of an appstore for Samsung TVs without having access to such a device. Google Play, SlideMe, Apple Appstore and the Amazon appstore for Android are all app stores for mobile platforms, Binpress is a source code only app store and Intel AppUp is mostly a desktop platform. In Table 1 some usage statistics of the six selected app stores is listed. It should be noted that since most app store owners do not always provide up to date information the usage statistics originate from many different sources and are not all from the same time. More detailed usage statistics and the sources used can be found in the respective case study reports.

	# apps	# downloads	paid (%)
Google play	450.000	10.000.000.000+	28%
SlideMe	20.000	n.a.	26%
Apple appstore	650.000	30.000.000.000+	66%
Amazon appstore for Android	30.000	2.000.000+	65%
Binpress	369	n.a.	100%
Intel AppUp	3.000	810.000	64%

Table 1- Comparison of usage statistics of selected app stores

5.2. Preliminary research

In order to create the input for the first case study an initial model was created. First of all scientific literature on this subject was consulted. This resulted in a number of papers (Kim, Kim, & Lee, 2010), (Holzer & Ondrus, 2011) in which some measures for app stores (in a mobile context) where given. In a similar fashion three Google searches were performed with search queries “app store comparison” and “Android market vs. app store”. The search was limited to display only results from the last year or the last two years and in the one case also to display only results from blogs. This resulted in 20 blog articles or webpages that compared two or more app stores with each other. The comparisons differed in perspective from a very developer centric perspective to an exclusively end user perspective and anything in between. The different measures mentioned in each comparison where compiled into a list and then the duplicate entries were removed. This resulted in a list of 59 measures as listed in Table 2.

app approval process
app discovery
app distribution
app distribution integration
app quality
app quantity
app search quality
app security
app store growth
app submission policy
average spending per user
benefit sharing
centralization of portal
control over app
deals
developer story
developers cut
device compatibility
device control
device integration
ease of app distribution
entry barriers
friendly attitude
geographical availability
growth of the platform
in-app advertising
integration with website
interface
learning costs
market demand
market size
number of apps
number of developers
number of end users
paid vs. free apps
pay out methods
payment options
platform entry barriers
platform integration
platform openness
possible promotion
price policy
refund policy
relative price point
removal policy
required development effort
resolution of copyright issues
return policy
revenue comparison
revenue potential
review features
review process fairness
setup costs
store growth
termination costs
try before buy
usefulness of tooling
user spending profile
website integration

Table 2 - Measures from app store comparisons in literature

A lot of the 59 measures are similar or at least a bit overlapping and thus the lists required some substantial grouping and editing. At first the list was split into two groups: a group of measures that can be directly influenced by the app store or platform owner and a group of measures that cannot be directly influenced by the owner (but might be indirectly influenced). The first group consisted of 42 elements and the latter of 17 elements. The group of measures that cannot be directly influenced was named “characteristics”, the other group was split into a group named “features” and a group named “policies”, in correspondence with the conceptual model. After reviewing the content of the three groups a fourth group was added that contains the measures that were referring to the software ecosystem as a whole as opposed to the app store itself. This group will be used to describe the position of the app store in relation with the software ecosystem. This resulted in the initial app store classification model as depicted in Figure 24. This model can by no means be considered to be a complete or even accurate classification model but it will be used as a starting point for the case studies.

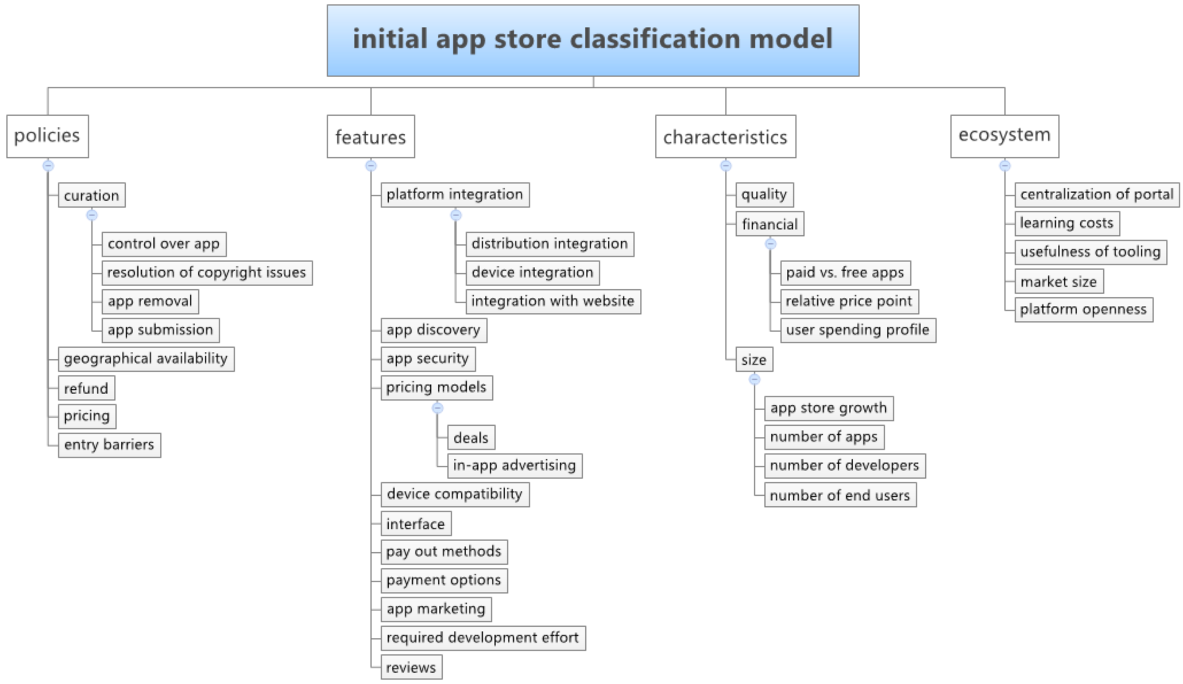


Figure 24 - Initial app store classification model

5.3. Case descriptions

5.3.1. Case study 1 (Google play)

The Google Play app store is the biggest app store in the Android ecosystem. It was launched by Google Inc. as the Android Market at 23 October 2008. In March 2012 is was re-branded to Google Play. It serves the Android ecosystem, which is an open source operating system for mobile devices and tablet computers created by the Open Handset Alliance.

From its inception in 2008 it allows developers to sell applications and games to end users. After its merger with Google Music it was re-branded to Google Play and it became a digital multimedia content service that also sells books, music and movies.

The scope of this case study is limited towards the distribution of apps and games only, since these fall within the definition of an app store.

As of February 2012 the Android platform had more than 300 million devices activated and Google Play listed over 450000 apps and games [70]. In December 2011 Google announced it had served over 10 billion downloads from the Android Market and a growth rate of 1 billion downloads a month [71]. 28% of the listed apps are paid while 72% is offered for free [72].

5.3.2. Case study 2 (SlideMe)

The SlideMe app store introduces itself as the third app store in the Android ecosystem, after Google Play and the Amazon Appstore. It was launched April 2008. It serves the Android ecosystem, which is an open source operating system for mobile devices and tablet computers created by the Open Handset Alliance.

SlideMe provides an alternative app store for devices that for some reasons cannot install Google Play. Over 120 OEM devices use SlideMe as their alternative app store. For users with Google Play the SlideMe app store can be a secondary app store that provides content that is not geographically available on Google Play or alternative payment methods.

One of the most compelling features of SlideMe is that they do not charge fees other than the payment processing fees. SlideMe earns its money by providing white-label curated app stores for OEM devices.

As of March 2012 the Android platform had more than 300 million devices activated and a count on SlideMe listed over 20000 apps and games [7]. 26% of the listed apps are paid while 74% is offered for free [7].

5.3.3. Case study 3 (Apple App store)

After the launch of the revolutionary iPhone in January 2007 apple waited 16 months before unveiling their next big revolution in the mobile phone business: the introduction of the Apple Appstore on 10th of July 2008 [100, 101]. The Apple appstore is the most successful app store in the market of mobile devices and is probably the best known too. The app store hosts applications for mobile devices made by Apple running the iOS platform, namely the different versions of the iPhone and iPad tablet. From the start the Apple Appstore was integrated with their already popular iTunes music download service [101].

Under app developers the Apple appstore is notorious for its strict approval process that apps have to go through before publishing. It was also the first app store featuring in app billing and content subscriptions [102, 103]. The number of revenue models together with the success of the iPhone device have attracted a large number of developers ensuring a rapid growth of the number of apps immediately after its inception.

In June 2012 more than 30.000.000.000 downloads were reported from the Apple Appstore. In that month over 650.000 apps were available in the Apple Appstore, making it the biggest existing app store. [98]. It features over 200% growth in number of app downloads since hitting the 10th billion download in January 2011 and around 85% growth in number of apps from 350.000 also in January 2011 [99]. Of all the iPhone apps in the Apple Appstore 65% has to be paid for and of the iPad apps 71% is paid [106].

5.3.4. Case study 4 (Binpress)

Binpress is a marketplace for source code, or as they describe themselves on Crunchbase: "Binpress offers a platform for software development companies and developers to publish and discover source-code components for rapid development. Source-code components offered on the marketplace are curated to meet industry standards and best practices, offering best-in-class solutions for common needs in software development that reduce the time and costs of developing software." [78]. Binpress was founded in January 2011 by Lionite, a web development shop and startup incubator located in Tel Aviv, Israel [66, 78].

Binpress hosts source code for a variety of 12 programming languages with 16 frameworks and 18 platforms. According to Binpress its main advantage over its closest competitor CodeCanyon are their lower commission rates, with a rate of 30% compared to 30% - 75% of CodeCanyon. Also Binpress does not require exclusivity of the offered source code to their store [63]. Because Binpress is a marketplace that sells and buys from developers it offers quite a few features tailored to their needs, such as issue tracking and component support forums.

In June 2012 Binpress hosted 369 components [2]. Sales numbers are not published but the site boasts 463799 hours of saved development time. Since developers can estimate the saved development time themselves this does not give any information on the actual sales amount. For example their most popular component estimates 200 hours of saved time and the eight most popular claim to save 118 hours on average. If this was representative for all components in the store this would amount to over 3900 sold components in total, however this should be regarded as a wildly speculating guess.

5.3.5. Case study 5 (Amazon appstore for Android)

The Amazon appstore for Android opened March 22nd 2011 and is the second most important app store in the Android software ecosystem [130]. Founded by online retail giant Amazon it offers some serious competition to the original Google Play app store. The app stores website piggy backs on the existing web retail infrastructure of Amazon and its massive user and credit card details database.

The most eye catching features that distinguish the Amazon appstore for Android from Google Play are the availability of the Deal of the Day promotion and the possibility to test drive apps before buying [130, 131]. The deal of the day promotion is a daily promotion which lets users download a normally paid app for free, for one day only. The developer of the app will in that case receive 20% of the list price instead of 70% of the sales price which they would normally get. It is rumored however that Amazon uses the increased exposure offered by the deal of the day to get developers to agree to a 0% revenue share [133].

The main disadvantage of the Amazon appstore for Android is that it is not installed by default on most Android devices. Users have to manually install the native app store application on their device thereby allowing apps from unknown sources to be installed, a potential security risk [117]. This disadvantage of course does not exist with the popular android based Kindle Fire devices sold by Amazon themselves. For these devices the Amazon appstore for Android is installed by default and the preferred method of downloading apps.

The number of apps in the Amazon appstore for Android is reported to be have grown from around 4000 at its inception to over 31000 [129, 132] at its first birthday on March 15th 2012. Rather than providing an exact number the amount of downloads was described by the Amazon executive as “millions of apps and games” [132]. The amount of paid application versus the amount of free applications in the Amazon appstore is about 65% [135].

5.3.6. Case study 6 (Intel AppUp)

On the 7th of January 2010 semi-conductor manufacturer Intel announced at the 2010 Consumer Electronics Show in Las Vegas the launch of the beta version of their app store called Intel AppUp [88]. The initial focus of the Intel AppUp store was on applications for the at that time popular netbook devices, and more specifically on netbooks that used the Intel Atom platform [89]. Supposedly using its leverage as chip supplier Intel negotiated with netbook and laptop manufacturers such as Asus, Dell and Samsung to distribute their app store pre-installed on their devices [90]. On the consumer side Intel made partnerships and created many affiliate stores for retail companies such as Staples, Croma, Dixons, Best Buy and Walmart [82, 90].

Described in 2011 by Peter Biddle, the head of the Intel AppUp store, as “the world's largest app store that nobody's ever heard of” [71] it currently allows transactions from 47 countries. Nowadays the Intel AppUp store hosts apps for platforms Adobe Air, Microsoft Silverlight, the Linux-based MeeGo operating system for mobile devices and last but not least native Windows XP and Windows 7 applications. It features apps in 22 categories ranging from Books and Business to Travel, Utilities and Weather.

According to Peter Biddle the mission of AppUp is to “curate app development and distribution tools in such a way that ‘mom-and-pop’ developers can create not just their own apps easily, but manage their own distribution in app stores of their own.” [71]. Even though the apps on AppUp are currently also sold in more than 20 affiliate stores almost all of them are owned by big chain retail companies, thus indicating that the original plan has not materialized yet [82]. In 2011 Intel AppUp moved away from its initial focus on netbooks since that market plummeted after the popularity of tablets. The new focus became normal PCs and laptop computers , and ultrabooks in particular [91, 92]. The abandonment of the MeeGo platform made Intel decide to start supporting the Tizen platform, a combination of the best of MeeGo and LiMo and more importantly HTML5 apps [92].

In 2011 it was reported that 20,000 developers that filed their tax information for the AppUp developer program [71]. At that time the app store had 350,000 registered users with 810,000 downloads between them [91] from a choice of over 3,000 apps [93].

5.4. Model evolution

For each case study data was gathered from three different sources. First scientific literature regarding the app store was searched and if found added to the document database. Next using direct observations by the researcher all the visible features and policies were documented using screenshots and added to the document database. Lastly data was gathered by searching he documentation provided by the app store owner and documents provided by other parties, mostly from larger blogs. The next step was analysing the data that was collected. Each case study used the model provided by the previous study to create an app store characterization. If the collected data suggested improvements to the model these were applied to the model. In most cases this meant that when a feature was observed at an app store that could not yet be represented in the model this was added to the model. In other cases some elements of the model were renamed to be more generic and thus include the observed feature or policy. This resulted in a more refined app store characteristics model after each case.

	# Features	# Policies
Initial model	14	8
after Case study 1: Google Play	28	10
after Case study 2: SlideMe	35	17
after Case study 3: Apple Appstore	40	21
after Case study 4: Binpress	57	20
after Case study 5: Amazon appstore	64	24
after Case study 6: Intel AppUp	67	24

Table 3 - App store classification model growth

The evolution of the model throughout the case studies is illustrated with the growth in numbers in Table 3 and a visual representation in Figure 25. For each case study the resulting case study report is listed in the appendices, C1 to C6 respectively. In each of the case study reports the refined model is listed together with a short explanation on the refinements, renamings, additions and other changes and the reasoning behind the changes made. After the last of the six case studies the app store classification model consisted of 67 different features and 24 policies. In Figure 26 the model is shown as it was after the sixth and last case study of the Intel AppUp app store. The model elements are categorized if deemed useful and ordered alphabetically.

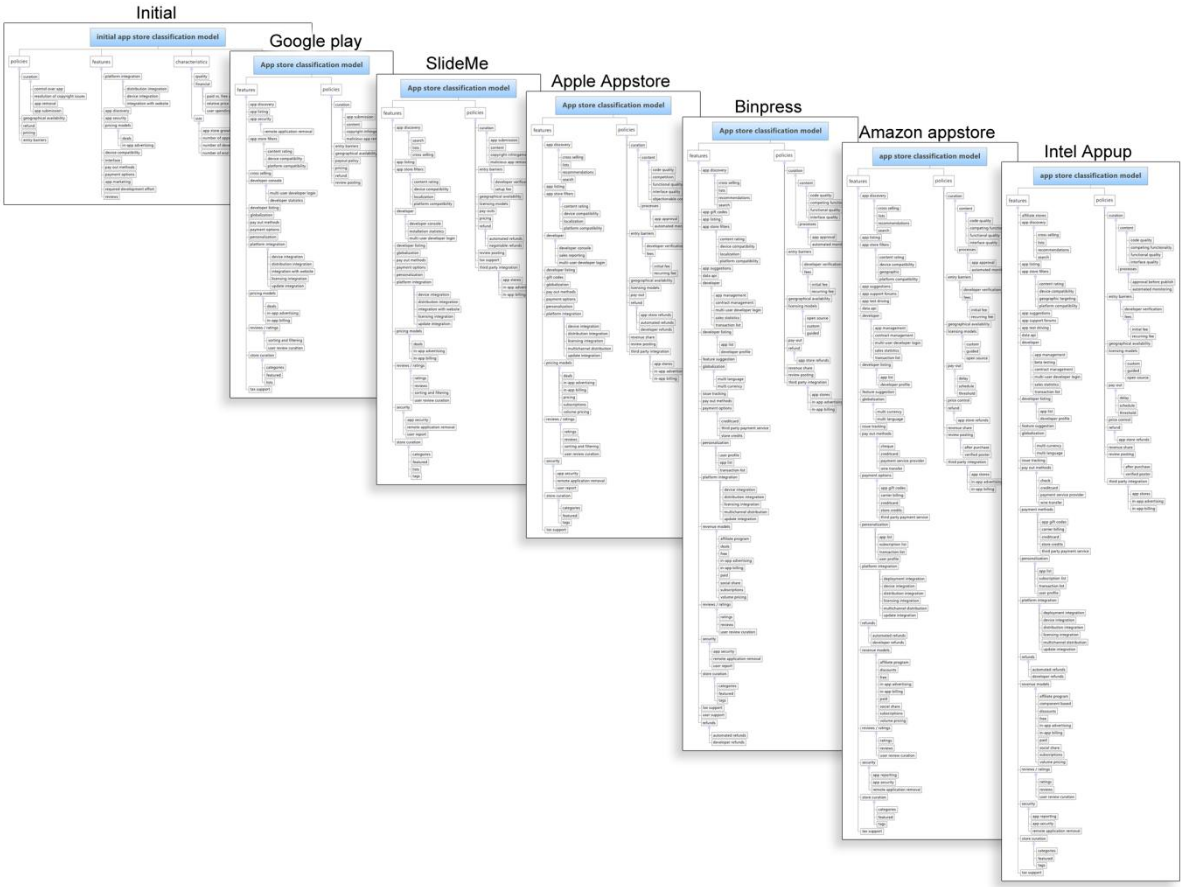


Figure 25 – Visual representation of model evolution between case studies

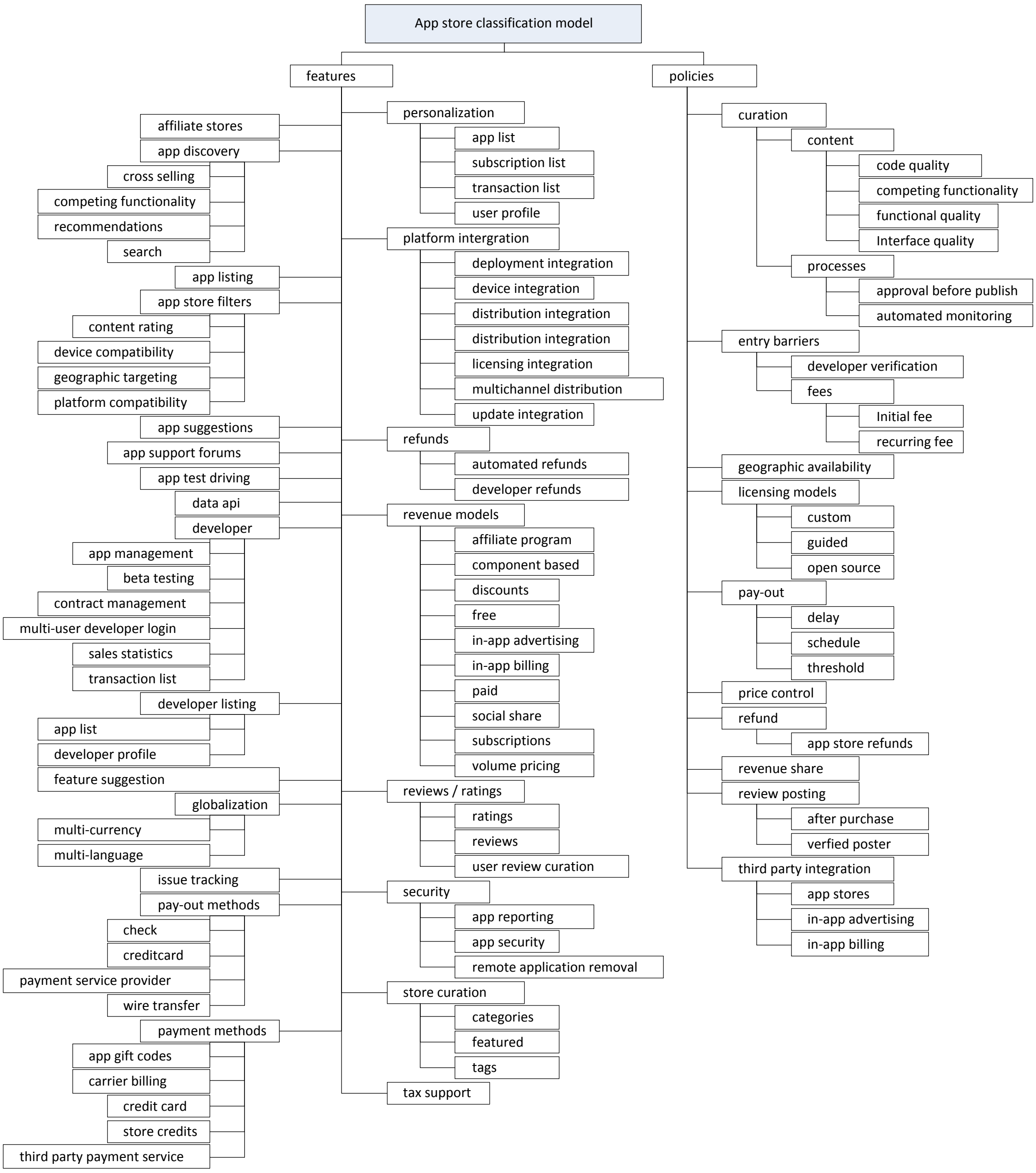


Figure 26- App store classification model after last case study

5.5. Model finalization

5.5.1. Element descriptions

After the first round of case studies had been performed the model that resulted from the last case study has to be applied to all the earlier case studies. In order to do this first for each feature and policy in the model a description had to be finalized. This description has to be descriptive enough that it can be used to identify the feature of policy in the app store. The data from all case studies was used to determine the proper scope for each description, such that it would fit in all six cases. An excerpt from the table with an example of the descriptions can be found in Table 4. The full table with all the descriptions can be found in Appendix A – Model element descriptions.

Feature name	Description
. .	. .
app security integration	The appstore integrates with a platform security system such as app privileges.
app security reporting	Users can report apps to the app store owner that they consider a security risk or otherwise harmful
app test driving	Users can test drive apps before purchasing them.
content rating filter	Apps get a content rating which depict the appropriateness of the apps content, generally divided for children of different age groups. Users set a level of content rating they would like to show.
device compatibility filter	Apps have a list of compatible devices. Users can filter for apps that are compatible with their device. Partial availability would be when apps are filtered on device capabilities.
remote application removal	The owner can remove harmful apps from the end user device without the help or consent of the end user.
. .	. .

Table 4 - Excerpt from table with feature descriptions (see Appendix A – Model element descriptions for full table)

5.5.2. Revisiting the case studies

Now that all features and policies had been described the document database for each case was revisited and used to fill the model with values. Most of the features were marked with “yes”, “no” or “partial” in order to describe their presence in the appstore. In a few cases a numeric value was used (e.g. payment methods) and in other a monetary value (e.g. pay-out threshold). In case a certain features or policy was obviously not applicable to the app store in case it was marked as “na” (not applicable). This is the case when for example an appstore has no content subscriptions model, in that case there was obviously no subscription list available too. If needed a note was added to clarify and give extra information, in particular with features marked as partially available. Lastly each feature and policy is accompanied by one or preferably more reference numbers. The reference numbers refer to the documents in the document database that form the basis on which the feature was identified. When filling in the model it was always tried to include converging data from different sources. Most of the time this meant both a screenshot with a documentation document. An example of the results of this step can be found in Table 5, an excerpt of the app store classification model for the Google Play app store. The completed models for each case study can be found in the case study reports in appendices C1 to C6 respectively.

Feature		**Notes**	**Doc. Ref.**
affiliate stores	no		
app discovery			
cross selling	yes	App listing contains section with more apps from developer, apps other users also installed and apps other users also viewed.	14
lists	yes		11, 13, 14, 30
recommendations	no		
search	yes		13
app listing	yes	Apps are listed in the app store with some basic information including: description, screenshots, videos, version, last updated date, content rating, size, statistics, price, developer, user rating and release notes	14
app store filters			
content rating	yes	The content of apps are rated by the developer in 4 categories: Everyone, Low maturity, Medium maturity, High maturity. Users can report wrongly rated apps after which owner change app rating. Users can filter out apps with a certain content rating.	14, 25, 31, 32, 33, 34
device compatibility	yes	A list of hardware devices used is stored with the account, apps can have a list of compatible devices.	13, 14, 19, 28
.	.	.	.
.	.	.	.
.	.	.	.

Table 5 – Small excerpt from filled in app store classification model for Google Play

5.5.3. Observations

After the data for each of the case studies was available four observations could be made. First of all it became clear that there was a set of common features shared by all app stores. This is not really surprising since all app stores were selected using the same definition and set of criteria. The fifteen core features are: app categories, app listing, app lists, developer app management, developer transaction list, distribution integration, ,featured apps, free revenue model, paid revenue model, pay out methods, payment methods, platform compatibility filter, ratings, reviews and search. The fifteen core features can be mapped to (parts of) the definition of an app store used in this research:

> *An app store is an **online curated marketplace** serving on or more software ecosystems that allows software developers to **sell and distribute** their products to users of a **software platform**.*

Definition	Core feature
Online curated marketplace	App categories, app listing, app lists, featured apps, ratings, reviews, search
Sell and distribute	Developer app management, developer transaction list, distribution integration, free revenue model, paid revenue model, pay out methods, payment methods,
Software platform	Platform compatibility filter

Table 6 - Mapping core features to definition

The second observation is that the case of Binpress was a bit of an outlier compared to the other cases. The difference between Binpress and the other cases can be explained by the fact the Binpress is the only code marketplace of the six. Eleven features or policies have been marked to be not applicable to the case of Binpress, ten of which directly have to do with the fact that it sells source rather than runtime apps. For example it is understandably hard for Binpress to have a remote application removal feature. Besides the 10 elements that did not apply to source code app stores the data shows four features that are unique to Binpress: app suggestions, feature suggestions, issue tracking and a social share revenue model. Two other features app support forums and the ability for users to create a profile are only shared with one other case. These features rather fit the developer-to-developer nature of a source code marketplace.

A third observation that could be made was that some features and policies are mutually exclusive and others are dependent on each other. For example the review posting policies “only after purchase” and “only verified users” cannot exist at the same time. Another example of features that are likely mutually exclusive are developer refunds versus app store refunds. In case developers can issue refunds themselves the app store owner will redirect refund requests to the app developer rather than handling them by themselves. Features that are dependent on each other are for example the subscription revenue model and the subscriptions list. The latter would make no sense without the former.

A fourth observation that was made during the data collection was that app store owners that are not owners of the ecosystem of software platform often try to replicate parts of the platform they don’t have direct access to. For example both SlideMe and the Amazon app store for Android have created their own copy of a licensing API, even though there is one already available by the platform owner Google.

5.5.4. Restructuring and categorization

Now that the data for all case studies is available the data can be used to restructure the model into a more useful form than the current alphabetic list. First of all the fifteen common features were labelled as core features and places separately in the model. For each of the remaining 44 features and 23 policies the main influenced app store characteristic was identified. These app store characteristics were then used to group the features and policies. The reason behind using the influenced app store characteristics as the main differentiator is based on conceptual model used in this thesis. The conceptual model assumes that app store owners use features and policies in order to influence the app store characteristics. With that it can be assumed that an app store owner is more interested in the effect of a certain feature or policy on the app store characteristics than the actual feature itself. It therefore makes sense to differentiate the features and policies into groups based on the app store characteristics that is mainly influenced by it. Of course it is simplistic to assume that each feature only influences one app store characteristic. However since the purpose is to categorize the features and policies and they only need one category it will suffice to use the main influenced characteristic.

After the main influenced app store characteristic was identified for each feature and policy by the researcher, a list of app store characteristics could be derived. The characteristics are split up in user focused characteristics and developer focused characteristics (Table 7). Of course this list of app store characteristics is far from complete. First of all since it only lists characteristics that can directly be influenced by implementing a certain feature or policy and secondly because only the main influenced characteristics were used. It should also be noted that the relation between the existence of a feature or policy and an app store characteristic is not necessarily always positive. For example if an app store has a competing functionality curation policy this negatively influences the openness of the app store to a developer. Using the aforementioned categorization the resulting final app store classification model can be found in Figure 27.

User focused	
app store usability	The usability of the app store to the end user
app findability	The ability for a user to find apps that suit their needs
app quality	The quality level of the apps in the app store
developer quality	The quality level of the developers offering apps in the app store
Developer focused	
app store usability	The usability of the app store to a developer
visibility	The visibility of the apps made by the developer to its potential users
monetization potential	The potentially ability for a developer to monetize their apps in the app store
user interaction	The potential for interaction with and feedback from users
openness	The openness of the types of apps and technologies allowed in the app store

Table 7 - App store characteristics influenced by app store features and policies

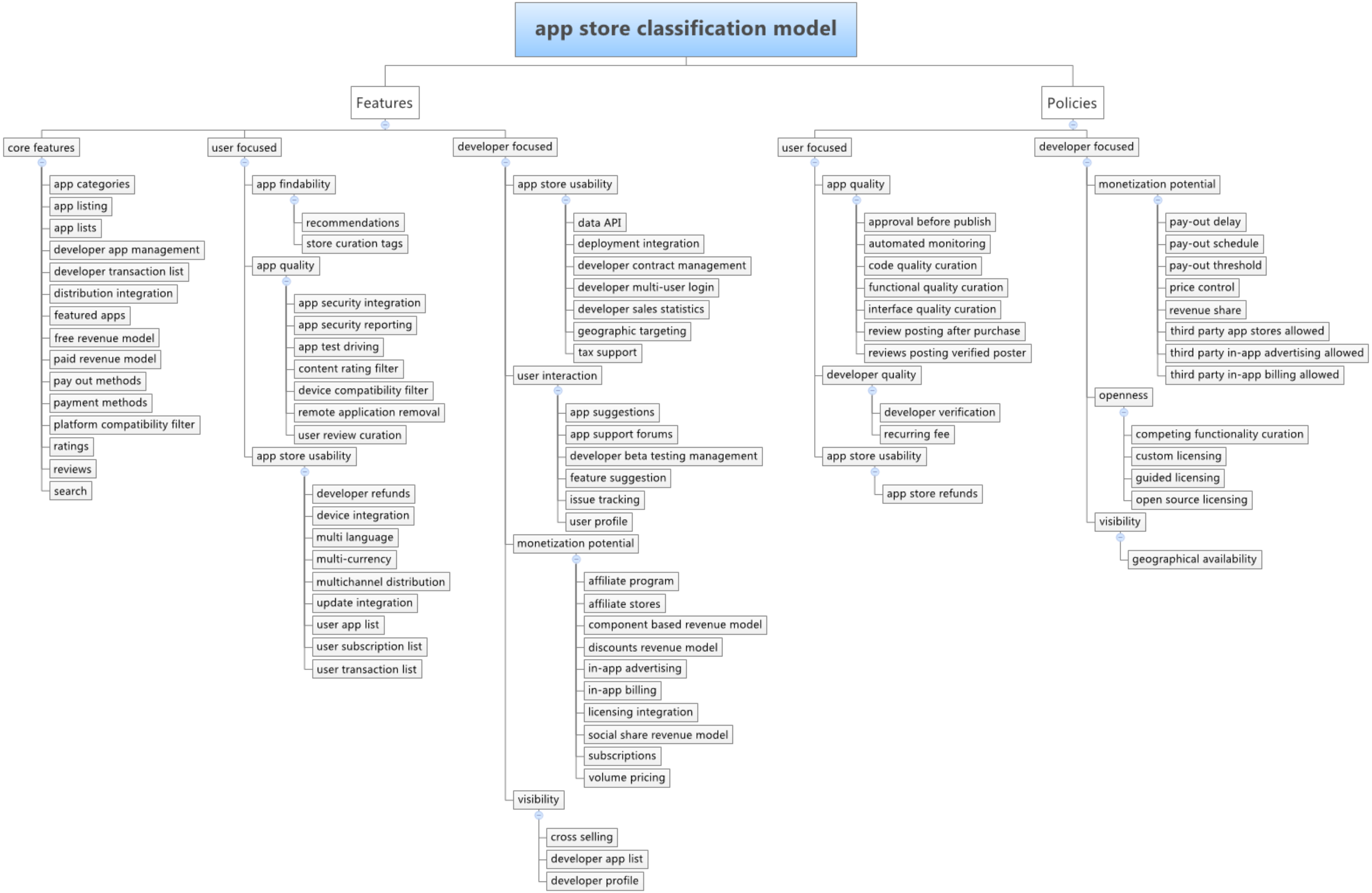


Figure 27 - Final app store classification model

5.5.5. Case comparison

Now that the model has been completed and all data is available a comparison table is made. An excerpt of the full table can be found in Table 8, the full table itself can be found in Appendix B – App store comparison table.

	Intel AppUp	Binpress	Amazon appstore	Apple appstore	SlideMe	Google play
.	.	.	.	.	.	.
.	.	.	.	.	.	.
User focused features						
app findability						
recommendations	no	no	yes	yes	no	no
store curation tags	partial	yes	partial	no	yes	no
app quality						
app security integration	no	na	yes	na	yes	yes
app security reporting	yes	na	yes	no	yes	yes
app test driving	no	no	yes	no	no	no
content rating filter	yes	no	yes	yes	yes	yes
device compatibility filter	partial	na	partial	partial	partial	yes
remote application removal	no	na	no	yes	no	yes
user review curation	no	yes	yes	yes	no	yes
app store usability						
automated refunds	no	no	no	no	no	yes
developer refunds	no	no	no	no	no	yes
device integration	partial	na	partial	yes	partial	yes
multi language	yes	no	yes	yes	yes	yes
multichannel distribution	no	no	yes	yes	no	yes
multi-currency	yes	no	yes	yes	no	yes
update integration	yes	no	yes	yes	yes	yes
user app list	yes	yes	yes	yes	no	yes
user subscription list	na	na	yes	no	na	yes
user transaction list	yes	yes	yes	no	yes	Yes
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Table 8 - Excerpt from the app store comparison table

6. Relation between strategy and features and policies

In order to explore the relation between the app store strategy factors and features and policies a mapping table was made by the researcher. The data in the table is based on a common sense approach to the question “why would an app store owner implement feature / policy X”. The full table can be found in Appendix E – Mapping features and policies to strategy factors. In Table 9 the policy part of the table is shown. All core features were left out of the model because they are implemented by definition into an app store. For almost all other features the answer to the question why it was implemented would be along the lines of “it increases app store characteristic X which would be beneficial to user / developers”. The feature “multi-language” would for example increase the usability of the app store to end users, and therefore it would map to the user focused strategy factor. However on the policies part of the table both strategy factors innovation and control were mapped more frequently, therefore offering a more interesting perspective. A plus sign in the mapping table represents a positive relation between the policy and the factor. The other way around a minus sign represents a negative relation. So for example a higher payout delay would correspond with a lower developer focus.

	User focused	Developer focused	Innovation	Control	Growth	Monetization
User focused policies						
app quality						
approval before publish	+			+		
automated monitoring	+		+			
code quality curation	+			+		
functional quality curation	+			+		
interface quality curation	+			+		
review posting after purchase	+			+		
reviews posting verified poster	+			+		
developer quality						
developer verification	+			+		
recurring fee	+					
app store usability						
app store refunds	+					
Developer focused policies						
monetization potential						
pay-out delay		-				
pay-out schedule		-				
pay-out threshold		-				
price control		+				
revenue share		+				-
third party app stores allowed		+	+			
third party in-app advertising allowed		+	+			
third party in-app billing allowed		+	+			
openness						
competing functionality curation		+		+		
custom licensing		+	+			
guided licensing		+		+		
open source licensing		+	+			
visibility						
geographical availability		+				

Table 9- Table mapping policies to app store strategy factors

7. Discussion

In this chapter reflections are made and the limitations on the results from the previous chapters are discussed.

The first part of this research tries to identify what an app store strategy is an how it can be identified. In chapter 4 the results of a literature study are listed and an app store strategy model is proposed as a way to describe the app store strategy of an app store owner. Each factor of the model is derived by scoping down to the app store level from existing strategic concepts in scientific literature relevant to the domain of software ecosystems, software platforms and the economics of two-sided markets. This research could however do a better job validating the model. For example it is unclear whether the four factors of the model are an accurate and complete way of describing an app store strategy. Another question that remains unanswered is to what extent the model can be predictive of other concepts, especially since the mapping in chapter 6 practically uses only two factors of the available four. One could optimistically assume that the predictive power of the model probably extends beyond features and policies but that cannot be sufficiently underpinned with this research.

The model would also be improved if it had an assessment method with detailed measures that could be used to identify the position on the axis for each factor. However, determining the strategy of an app store owner will always be challenging without directly interviewing the owners. One viable way of doing this might be by compiling publicly available statements made by the owners about their app store strategy from interviews or press releases. The larger app store owners are understandably quite secretive with their strategies and from the smaller app stores there are often few public statements. Using other sources such as blog posts that discuss the strategy of a certain app store could also be possible, however many of these posts base their strategy assessment on the features and policies that they observed. Since this research tries to use the strategy as a way to advice which features and policies should be implemented this such a strategy assessment would not be useful.

7.1. Validity

The multiple case study research that aimed to define and identify app store features and policies resulted in the app store classification model. In order to improve the validity of this model several tactics were used as suggested by Yin (Yin, 2009). First of all the construct validity of the research was improved by, whenever available, using multiple sources of evidence in a convergent manner. In practice this meant that the existence of a certain feature was often underpinned by both direct observations (screenshots) and documentation provided by the app store owner or a third party. Also a chain of evidence was established for each feature and policy in each case study. Using references to the document database each observation can be followed back to its underpinning evidence.

The external validity of the app store classification model was improved by the use of replication logic. The created model is based on six consecutive case studies and was able to describe all observed features and policies. Also the statistics in Table 3 show that the model did not need many changes between the fifth and the sixth case suggesting that it reaches completeness. Of course performing more case studies until the model would not be refined anymore would always improve the external validity of the classification model. One possible problem with the external validity might be the amount of features and policies that were not applicable to the source code app store. In order to improve the external validity of the model for source code app store it would be useful to add more case studies on source code marketplaces.

The reliability of the case study results was improved by creating a case study protocol. This case study protocol ensured that the used field procedures stayed the same between the different cases. The reliability was further improved by creating a document database for each case study. When data was gathered it was consistently added to the document database before any further analysis. The document database allows the analysis of the research to be repeated based on the same data. This proved quite valuable when the cases had to be revisited with the completed model. In many cases no additional data had to be gathered even though the model had significantly grown.

7.2. Further validation

The third research question tried to establish a relation between the app store strategy and the features and policies. This resulted in a table that mapped the features and policies to the app store strategy factors. The table is based on the judgment and common sense of the researcher which might be the first of its weaknesses. The relations as proposed by the mapping table should be further validated. This could for example be done by comparing the mappings with existing cases for which the strategy and a filled in app store classification model have been determined.

Another potential problem with the third research question might be that the initial assumptions made in the conceptual model are not entirely correct. The conceptual model assumes that app store owners first identify their app store strategy and then translate this strategy to features and policies in order to ultimately influence app store characteristics. The flow through this model follows the timing in which the concepts are actually implemented, however it is very well possible that the decision model used by the owners has a different direction. App store owners could just as well translate their app store strategy into a set of app store characteristics they would like to improve (Figure 28). This set of characteristics can then be mapped to the actual features and policies they choose to implement. The fact that such a different direction is also thinkable indicates that there might be a three way relation between the app store strategy, the app store characteristics and the actual features and policies.

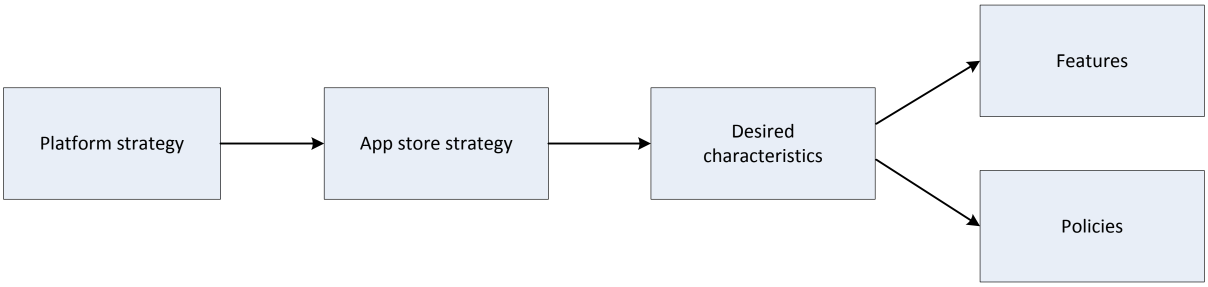


Figure 28 - Alternative conceptual model

The mapping table also shows that, especially for the feature part, the mapping mostly consists of choosing the subsidized side, either developer or user. In practice however the implemented features may not necessarily play a big role in the subsidizing of a side. For example Intel AppUp subsidizes their developer side using marketing tools such as contests with cash prices for developers. Whether or not a certain set of features and policies is implemented might not be as dependent on the app store strategy as was assumed.

After the strategic factors for each feature and policy have been mapped it is particularly tempting to try and reverse the order in the conceptual model and try to determine the strategy of the app store cases by looking at their feature and policy data. There are however some problems with this approach. First the absence of a feature does not necessarily imply that it is not there for a strategic reason. There could very well be legal reasons disallowing certain features. Another reason might be that the app store is not yet feature complete. A certain feature might be desired by an app store owner but not yet available because of development costs. Therefore it would be hard to draw conclusions from the mere existence of features. Another problem is that the features and policies almost only dependent on two of the four factors which would mean that a complete strategy cannot be derived.

8. Conclusions and further research

In this chapter concludes this master thesis document and will revisit the research questions. The three research questions will be listed and as far as possible be answered with the results of the research. After that the main research question will be discussed together with the research objective. Finally some recommendation for further research are made.

In order to scope this research one of the first results of this research was the following definition of an app store:

> *An app store is an online curated marketplace serving on or more software ecosystems that allows software developers to sell and distribute their products to users of a software platform.*

The first research question is RQ1: **What are app store strategies and how can they be identified?** In this master thesis literature research has been used to create an app store strategy model. This app store strategy model gives insight in what app store strategies are and how they can be described. The app store strategy model is currently lacking an assessment method that could be used to actually identify a strategy.

The second research question is RQ2: **What are the features and policies of an app store and how can they be identified within app stores?** This question is answered with the creation of the app store classification model. The classification model lists the features and policies that can be found in the app store. For each element of the model a description is available that can be used to identify the feature or policy in an app store.

The third research question is RQ3: **What is the relation between app store features and policies and platform business strategies?** The relations between the features and the policies is answered by a table mapping features and policies to app store strategy factors.

The overall question of this research was the following: How can app store owners translate their app store strategy into app store features and policies? Even though not all research question are fully answered yet this research might already be useful for app store owners. The app store strategy model could be helpful to an app store owner articulating its app store strategy. The app store classification model can give insight into what features and policies exist and the categorization on main influenced app store characteristics can provide some insight in what features and policies to consider if a certain characteristics are desired. The app store characteristics are also the main subject recommended for further research. Further research could try to discover all app store characteristics that are of importance to developers and users. More insight in these app store characteristics might provide more guidance when translating the app store strategy into an app store.

9. References

Apple Inc. v. Amazon.com Inc., 11-1327 (U.S. District Court, Northern District of California 3 18, 2011).

Baldwin, C., & Woodard, J. (2008). The Architecture of Platforms: A Unified View. *Harvard Business School Finance Working Paper No. 09-034*.

Bosch, J. (2009). From software product lines to software ecosystems. *SPLC '09 Proceedings of the 13th International Software Product Line Conference* (pp. 111-119). Pittsburgh: Carnegie Mellon University.

Boucharas, V., Jansen, S., & Brinkkemper, S. (2009). Formalizing Software Ecosystem Modeling. *Proceedings of the 1st international workshop on Open component ecosystems* (pp. 41-50). ACM.

Brinkkemper, S., van Soest, I., & Jansen, S. (2009). Modeling of product software businesses: Investigation into industry product and channel typologies. *Information systems development: challenges in practice, theory, and education, 1*, 307-325.

Cusumano, M. A. (2010). *Staying power: six enduring principles for managing strategy and innovation in an uncertain world (lessons from Microsoft, Apple, Intel, Google, Toyota and more).* Oxford: Oxford University Press.

Dul, J., & Hak, T. (2008). *Case study methodology in business research.* Oxford: Butterworth-Heinemann.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., & Yoo, Y. (2011). *Dynamic Structures of Control and Generativity in Digital Ecosystem Service Innovation: The Cases of the Apple and Google Mobile App Stores.* London: London School of Economics and Political Science.

Economides, N., & Salop, S. C. (1992). Competition and integration among complements and network market structure. *The Journal of Industrial Economics, 40*(1), 105-123.

Eisenmann, T. R. (2006). Managing Networked Businesses: Course Overview for Educators.

Eisenmann, T., Parker, G., & Van Alstyne, M. W. (2006, October). Strategies for Two-Sided Markets. *Harvard Business Review*.

Elmer-DeWitt, P. (2010, 6 28). *iPhone 4 stockouts: 60% of Apple Stores, 100% of the rest.* Retrieved 5 19, 2011, from Fortune: http://tech.fortune.cnn.com/2010/06/28/iphone-4-stockouts-60-of-apple-stores-100-of-the-rest/

Evans, D., & Schmalensee, R. (2007). *Catalyst code: the strategies behind the world's most dynamic companies.* Boston: Harvard Business School Press.

Farrell, J., & Saloner, G. (1985). Standardization, Compatibility, and Innovation. *The RAND Journal of Economics, 16*(1), 70-83.

Gao, L., & Iyer, B. (2006). Analyzing complementarities using software stacks for software industry acquisitions. *Journal of Management Information Systems, 23*(2), 119-147.

Ghazawneh, A., & Henfridsson, O. (2010). Governing Third-Party Development through Platform Boundary Resources. *ICIS 2010 Proceedings.* Paper 48.

Hagel, J., Brown, J., & Davison, L. (2008). Shaping strategy in a world of constant disruption. *Harvard Business Review, 86*(10), 80-89.

Holzer, A., & Ondrus, J. (2011). Mobile application market: A developer's perspective. *Telematics and Informatics, 28*, 22-31.

Iansiti, M., & Levien, R. (2004, March). Strategy as Ecology. *Harvard Business Review, 82*(3), 68-81.

Iansiti, M., & Levien, R. (2004). *The Keystone Advantage: What te New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability.* Boston: Harvard Business School Press.

Jansen, S., & Brinkkemper, S. (2008). Applied multi-case research in a mixed-method research project: Customer configuration updating improvement. *Information Systems Research Methods, Epistemology and Applications*.

Jansen, S., Brinkkemper, S., & Finkelstein, A. (2007). Providing transparency in the business of software: a modeling technique for software supply networks. In *Establishing The Foundation Of Collaborative Networks* (pp. 677-686). Springer.

Jansen, S., Finkelstein, A., & Brinkkemper, S. (2009). A sense of community: A research agenda for software ecosystems. *Proceedings of the International Conference on Software Engineering 2009.*

Katz, M. L., & Shapiro, C. (1985). Network Externalities, Competition, and Compatibility. *The American Economic Review, 75*(3), 424-440.

Kim, H., Kim, I., & Lee, H. (2010). The Success Factors for App Store-Like Platform Businesses from the Perspective of Third-Party Developers: An Empirical Study Based on A Dual Model Framework. *PACIS 2010 Proceedings. Paper 60.*, (pp. 272-283).

Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews.* Keele, Staffs, UK: Joint Technical Report, Department of Computer Science, Keele University.

Liebowitz, S., & Margolis, S. (1998). Network Externalities (Effects). In *The New Palgrave Dictionary of Economics and the Law.* London: MacMillan.

Messerschmitt, D. G., & Szyperski, C. (2005). *Software Ecosystem: Understanding an Indispensable Technology and Industry.* Cambridge: The MIT Press.

Metcalfe, B. (1995). From the Ether: Metcalfe's Law: A network becomes more valuable as it reaches more users. *Infoworld, 17*(40).

Moore, J. F. (1993, May-June). Predators and Prey: A New Ecology of Competition. *Harvard Business Review, 71*, 75-86.

Moore, J. F. (1996). *The Death of Competition: Leadership & Strategy in the Age of Business Ecosystems.* New York: HarperBusiness.

Osterwalder, A. (2004). *The Business Model Ontology - a proposition in a design. Dissertation.* University of Lausanne, Switzerland.

Parker, G., & Van Alstyne, M. (2000). Information complements, substitutes, and strategic product design. *Proceedings of the twenty first international conference on Information systems* (pp. 13-15). Association for Information Systems.

Parker, G., & Van Alstyne, M. (2005). Two-sided network effects: A theory of information product design. *Management Science, 51*(10), 1494-1504.

Parker, G., & Van Alstyne, M. (2010). Innovation, openness & platform control. *Proceedings of the 11th ACM conference on Electronic commerce* (pp. 95-96). ACM.

Rochet, J., & Tirole, J. (2003). Platform competition in two-sided markets. *Journal of the European Economic Association, 1*(4), 990-1029.

Rochet, J., & Tirole, J. (2006). Two-sided markets: a progress report. *The RAND Journal of Economics, 37*(3), 645-667.

Rysman, M. (2009). The economics of two-sided markets. *The Journal of Economic Perspectives, 23*(3), 125-143.

Sundararajan, A. (2007). Local network effects and complex network structure. *Contributions to Theoretical Economics, 7*(1).

Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Platform evolution: coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research, 21*(4), 675-687.

Utterback, J., & Meyer, M. (1993). The product family and the dynamics of core capability. *Sloan Management Review, 34*, 29-47.

van den Berk, I., Jansen, S., & Luinenburg, L. (2010). Software Ecosystems: A Software Ecosystem Strategy Assessment Model. *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (pp. 127-134). ACM.

Warren, T. (2010, 6 28). *Microsoft planning 'Windows Store' App Store for Windows 8.* Retrieved 4 20, 2011, from Neowin.net: http://www.neowin.net/news/microsoft-planning-039windows-store039-app-store-for-windows-8

Weerd, I. v., & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. In M. R. Syed, & S. N. Syed, *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications* (pp. 28-38). Hershey: Idea Group Publishing.

West, J., & Mace, M. (2010). Browsing as the killer app: Explaining the rapid success of Apple's iPhone. *Telecommunications Policy, 34*(5-6), 270-286.

William O. Ferron, J. (2011, 1 11). Opposer Microsoft Corporation's Motion For Summary Judgment. *In the United States Patent and Trademark Office before the Trademark Trial and Appeal Board*. Opposition No. 91195582.

Yin, R. K. (2009). *Case Study Research: Design and Methods* (4 ed.). SAGE Publications, Inc.

Appendix A – Model element descriptions

In the table below each feature of the app store classification model is listed. For each feature or policy a short description is given that is used to determine whether an app store has that certain feature.

Feature name	**Description**
Core features	
app categories	Apps are listed in different categories and subcategories. Users can browse categories.
app listing	The app store lists apps, usually together with name, description and images etc.
app lists	The app store features a number of pre compiled lists, e.g. top selling lists or latest additions.
developer app management	Developers can manage their apps in a developer console.
developer transaction list	Developers have access to a list of all transactions made to their account.
distribution integration	The app store covers the whole distribution of an app, i.e. no additional manual steps are required to download or install the app on a device.
featured apps	Some apps can be featured by the app store owner. They receive a more prominent place in the app store.
free revenue model	Apps can be offered for free.
paid revenue model	Apps can be offered for certain amount of money.
pay out methods	Number of pay out methods
payment methods	Number of payment methods
platform compatibility filter	Apps have information on their platform compatibility. Users can filter apps that are not compatible with the platform they use.
ratings	Apps can be rated.
reviews	Users can leave reviews of an app.
search	Users can search for apps using search keywords.
User focused features	
app findability	
recommendations	The app store recommends apps based on the usage profile of the user.
store curation tags	Developers can associate tags with their apps. Users can search for these tags or filter with these tags.
app quality	
app security integration	The appstore integrates with a platform security system such as app privileges.
app security reporting	Users can report apps to the app store owner that they consider a security risk or otherwise harmful
app test driving	Users can test drive apps before purchasing them.
content rating filter	Apps get a content rating which depict the appropriateness of the apps content, generally divided for children of different age groups. Users set a level of content rating they would like to show.
device compatibility filter	Apps have a list of compatible devices. Users can filter for apps that are compatible with their device. Partial availability would be when apps are filtered on device capabilities.
remote application removal	The owner can remove harmful apps from the end user device without the help or consent of the end user.
user review curation	Users can curate reviews and filter reviews based on this curation. E.g. rate the helpfulness of a review.
app store usability	
automated refunds	Users can apply for an automatic refund within a certain time limit
developer refunds	Developers can initiate a refund for the price of the app to a user
device integration	The app store is by default available on devices used in the ecosystem.
multi language	The appstore supports multiple languages.

multichannel distribution	Users can use multiple channels to acquire apps, e.g. both a native app and a website.
multi-currency	The appstore supports multiple currencies.
update integration	Downloaded or purchased apps can be updated automatically or by using only one click.
user app list	A list of apps downloaded or purchased by a user is available.
user subscription list	A list of all content subscriptions of user is available.
user transaction list	A list of all transactions made by a user is available.

Developer focused features

feedback potential

app suggestions	Users can leave suggestions for apps they would like to see in the appstore. Developers can browse the suggestions.
app support forums	Each app has its own support forum where users can interact with the developer to solve their problems with the app.
developer beta testing management	Developers can invite beta testers for their apps. Beta testers are given access to apps that or not publicly available yet.
feature suggestion	Users can suggest new features to be added to an app. Developers can browse the suggestions.
issue tracking	Users can report issues with an app. These issues are listed and developers can use them to track their progress.
user profile	Users can fill in additional information about themselves in their user profile.

monetization potential

affiliate program	The app store offers an affiliate program where money can be earned for sales resulting from traffic redirected by a user.
affiliate stores	Content of the app store is also distributed in affiliate stores. These affiliate stores can use different branding but should be based on the same app store platform.
component based revenue model	Developers can offer app components on the app store and share in the revenue of apps that use their component.
discounts revenue model	Apps are offered at discounted rates for limited time frames.
in-app advertising	The app store integrates an in-app advertising system that can be used by developers to monetize their apps.
in-app billing	The app store offers in-app billing functionality where users can buy additional content.
licensing integration	A licensing service is offered that allows developers to let the apps check on startup whether they have a valid license bought using the app store.
social share revenue model	Apps can be offered after a user spreads the word on social media or leaves his email address to be used for promotion.
subscriptions	Developers can offer content subscriptions to users.
volume pricing	Developers can offer volume pricing.

app store usability

data API	Developers can get data from the app store using an API.
deployment integration	Developers can automate the deployment of their apps to the app store.
developer contract management	Developers can see and manage the legal contracts they have with the app store owner.
developer multi-user login	Multiple users can be added to a developer account. Optionally users can have different roles and access rights associated with these roles.
developer sales statistics	Developers have access to sales statistics in order to determine trends in their sales.
geographic targeting	Apps can be targeted to geographic regions.
tax support	The app store applies legally required taxation for the developer.

visibility

cross selling	Together with the current app shown a list of apps that also might be of interest to the user is shown in order to increase the sales.
developer app list	A list of apps made by each developer is available.
developer profile	Each developer has a profile page with additional information on the development organization.

Policy name	**Description**
User focused policies	
app quality	
approval before publish	Apps are checked by the owner for compliance to the content guidelines of an app store before they are published. This review is generally done manually.
automated monitoring	The app store uses an automated system to check for apps that do not comply to the guidelines.
code quality curation	The quality of the code of apps is checked by the owner. Often these checks involve proper use of the SDK and a check for security problems. Apps that are below a certain quality level are not approved or removed.
functional quality curation	The functional quality of apps is curated by the owner. Apps that offer not enough value to the end user are not approved or removed.
interface quality curation	The owner checks apps for compliance with interface guidelines. Apps that do not comply are not approved or removed.
review posting after purchase	Reviews for an app can only be posted by users that have downloaded or purchased the app using the app store.
reviews posting verified poster	Users that are verified by the app store owner can post reviews for any app, e.g. users that have completed a valid purchase or users that are member for over a certain time period.
developer quality	
developer verification	Developers have to prove their identity to the app store owner before they can fully use the appstore and receive payments.
recurring fee	A recurring fee is required to be a developer at an app store, usually a yearly amount.
app store usability	
app store refunds	The app store owner has a clear refund policy and provides refunds on request of a user
Developer focused policies	
monetization potential	
pay-out delay	The delay between the payout and the last day of the scheduled date range.
pay-out schedule	The schedule payment schedule of the revenue share of the sales to the developer, usually monthly.
pay-out threshold	The minimum amount required to be eligible for a payout.
price control	The party that can control the price of an app, either the developer or the app store owner.
revenue share	The percentage revenue share that goes to the developer.
third party app stores allowed	Apps are allowed to reference other app stores.
third party in-app advertising allowed	Apps are allowed to use third party in-app advertising.
third party in-app billing allowed	Apps are allowed to use a third party system for in app purchases.
openness	
competing functionality curation	Apps that have features that compete with the business of the app store owner are not approved or removed.
custom licensing	Developers can provide their own custom EULA, not limited by the app store owner.
guided licensing	The app store owner provides and enforces guidelines for EULAs.
open source licensing	Developers can use open source licenses to publish their apps.
visibility	
geographical availability	The number of countries an app store is available in.

Appendix B – App store comparison table

Features

	Intel AppUp	Binpress	Amazon appstore	Apple appstore	SlideMe	Google play
Core features						
app categories	yes	yes	yes	yes	yes	yes
app listing	yes	yes	yes	yes	yes	yes
app lists	yes	yes	yes	yes	yes	yes
developer app management	yes	yes	yes	yes	yes	yes
developer transaction list	yes	yes	yes	yes	yes	yes
distribution integration	yes	partial	yes	yes	yes	yes
featured apps	yes	yes	yes	yes	yes	yes
free revenue model	yes	partial	yes	yes	yes	yes
paid revenue model	yes	yes	yes	yes	yes	yes
pay out methods	1	1	2	1	3	1
payment methods	2	3	2	4	3	2
platform compatibility filter	yes	yes	yes	yes	yes	yes
ratings	yes	yes	yes	yes	yes	yes
reviews	yes	yes	yes	yes	yes	yes
search	yes	yes	yes	yes	yes	yes
User focused features						
app findability						
recommendations	no	no	yes	yes	no	no
store curation tags	partial	yes	partial	no	yes	no
app quality						
app security integration	no	na	yes	na	yes	yes
app security reporting	yes	na	yes	no	yes	yes
app test driving	no	no	yes	no	no	no
content rating filter	yes	no	yes	yes	yes	yes
device compatibility filter	partial	na	partial	partial	partial	yes
remote application removal	no	na	no	yes	no	yes
user review curation	no	yes	yes	yes	no	yes
app store usability						
automated refunds	no	no	no	no	no	yes
developer refunds	no	no	no	no	no	yes
device integration	partial	na	partial	yes	partial	yes
multi language	yes	no	yes	yes	yes	yes
multichannel distribution	no	no	yes	yes	no	yes
multi-currency	yes	no	yes	yes	no	yes
update integration	yes	no	yes	yes	yes	yes
user app list	yes	yes	yes	yes	no	yes
user subscription list	na	na	yes	no	na	yes
user transaction list	yes	yes	yes	no	yes	yes
Developer focused features						

	Intel AppUp	Binpress	Amazon appstore	Apple appstore	SlideMe	Google play
feedback potential						
app suggestions	no	yes	no	no	no	no
app support forums	no	yes	no	no	yes	no
developer beta testing management	yes	no	no	no	no	no
feature suggestion	no	yes	no	no	no	no
issue tracking	no	yes	no	no	no	no
user profile	no	yes	yes	no	no	no
monetization potential						
affiliate program	no	yes	no	yes	no	no
affiliate stores	yes	no	no	no	yes	no
component based revenue model	yes	no	no	no	no	no
discounts revenue model	no	no	yes	no	no	partial
in-app advertising	no	na	no	yes	no	yes
in-app billing	yes	na	yes	yes	no	yes
licensing integration	yes	no	yes	na	yes	yes
social share revenue model	no	yes	no	no	no	no
subscriptions	no	no	yes	yes	no	yes
volume pricing	no	yes	no	yes	no	no
app store usability						
data API	no	yes	yes	yes	no	yes
deployment integration	no	yes	no	no	no	no
developer contract management	yes	no	no	yes	no	no
developer multi-user login	yes	no	yes	yes	no	yes
developer sales statistics	partial	yes	yes	yes	no	yes
geographic targeting	yes	no	yes	yes	yes	yes
tax support	yes	no	yes	yes	yes	partial
visibility						
cross selling	yes	yes	yes	yes	yes	yes
developer app list	no	yes	yes	yes	yes	yes
developer profile	no	yes	partial	no	yes	no

Policies

	Intel AppUp	Binpress	Amazon appstore	Apple appstore	SlideMe	Google play
User focused policies						
app quality						
approval before publish	yes	yes	yes	yes	yes	partial
automated monitoring	no	no	no	no	no	yes
code quality curation	yes	yes	yes	yes	no	yes
functional quality curation	no	yes	no	yes	yes	no
interface quality curation	yes	na	no	yes	no	no
review posting after purchase	yes	yes	no	yes	yes	yes
reviews posting verified poster	no	no	yes	no	no	no
developer quality						

	Intel AppUp	Binpress	Amazon appstore	Apple appstore	SlideMe	Google play
developer verification	no	no	no	yes	no	partial
recurring fee	partial	no	partial	yes	no	yes
app store usability						
app store refunds	partial	yes	partial	partial	yes	No
Developer focused policies						
monetization potential						
pay-out delay	0	14	30	45	60	1
pay-out schedule	monthly	monthly	monthly	monthly	monthly	monthly
pay-out threshold	0	$50	$10 / $100	$150	$100	$1
price control	developer	developer	owner	developer	developer	developer
revenue share	70%	70%	70% - 20%	70%	98%-80%	70%
third party app stores allowed	no	yes	no	no	partial	no
third party in-app advertising allowed	yes	na	yes	no	yes	yes
third party in-app billing allowed	no	na	no	no	yes	no
openness						
competing functionality curation	no	no	no	yes	no	no
custom licensing	no	no	no	no	yes	no
guided licensing	yes	yes	yes	yes	yes	yes
open source licensing	yes	yes	no	no	yes	yes
visibility						
geographical availability	47	Worldwide	1 (6)	155	Worldwide	129

Appendix C1 - Case Study Report: Google Play

Data collection

The case study protocol prescribes the creation of a case study database containing all the data of the case study. In order to fill this database data for this case study was gathered in the following three ways. First of all relevant literature was searched using Google Scholar, which unfortunately resulted in only a few usable papers.

Secondly direct observations were made using a HTC Legend phone and the Google Play website on http://play.google.com. Both the user account and the developer account were in the name of the researcher. The observations were recorded using screenshots of all steps taken.

Finally documentation data was collected from the documentation provided by the owner of the app store, in this case Google. Notable sources of this data are the Google Play for developer help (http://support.google.com/googleplay/android-developer) and the official Android developers blog (http://android-developers.blogspot.com/). Additional documentation was found by searching for many other websites and blogs written by android users or android developers.

All the data was collected between December 2011 and May 2012. An overview of the case study database can be found in Table 10.

Refined model

After studying the available data about Google Play the app store classification model has been refined using the initial model as a guide. As could be expected this first reality check meant it had to be changed at several points. The results can be seen in Figure 29.

The most important difference with the initial model is the removal of two categories: characteristics and ecosystem. The reason for removing these two categories from the classification model have to do with the scope of this research. The initial model was created by aggregating app store comparisons found on the internet, most of which had a developer perspective on app stores. This research however takes the app store owner perspective and is concerned with the variables an owner could directly influence. For example the number of end users is of great importance for a developer to decide whether to join an app store but it is something an owner cannot influence directly. The ecosystem variables have similar problems in context of this research. The health of an ecosystem could be of great importance to a developer but since the owner of an app store is not necessarily the owner of an ecosystem we cannot classify an app store based on its ecosystem.

The second most important refinement is that the features category has doubled in size. Many existing variables were renamed or moved to a different subcategory. For example app store filters, developer console and store curation features were added. The policies category stayed almost the same, only the review posting policy was added.

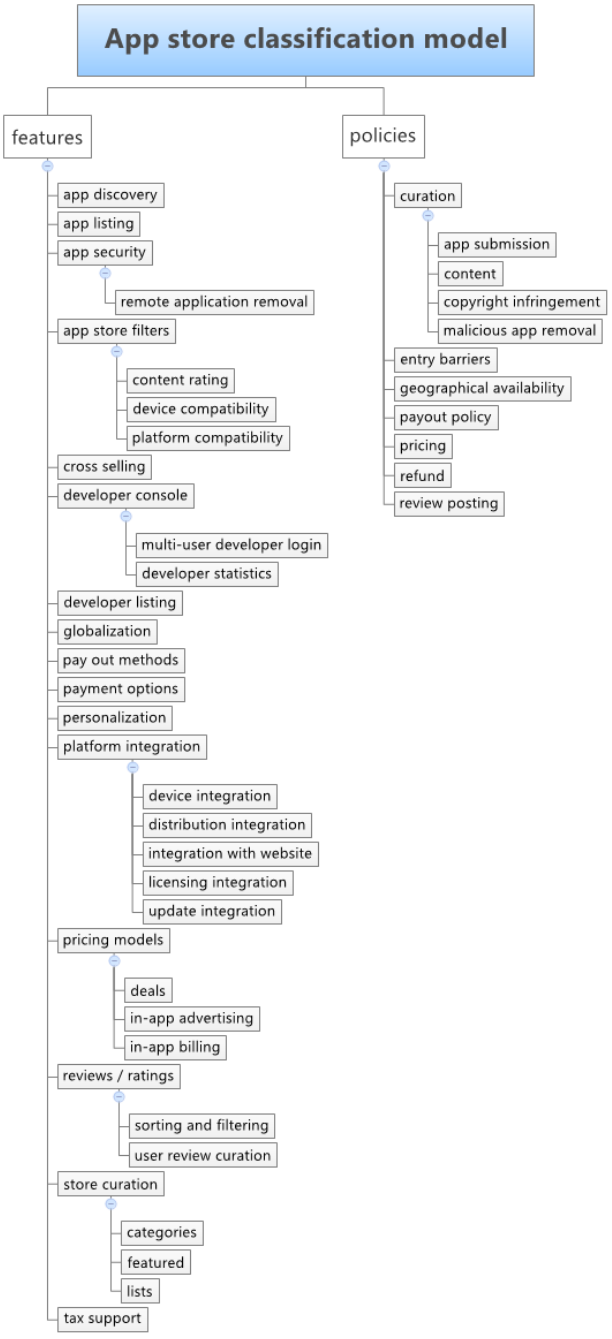


Figure 29 – Google Play: Refined app store classification model

Revisited model

After the app store classification model has been made final this case study had to be updated to reflect those changes. The final list of features and policies can be found below.

Features

Features

Name			
affiliate stores	no		
app discovery			
cross selling	yes	App listing contains section with more apps from developer, apps other users also installed and apps other users also viewed.	14
lists	yes		11, 13, 14, 30
recommendations	no		
search	yes		13
app listing	yes	Apps are listed in the app store with some basic information including: description, screenshots, videos, version, last updated date, content rating, size, statistics, price, developer, user rating and release notes	14
app store filters			
content rating	yes	The content of apps are rated by the developer in 4 categories: Everyone, Low maturity, Medium maturity, High maturity. Users can report wrongly rated apps after which owner change app rating. Users can filter out apps with a certain content rating.	14, 25, 31, 32, 33, 34
device compatibility	yes	A list of hardware devices used is stored with the account, apps can have a list of compatible devices.	13, 14, 19, 28
geographic targeting	yes		45
platform compatibility	yes	The compatibility with platform versions is listed with the app.	14
app suggestions	no		
app support forums	no		
app test driving	no		
data API	yes	Google Play Android Developer API	77
developer			
app management	yes		27
beta testing	no		
contract management	no		
multi-user developer login	yes	Developers can add multiple logins to the same account, using one of them as account owner. Access to financial data can be restricted.	73
sales statistics	yes	Many installation metrics, i.e. countries, platform versions, device models. Timeline data available.	53
transaction list	yes		53
developer listing			
app list	yes		20
developer profile	no		
feature suggestion	no		
globalization			
multi-currency	yes		11, 43
multi language	yes		11, 44
issue tracking	no		
pay out methods			

check	no		
credit card	no		
payment service provider	no		
wire transfer	yes	Payouts are processed by Google checkout using bank transfers.	39, 40, 41
payment methods			
app gift codes			
carrier billing	yes		35, 36, 37, 38
credit card	yes	Payment for apps is generally done using a Google wallet account. This account connects with the following credit cards: American Express, Discover, MasterCard, Visa, Visa Electron. Depending on the carrier network of the user carrier billing can also be supported.	35, 36, 37, 38
store credits	no		35, 36, 37, 38
third party payment service	no		35, 36, 37, 38
personalization			
app list	yes		17
subscription list	yes		78
transaction list	yes		18
user profile	no		
platform integration			
deployment integration	no		
device integration	yes	The app store is available through an app running on the hardware device. Google licenses hardware devices that will have the google play application installed.	51
distribution integration	yes	App downloads and installation are handled by the app store application on the device and do not need any further manual steps.	76
licensing integration	yes	Google play offers a licensing validation services that can be integrated into apps using a library.	57
multichannel distribution	yes	Users can push apps from the website to their hardware device.	22,29
update integration	yes	App updates are distributed by the app store application. Updates can be installed manually by the user or automatically on request of the user. Apps with changing security permissions cannot be installed automatically.	74, 75
refunds			
automated refunds	yes	Refunds available for end users within 15 minutes after buying.	60, 63, 80
developer refunds	yes	Developers can offer a refund anytime through the payment gateway. Google has the right to refund up to 48 hours after a purchase.	60, 63, 80
revenue models			
affiliate program	no		
component based	no		
discounts	partial	Only incidental promotions.	50
free	yes		25, 42, 43
in-app advertising	yes	In-app advertising is allowed and even supported and Google does not impose any restrictions on the choice of mobile advertising network.	44
in-app billing	yes	In-app billing is available and supported as a Google Play service.	55
paid	yes	The price of paid apps ranges depending on the used currency, for example from €0.50 to €100. Pricing can differ between countries and currencies.	25, 42, 43
social share	no		
subscriptions	yes		55, 78

volume pricing	no		
reviews / ratings			
ratings	yes		24, 46
reviews	yes		24, 47
user review curation	yes	Users can indicate the usefulness of a review and report it for spam.	24
security			
app reporting	yes		
app security	yes	Android apps use a permission based security system and request a series of permission levels they will use. Users can review permissions before installing an application.	4, 5, 16, 21
remote application removal	yes	App store owner can remotely remove apps from devices of end users.	67
store curation			
categories	yes	Apps are categorized in two sections: apps and games. Each section contains a number of categories, respectively	12
featured	yes	Apps that are featured will be show up in the staff picks lists and on three different levels in the app store application: the home screen, the category page, the subcategory page. There are no paid promotional spots or advertisements.	1, 52
tags	no		
tax support	partial	Depending on developer location.	40, 56

Policies

Policies			
curation			
content			
code quality	yes		58, 59, 68
competing functionality	no		
functional quality	no		
interface quality	no		
processes			
approval before publish	partial	No mandatory initial review except for the general security review by "bouncer".	58, 59
automated monitoring	yes	"bouncer" tool	58, 59
entry barriers			
developer verification	partial	To receive payments a bank account should be verified.	65
fees			
initial fee	no		
recurring fee	yes	Google charges a $25 registration fee for developers.	65
geographical availability	129	Paid apps are available to end users in 129 countries. Apps can be sold by merchants from 29 countries.	61, 64
licensing models			
custom	no		60
guided	yes		60
open source	yes		60, 79
pay-out			
delay	1		66
schedule	monthly		66
threshold	$1		66
price control	developer		66

app store refunds	No	Google has the right to refund up to 48 hours after a purchase.	60, 63, 80
revenue share	70%	Google charges a transaction fee of 30% for each transaction.	62
review posting			
after purchase	yes	Users that have installed an app can leave a rating and comment on an application.	24, 48, 49
verified poster	no		
third party integration			
app stores	no		69
in-app advertising	yes		69
in-app billing	no		69

Document database

ID	Type	Document title	Year
1	DocW	Featured Levels on Android Market	2012
2	Doc	About Google Play	2012
3	Doc	Play Anywhere	2012
4	Lit	A Study of Android Application Security	2012
5	Lit	Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets	2012
6	Lit	Google Android: An Emerging Software Platform For Mobile Devices	2010
7	Lit	Android: Changing the Mobile Landscape	2011
8	Lit	Android vs Windows Mobile vs Java ME: A comparative Study of Mobile Development Environments	2010
9	DocW	Quality Of Android Market Apps Is 'Pathetically Low': Developer	2011
10	DocW	Android is a mess, say developers	2011
11	Obs	Screenshot1	2011
12	Obs	Screenshot2	2011
13	Obs	Screenshot3	2011
14	Obs	Screenshot4	2011
15	Obs	Screenshot5	2011
16	Obs	Screenshot6	2011
17	Obs	Screenshot7	2011
18	Obs	Screenshot8	2011
19	Obs	Screenshot9	2011
20	Obs	Screenshot10	2011
21	Obs	Screenshot11	2011
22	Obs	Screenshot12	2011
23	Obs	Screenshot13	2011
24	Obs	Screenshot14	2011
25	Obs	Screenshot15	2011
26	Obs	Screenshot16	2011
27	Obs	Screenshot17	2011
28	DocW	Google adds device compatibility checks to the Android Market	2011
29	DocW	Android Market website will push apps to your phone	2011
30	Doc	New ways to discover great apps on Android Market	2011
31	Doc	Content Rating for Android Market	2010
32	Doc	Rating your application content for Google Play	2012
33	Doc	Android Market Developer Program Policies	2012

ID	Type	Document title	Year
34	Doc	Application Content Ratings	2012
35	Doc	Payment Methods	2012
36	Doc	Carrier Billing	2012
37	Doc	More Payment Options in Android Market	2010
38	DocW	Google Wallet to Become the Payment Method for Android Market	2011
39	Obs	Screenshot 18	2012
40	Obs	Screenshot 19	2012
41	Obs	Screenshot 20	2012
42	Doc	Prices and supported currencies	2012
43	Doc	Selling Apps in Multiple Currencies	2012
44	Doc	Advertising without Compromising User Experience	2012
45	Doc	Filters on Google Play	2012
46	DocW	User Reviews on Android Market Get Advanced Filters	2011
47	Doc	Comment Posting Policy	2012
48	Obs	Screenshot 21	2012
49	Obs	Screenshot 21	2012
50	DocW	[Editorial] Is Google's 10 Billion Apps Promotion Hurting Developers Who Aren't Part Of It? It Sure Looks That Way	2011
51	Doc	Frequently Asked Questions	2012
52	Doc	Featured app lists	2012
53	Doc	New App Stats for Publishers on Android Market	2012
54	DocW	Android Isn’t About Building a Mobile Platform	2011
55	Doc	In-app Billing	2012
56	Doc	Specifying tax rates	2012
57	Doc	Licensing Overview	2012
58	Doc	Publishing on Google Play	2012
59	DocW	Google Begins Security Review Process for Android Apps	2012
60	Doc	Android Market Developer Distribution Agreement	2012
61	Doc	Paid App Availability	2012
62	Doc	Transaction Fees	2012
63	Doc	Returning apps	2012
64	Doc	Supported locations for merchants	2012
65	Doc	Developer Registration	2012
66	Doc	Processing orders and receiving payouts	2012
67	Doc	Exercising Our Remote Application Removal Feature	2012
68	Doc	Report alleged copyright infringement:	2012
69	Doc	Android Market Developer Program Policies	2012
70	Doc	Andy Rubin	2012
71	Doc	10 Billion Android Market Downloads and Counting	2011
72	DocW	Distribution of free vs. paid Android apps	2012
73	Doc	Making the Android Developer Console work for your whole team	2012
74	Doc	Updates to Downloaded Apps	2012
75	Doc	Publishing Updates on Google Play	2012
76	Doc	Downloading	2012
77	Doc	Google Play Android Developer API	2012
78	Doc	Subscriptions on Google Play	2012
79	DocW	Android Market should stimulate Open Source Apps	2010
80	Doc	Returning apps	2012

ID	Type	Document title	Year
81	DocW	Why Google isn’t worried about Android revenue	2012
82	DocW	Google announces Android App Market paid application details, revenue share model	2008

Table 10 – Overview of the case study database for Google Play

Each table row represents a document in the case study database. The documents are given numbers to identify them and are coded to indicate their source: “Lit” stands for scientific literature, “Obs” for direct observation, “Doc” for documentation provided by the owner and “DocW” for documentation retrieved from other places on the web.

Appendix C2 - Case Study Report: SlideMe

Data collection

The case study protocol prescribes the creation of a case study database containing all the data of the case study. In order to fill this database data for this case study was gathered in the following three ways. First of all relevant literature was searched using Google Scholar, which unfortunately did not result in any usable papers.

Secondly direct observations were made using a HTC Legend phone and the SlideMe website on http://slideme.org. Both the user account and the developer account were in the name of the researcher. The observations were recorded using screenshots of all steps taken.

Finally documentation data was collected from the documentation provided by the owner of the app store. Notable sources of this data are the SlideMe frequently asked questions (http://slideme.org/faq) and the official SlideMe Blog (http:// http://slideme.org/blog /). Additional documentation was found by searching for other websites and blogs written by android users or android developers that have used SlideMe. This last category however did not yield a lot of results.

All the data was collected between December 2011 and May 2012. An overview of the case study database can be found in Table 11.

Refined model

After studying the available data and using the initial model as a guide the model has been refined. Since the SlideMe market differs from Google Play the model had to be changed at several points. The results can be seen in Figure 30.

The main changes on the feature side are the addition of several subcategories. Parent categories do not use their own explanation anymore but depend on their sub items, see for example developer console and reviews / ratings. App security changed to a sub of security which was extended with two other elements, namely remote application removal and user report.

On the policy side entry barriers has gotten child elements developer verification and setup fees. A classification for the allowed licensing models was added. The refunds category was spliced into negotiable refunds and automatic refunds. Tax support and third party integration policies were added as well.

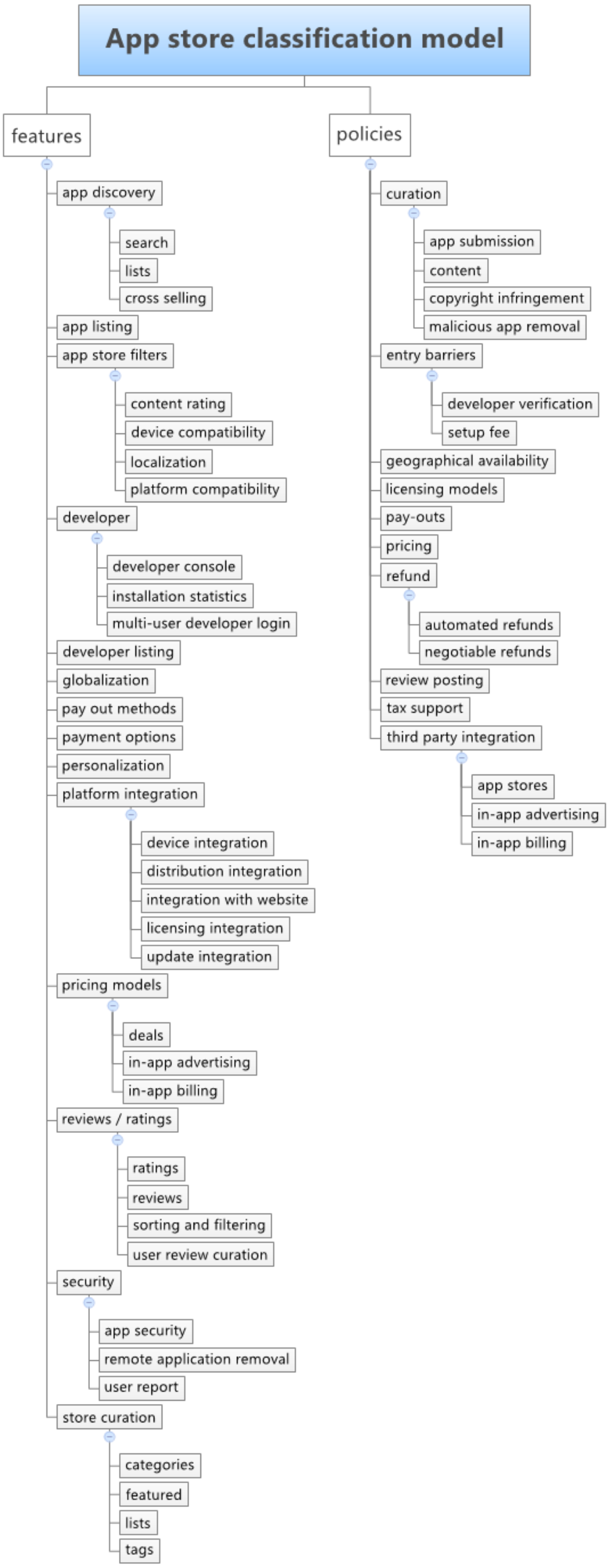


Figure 30 – SlideMe: Refined app store classification model

Revisited model

After the app store classification model has been made final this case study had to be updated to reflect those changes. The final list of features and policies can be found below.

Features

Features

Name	**Available**	**Notes**	**References**
affiliate stores	yes	SlideMe offers custom versions of their native app.	63
app discovery			
cross selling	yes	More apps by developer	21
lists	yes	Lists by curated items: by category, top apps, newest apps, local apps, tags.	7, 8, 9, 22, 23, 24, 62
recommendations	no		
search	yes		27, 28, 41
app listing	yes	Apps are listed in the app store with several attributes: Name, rating, size, version, developer, last update, price, description, screen shots, permissions and compatibility. The website also features parental rating, app languages, download statistics.	10, 11, 12, 13, 47, 49
app store filters			
content rating	yes	Apps have a movie style parental rating with categories ranging from NR (not rated) to NC-17 (Not allowed for 17 and under)	47, 50
device compatibility	partial	SlideMe does not have a compatibility feature on device level. It does however use a set the standard android set of hardware requirements to check for compatibility in addition with some software requirements. Users can also report incompatibilities.	18, 47, 49
geographic targeting	yes	Apps can be targeted to a specific country and language.	32, 47
platform compatibility	yes		15, 49
app suggestions	no		
app support forums	yes		51
app test driving	no		
data API	no		
developer			
app management	yes		45, 46
beta testing	no		
contract management	no		
multi-user developer login	no	No multiuser developer login has been found	
sales statistics	no	No installation statistics have been found	
transaction list	yes		45, 46
developer listing			
app list	yes		16, 64
developer profile	yes		16, 64
feature suggestion	no		
globalization			
multi-currency	no		32, 47, 64
multi language	yes		32, 47, 65
issue tracking	no		
pay out methods			
check	no		
credit card	yes		65, 67

payment service provider	yes	Paypal and Amazon payments	65, 67
wire transfer	yes		65, 67
payment methods			
app gift codes	no		
carrier billing	yes		63, 66, 83
credit card	no		
store credits	yes		44, 66, 81
third party payment service	yes	Amazon payments and paypal	44, 66, 81
personalization			
app list	no		
subscription list	na		
transaction list	yes		45, 46
user profile	no		
platform integration			
deployment integration	no		
device integration	partial	A native app (SAM, Slideme Application Manager) is available but not default installed, however it requires android to be set to accept apps from unknown sources. In some cases the app is installed by the manufacturer, most of the time instead of Google Play.	1, 2, 3, 63
distribution integration	yes	App downloads and installation are handled by the app store application on the device and do not need any further manual steps.	36, 37, 38, 39, 40
licensing integration	yes	SlideMe integrates with their own Slidelock licensing API.	68, 69
multichannel distribution	no		
update integration	yes	The SAM application notifies if updates are available. No automatic updates.	25, 63
refunds			
automated refunds	no		80
developer refunds	no		80
revenue models			
affiliate program	no		
component based	no		
discounts	no		
free	yes		47
in-app advertising	no		
in-app billing	no		
paid	yes		47
social share	no		
subscriptions	no		
volume pricing	no		
reviews / ratings			
ratings	yes	SlideMe uses a 5 star rating system	13, 49
reviews	yes	SlideMe allows users to write a review of an app.	50, 52, 53
user review curation	no		
security			
app reporting	yes	Users can report apps that may contain a security threat.	17, 49
app security	yes	Android apps use a permission based security system and request a series of permission levels they will use. Users can review permissions before installing an application.	14, 19, 36, 39
remote application removal	no		63
store curation			

categories	yes	The mobile client features 20 categories.	7, 47
featured	yes	Featured apps are available in both the app and the website.	7, 56
tags	yes	SlideMe has implemented a tag system which can be used to describe an app. Users can search for application with a certain tag.	62
tax support	yes	SlideMe supports taxation by deducting value added tax before payout.	74, 76

Policies

Policies			
curation			
content			
code quality	no		
competing functionality	no		
functional quality	yes	SlideMe does not allow: Adult, Junk, Gambling. Also no malicious content (virus, worms) and to illegal content.	47, 76
interface quality	no		
processes			
approval before publish	yes	New and significantly changed apps are placed in an submission queue and have to be manually approved by SlideMe. SlideMe checks for Application permissions, Application icon , Application description, Developer information, Types of application and Duplicate applications. Approval usually takes 1 day.	47, 72, 73
automated monitoring	no		
entry barriers			
developer verification	no	No verification mechanism additional to the payout methods has been found. For tax purposes a w8-BEN for has to be submitted by all developers outside of the USA.	47
fees			71
initial fee	no		
recurring fee	no		
geographical availability	Worldwide	No country restrictions found, so everywhere available (with the exception of countries that are subject to trade restrictions).	71
licensing models		SlideMe allows all licenses, including commercial propriety and open source.	47
custom	yes		
guided	yes		
open source	yes		
pay-out			
delay	60		76
schedule	monthly		76
threshold	$100		76
price control	developer		47
app store refunds	yes	Refunds should be negotiated with the developer within 30 days of the purchase. Payment processing costs are for the developer.	79, 80, 84
revenue share	98%-80%	Only payment processing fees are charged. Fees depend on payment processor chosen by end user. Fees range from 1.5 % + $0.01 to 20% + $0.10.	81
review posting			
after purchase	yes		52
verified poster	no		

third party integration

app stores	partial	Links to competing app stores are allowed but in addition to a link to the SlideMe market.	47
in-app advertising	yes	SlideMe allows third party in-app advertising and even has its own advertising API in collaboration with Mobfox.	70
in-app billing	yes		71

Case study database

ID	Type	Document title	Year
1	Obs	App Screenshot1	2012
2	Obs	App Screenshot2	2012
3	Obs	App Screenshot3	2012
4	Obs	App Screenshot4	2012
5	Obs	App Screenshot5	2012
6	Obs	App Screenshot6	2012
7	Obs	App Screenshot7	2012
8	Obs	App Screenshot8	2012
9	Obs	App Screenshot9	2012
10	Obs	App Screenshot10	2012
11	Obs	App Screenshot11	2012
12	Obs	App Screenshot12	2012
13	Obs	App Screenshot13	2012
14	Obs	App Screenshot14	2012
15	Obs	App Screenshot15	2012
16	Obs	App Screenshot16	2012
17	Obs	App Screenshot17	2012
18	Obs	App Screenshot18	2012
19	Obs	App Screenshot19	2012
20	Obs	App Screenshot20	2012
21	Obs	App Screenshot21	2012
22	Obs	App Screenshot22	2012
23	Obs	App Screenshot23	2012
24	Obs	App Screenshot24	2012
25	Obs	App Screenshot25	2012
26	Obs	App Screenshot26	2012
27	Obs	App Screenshot27	2012
28	Obs	App Screenshot28	2012
29	Obs	App Screenshot29	2012
30	Obs	App Screenshot30	2012
31	Obs	App Screenshot31	2012
32	Obs	App Screenshot32	2012
33	Obs	App Screenshot33	2012
34	Obs	App Screenshot34	2012
35	Obs	App Screenshot35	2012
36	Obs	App Screenshot36	2012
37	Obs	App Screenshot37	2012
38	Obs	App Screenshot38	2012
39	Obs	App Screenshot39	2012

40	Obs	App Screenshot40	2012
41	Obs	Web Screenshot1	2012
42	Obs	Web Screenshot2	2012
43	Obs	Web Screenshot3	2012
44	Obs	Web Screenshot4	2012
45	Obs	Web Screenshot5	2012
46	Obs	Web Screenshot6	2012
47	Obs	Web Screenshot7	2012
48	Obs	Web Screenshot8	2012
49	Obs	Web Screenshot9	2012
50	Obs	Web Screenshot10	2012
51	Obs	Web Screenshot11	2012
52	Obs	Web Screenshot12	2012
53	Obs	Web Screenshot13	2012
54	Obs	Web Screenshot14	2012
55	Obs	Web Screenshot15	2012
56	Obs	Web Screenshot16	2012
57	Obs	Web Screenshot17	2012
58	Obs	Web Screenshot18	2012
59	Obs	Web Screenshot19	2012
60	Obs	Web Screenshot20	2012
61	Obs	Web Screenshot21	2012
62	Obs	Web Screenshot22	2012
63	Doc	Partners	2012
64	Obs	Web Screenshot23	2012
65	Doc	Payouts	2012
66	Doc	Payment methods	2012
67	Doc	SlideME MasterCard® - The Best Way to Get Paid	2012
68	Doc	Developers Licensing API (Remote Keys)	2012
69	Doc	SlideLock	2012
70	Doc	Introducing: SlideME Ads - The Premium Mobile Ad Network for SlideME Developers	2012
71	Doc	Developers	2012
72	Doc	What is the application approval process?	2012
73	DocW	App approval?	2012
74	Doc	Why are taxes deducted from my Applications' Selling Price?	2012
75	Doc	SlideME Application Development And Reseller Agreement (1)	2012
76	Doc	SlideME Application Development And Reseller Agreement (2)	2012
77	Doc	SlideME Application Development And Reseller Agreement (3)	2012
78	Doc	SlideME Application Development And Reseller Agreement (4)	2012
79	Doc	How do I get a refund for my purchase if the application doesn't work on my device?	2012
80	DocW	What about refunds?	2012
81	Doc	Rate schedule	2012
82	Doc	About SlideMe	2012
83	Doc	How do I receive my payments for my applications ?	2012

Table 11 – Overview of the case study database for SlideMe

Each table row represents a document in the case study database. The documents are given numbers to identify them and are coded to indicate their source: “Lit” stands for scientific literature, “Obs” for direct observation, “Doc” for documentation provided by the owner and “DocW” for documentation retrieved from other places on the web.

Appendix C3 - Case Study Report: Apple app store

Data collection

Data for the Apple app store case study has been gathered in three different ways. At first scientific literature on the app store was searched using Google Scholar. This resulted only in one usable paper. Secondly direct observations of the app store in use were made. Access for this observations was gained by using the iTunes client software on a Windows 7 pc and by using an Apple iPad 1 device. In order to gain access to the developer side a developer account was provided by project sponsor IDYN. Thirdly extensive documentation off the Apple app store was collected from three different sources: 1) by using their extensive support websites developer.apple.com and support.apple.com, 2) by searching tech blogs and news websites and 3) by blogs referred to by general web searches.

All the relevant documents found were put in a case study document database as prescribed by the case study protocol. All the data was collected between December 2011 and June 2012. An overview of the case study database can be found in Table 12.

Refined model

The initial model was refined at many points to better fit the Apple app store. First of all a recommendations feature was added to app discovery features. Installation statistics was renamed to sales reporting and gift codes were added as a new feature. The Integration with website feature was renamed to multi-channel distribution because it is necessarily a website that is used, i.e. Apple uses the iTunes software as alternative app store frontend. New features pricing, subscription and volume pricing were added to pricing models. The feature lists was removed from store curation features because it duplicated lists as app discovery feature. Tax support was moved from policy to feature. Tax support is actually a feature that can be implemented by the app store. How much tax and on which occasion could be a policy but this is not a policy of the app store owner but rather of the government.

Curation policies were split up into content and processes. The content policies describe five types of content policies and the processes describe how they are enforced. Code quality, competition, functional quality, interface quality and objectionable content were added to the content policies. App submission was renamed to app approval and malicious app removal was renamed to automated monitoring. Setup fee policy was split up into a setup fee and a recurring fee and negotiable refunds was split up in app store refunds and developer refunds. Finally pricing policy was renamed to revenue share policy to better follow the terminology that app stores use. The resulting app store classification model can be found in Figure 31.

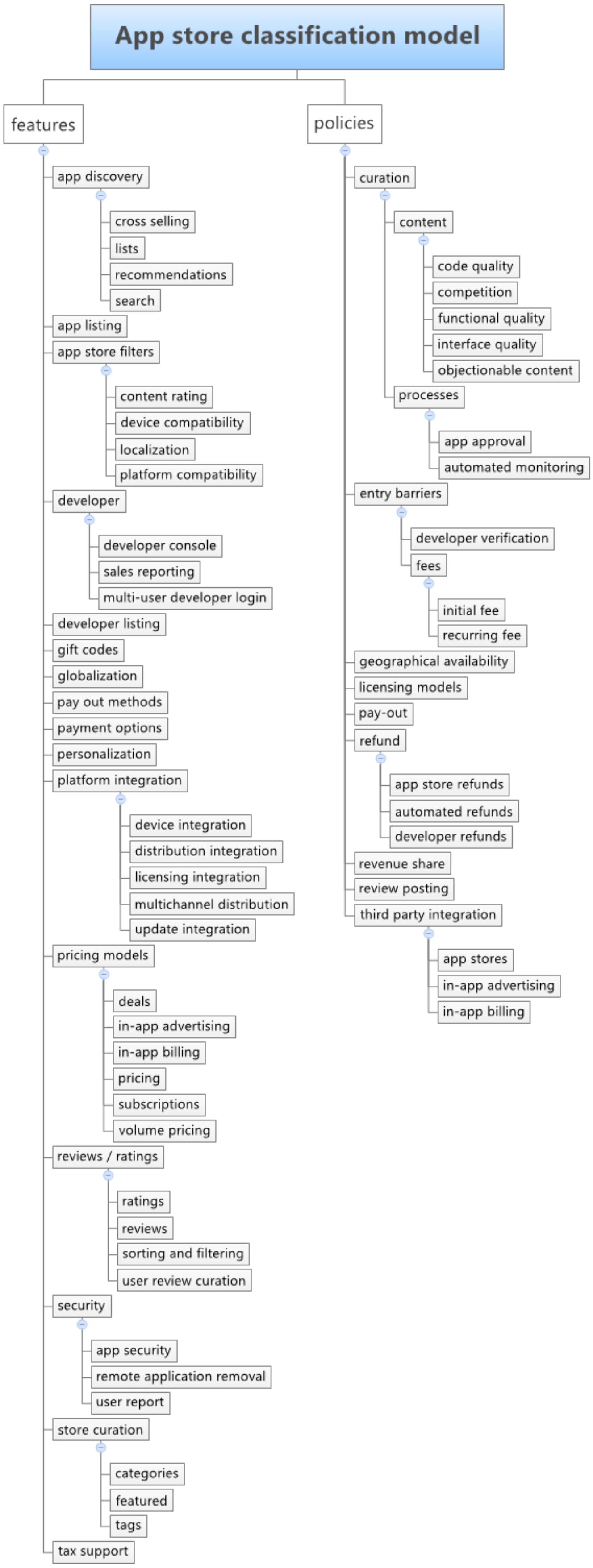


Figure 31 – Apple Appstore: refined app store classification model

Revisited model

After the app store classification model has been made final this case study had to be updated to reflect those changes. The final list of features and policies can be found below.

Features

Features

Name	**Available**	**Notes**	**Reference**
affiliate stores	no		
app discovery			
cross selling	yes	Available cross selling functionality is "more apps from developer" and "other users also bought".	37, 39
lists	yes	Available lists are: Featured, new, What's hot, staff picks, release date. Hit lists are Top paid for device, Top free for device, Highest earning for device.	7, 8 ,9 10, 11, 13, 87
recommendations	yes	The apple app store application has a "Genius for apps" function that gives app recommendations based on the users currently installed apps.	14, 15, 16, 87
search	yes		30, 31, 87
app listing	yes	Apps are listed with the following information: Title, icon, description, version, size, release date, developer, screen shots, category, languages, developer website, app support link, price, what's new.	37, 38, 39
app store filters			
content rating	yes	Content is rated on an a scale based on age: 4+, 9+, 12+, 17+ an no rating, the highest.	37, 59, 72
device compatibility	partial	The app store is differentiated for iPad and iPhone. Compatibility of devices is based on a list of capabilities. Devices that does not have the required capabilities cannot be downloaded.	20, 72
geographic targeting	yes	Developers can select regions where an app will be sold. Regions can have their own price level and EULA.	72
platform compatibility	yes	Apps can configure a minimum targeted iOS version.	37
app suggestions	no		
app support forums	no		
app test driving	no		
data API	yes	a search API for data on apps is available	105
developer			
app management	yes		72
beta testing	no		
contract management	yes		72
multi-user developer login	yes	A role based multi-user developer login system is available.	72
sales statistics	yes	App installation and sales statistics are available.	72, 81
transaction list	yes		72
developer listing			
app list	yes		54
developer profile	no	Only logo	54
feature suggestion	no		
globalization			
multi-currency	yes		72
multi language	yes	App metadata can be provided in multiple languages.	65
issue tracking	no		
pay out methods			

check	no		
credit card	no		
payment service provider	no		
wire transfer	yes		72
payment methods			
app gift codes	yes	Developers can issue promo codes to give away their apps.	55, 69
carrier billing	no		
credit card	yes		73
store credits	yes		73
third party payment service	yes		73
personalization			
app list	yes		80
subscription list	no		
transaction list	no		
user profile	no		
platform integration			
deployment integration	no		
device integration	yes	App store is available as native application on all hardware devices.	28
distribution integration	yes	App downloads and installation are handled by the app store application on the device and do not need any further manual steps.	47, 48, 49, 50
licensing integration	na	No licensing integration has been found, but because of the closed nature of the platform it cannot be expected either.	
multichannel distribution	yes	Apps downloaded using the iTunes application can be automatically downloaded to devices.	77
update integration	yes	Updated apps are listed in the app store application and can be updated with one click.	27, 28, 29
refunds			
automated refunds	no		84, 85, 86
developer refunds	no		84, 85, 86
revenue models			
affiliate program	yes		104
component based	no		
discounts	no		
free	yes		72, 79
in-app advertising	yes	Apple offers the iAd program which is integrated in iOS.	66, 68
in-app billing	yes	In app billing is available, on the app listing the top in-app sales are shown.	37, 62, 64
paid	yes	Apps are priced using a tiered model, ranging from $0.99 to $999	72, 79
social share	no		
subscriptions	yes	Apple app store supports subscription based billing for content providers.	63
volume pricing	yes	Apple offers the App Store Volume Purchase Program to businesses and educational institutes in the US which allows developers to set volume licensing prices.	67
reviews / ratings			
ratings	yes	Store uses a 5 star rating system	43
reviews	yes	Users can write app reviews. Reviews are displayed with the app listing.	38, 39, 40, 41, 42, 43, 44, 45, 58, 1
user review curation	yes	Reviews can be labeled for helpfulness in the iTunes app	42, 60, 61

		store application. In the native app users can sort reviews for helpfulness. Users can report reviews.	
security			
app reporting	no		
app security	na	No security system has been found. Apps are protected by using a closed system and the app store curation.	
remote application removal	yes		70, 71
store curation			
categories	yes	The app store offers 21 categories.	21, 22, 23
featured	yes	Apps can be featured and are shown on the frontpage and on the featured lists.	7, 13
tags	no		
tax support	yes	Taxation is fully supported and depending on location.	72

Policies

Policies			
curation			
content			
code quality	yes	Apple checks the apps for code quality, e.g. proper use of interfaces.	75
competing functionality	yes	Apps that compete with apples own products or services are not allowed in the app store, e.g. linking to external mechanisms for purchases or subscriptions to be used in an app is forbidden.	75, 92, 93
functional quality	yes	Apps that duplicate apps already in the App Store may be rejected. Apps that are not very useful, are simply web sites bundled as apps, or do not provide any lasting entertainment value may be rejected	75
interface quality	yes	Apps should comply with the iOS interface guidelines.	75, 76
processes			
approval before publish	yes	Apps are checked on submission for compliance to app review guidelines and other referenced guidelines.	74, 75
automated monitoring	no		
entry barriers			
developer verification	yes	The Legal Entity Information of a developer is validated before contracts can be generated.	72
fees			
initial fee	no		78
recurring fee	yes	A fee of €99 a year applies.	78
geographical availability	155		82, 83
licensing models			
custom	no		72, 90, 91
guided	yes	Developers can use their own EULA, however they must comply to a set of minimum rules that are incompatible with most open source licenses.	72, 90, 92
open source	no		72, 90, 93
pay-out			
delay	45		72, 97
schedule	monthly		72, 98
threshold	$150		72, 99
price control	developer		
app store refunds	partial	Refunds available through support system	84, 85, 86
revenue share	70%	70% for the developer, 30% for Apple.	79, 97
review posting			

after purchase	yes	Users can only leave reviews of downloaded / purchased apps.	45
verified poster	no		
third party integration			
app stores	no		75
in-app advertising	no		89
in-app billing	no		75, 87

Document database

ID	Type	Document title	Year
1	Obs	Screenshot1	2012
2	Obs	Screenshot2	2012
3	Obs	Screenshot3	2012
4	Obs	Screenshot4	2012
5	Obs	Screenshot5	2012
6	Obs	Screenshot6	2012
7	Obs	Screenshot7	2012
8	Obs	Screenshot8	2012
9	Obs	Screenshot9	2012
10	Obs	Screenshot10	2012
11	Obs	Screenshot11	2012
12	Obs	Screenshot12	2012
13	Obs	Screenshot13	2012
14	Obs	Screenshot14	2012
15	Obs	Screenshot15	2012
16	Obs	Screenshot16	2012
17	Obs	Screenshot17	2012
18	Obs	Screenshot18	2012
19	Obs	Screenshot19	2012
20	Obs	Screenshot20	2012
21	Obs	Screenshot21	2012
22	Obs	Screenshot22	2012
23	Obs	Screenshot23	2012
24	Obs	Screenshot24	2012
25	Obs	Screenshot25	2012
26	Obs	Screenshot26	2012
27	Obs	Screenshot27	2012
28	Obs	Screenshot28	2012
29	Obs	Screenshot29	2012
30	Obs	Screenshot30	2012
31	Obs	Screenshot31	2012
32	Obs	Screenshot32	2012
33	Obs	Screenshot33	2012
34	Obs	Screenshot34	2012
35	Obs	Screenshot35	2012
36	Obs	Screenshot36	2012
37	Obs	Screenshot37	2012
38	Obs	Screenshot38	2012

ID	Type	Document title	Year
39	Obs	Screenshot39	2012
40	Obs	Screenshot40	2012
41	Obs	Screenshot41	2012
42	Obs	Screenshot42	2012
43	Obs	Screenshot43	2012
44	Obs	Screenshot44	2012
45	Obs	Screenshot45	2012
46	Obs	Screenshot46	2012
47	Obs	Screenshot47	2012
48	Obs	Screenshot48	2012
49	Obs	Screenshot49	2012
50	Obs	Screenshot50	2012
51	Obs	Screenshot51	2012
52	Obs	Screenshot52	2012
53	Obs	Screenshot53	2012
54	Obs	Screenshot54	2012
55	Obs	Screenshot55	2012
56	Obs	Screenshot56	2012
57	Obs	Screenshot57	2012
58	Obs	Screenshot58	2012
59	DocW	Here's How iPhone App Store Ratings Work. Hint: They Don't.	2012
60	Obs	iTunes Screenshot 1	2012
61	Obs	iTunes Screenshot 2	2012
62	Doc	App Store Quick Reference: Getting Started with In-App Purchase on iOS and OS X Lion	2012
63	Doc	Subscriptions	2012
64	Doc	Selling with in-app purchase	2012
65	Doc	Build apps for the world	2012
66	Doc	iAd network	2012
67	Doc	Volume purchase program	2012
68	Doc	iAd	2012
69	DocW	Redeem Free Promo Code for an iPhone App	2012
70	DocW	Jobs confirms App Store kill switch; 60M apps downloaded	2012
71	DocW	Apple iPhone Kill Switch: Can CIOs Trust Apple?	2012
72	Doc	iTunes Connect Developer Guide (version 7.4, March 29, 2012)	2012
73	Doc	App Store Frequently Asked Questions (FAQ)	2012
74	Doc	App store aproval process	2012
75	Doc	App store review guidelines	2012
76	Doc	iOS Human Interface Guidelines	2012
77	Obs	iTunes Screenshot 3	2012
78	Doc	Which Developer Program is for you?	2012
79	DocW	An Analysis Of Apple’s Adjustment Of International App Store Prices	2011
80	Obs	iTunes Screenshot 4	2012
81	Doc	iTunes Connect Sales and Trends Guide	2012
82	Doc	iTunes Store: Which types of items can I buy in my country?	2012
83	DocW	App Store Launches in 32 New Countries	2012
84	DocW	How to Get a Refund from the App Store	2012
85	Doc	TERMS AND CONDITIONS	2012
86	DocW	Apple needs a better App Store refund method	2012

ID	Type	Document title	Year
87	DocW	iTunes App Store sucks, and finally Apple acknowledges it	2012
88	DocW	Apple gives Flattr micro-payment the thumbs down	2012
89	DocW	Google AdMob Ads iOS Fundamentals	2012
90	Doc	Instructions for Minimum Terms of Developer’s End-User License Agreement	2012
91	DocW	Open source and app stores: Where they mix, where they don't	2012
92	DocW	Apple Prepares To Pull Evi From App Store. Did It Slap-Down Siri?	2012
93	DocW	Apple Pulls Airfoil Speakers Touch From App Store Without Reason	2012
94	DocW	Is Apple Losing Its Grip on App Store Security?	2012
95	Doc	iOS Developer Program License Agreement	2012
96	Lit	Browsing as the killer app: Explaining the rapid success of Apple’s iPhone	2009
97	Doc	Distribute	2012
98	DocW	Mobile statistics	2012
99	Doc	Apple’s App Store Downloads Top 10 Billion	2011
100	Doc	Apple Reinvents the Phone with iPhone	2007
101	Doc	iPhone App Store Downloads Top 10 Million in First Weekend	2008
102	Doc	Apple Launches Subscriptions on the App Store	2011
103	DocW	App Store to feature subscription, in-app purchases	2009
104	Doc	Join the Affiliate Program	2012
105	Doc	Search API	2012
106	DocW	Distimo-Publication-February-2011.pdf	2011

Table 12 – Overview of the case study database for the Apple app store

Each table row represents a document in the case study database. The documents are given numbers to identify them and are coded to indicate their source: “Lit” stands for scientific literature, “Obs” for direct observation, “Doc” for documentation provided by the owner and “DocW” for documentation retrieved from other places on the web.

Appendix C4 - Case Study Report: Binpress

Data collection

Data for the Binpress case study has been gathered in three different ways. At first scientific literature on the app store was searched using Google Scholar. This did not yield any results. Secondly direct observations of the app store in use were made. Access for this observations was gained by creating a developer account and a non-developer account on Binpress. Thirdly documentation off the Binpress app store was collected from three different sources: 1) by using the documentation provided on their website binpress.com, 2) by searching tech blogs and news websites and 3) by blogs referred to by general web searches. Since Binpress is relatively small most data comes from direct observations and their own documentation.

All the relevant documents found were put in a case study document database as prescribed by the case study protocol. All the data was collected between February 2012 and July 2012. An overview of the case study database can be found in Table 13.

Refined model

Refining the app store classification model with a case study from a different domain than mobile operating systems allows for a fresh perspective. As was expected a number of features are added to the model, some features and policies are renamed and a few are removed from the model altogether. To start of the gift codes feature is renamed to app gift codes to distinguish it from store credit. Feature app suggestions is added which allows users to describe their wishes for a new app to developers. The feature data API is added to the model. A data API is a programming interface that can be used to receive data about the apps from the app store. The developer feature group is updated into five features. The initial feature developer console is dropped in favor of app management and contract management. Sales reporting is split up into sales statistics and transaction list resulting in the following five features: app management, contract management, multi-user developer login, sales statistics and transaction list.

Initial feature developer listing is split up to allow for more detail. New features app list and developer profile are added to this category. The feature globalization is split up into two new features: multi-currency and multi-language. Issue tracking is added as a new feature because Binpress implements a system that can track bugs and other issues with components. Payment options features are now split up into three different features: credit card, third party payment service and store credits. Store credit represents a wallet system that is offered by the app store. Personalization is also split up into three new features. User profile, app list and transaction list features are added as sub features.

Platform integration was extended with a deployment integration feature. This feature describes whether the app store integrates with the development platform for app deployment. The name of pricing models is changed to revenue models, since not all revenue models actually have a price. The sub features are also changed: Affiliate program and social share models are added and pricing is split up into free and paid models. Sorting and filtering are removed as separate feature from rating/reviews because it is deemed to specific and on a different level of granularity than the other features. Because they actually have to be implemented as feature automatic refunds and developer refunds are moved from policies to features. Policy competition is renamed to the more descriptive competing functionality and objectionable content policy is removed because objectionable content is obviously unwanted by any app store.

The resulting app store classification model can be found in Figure 32 and Figure 33.

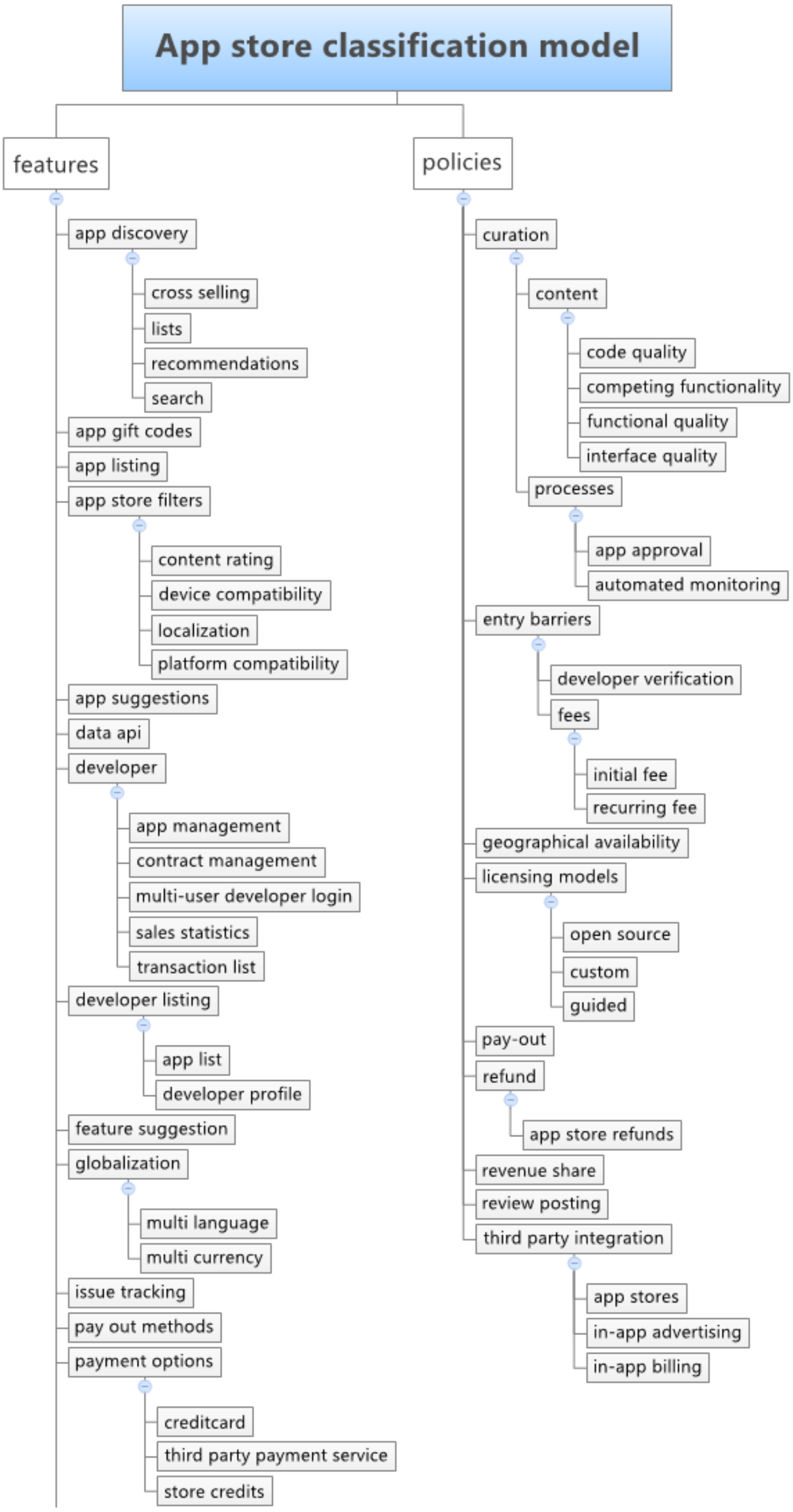


Figure 32 – Binpress: Refined app store classification model (part 1)

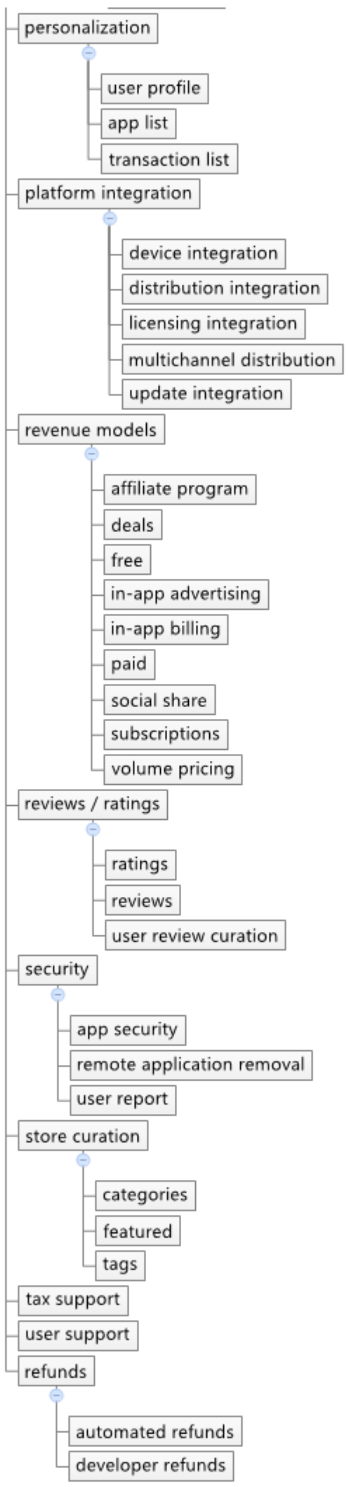


Figure 33 - Binpress: Refined app store classification model (part 2)

Revisited model

After the app store classification model has been made final this case study had to be updated to reflect those changes. The final list of features and policies can be found below.

Features

Features

Name	Available	Notes	Reference
affiliate stores	no		
app discovery			
cross selling	yes	Each component has a list of related components on its listing page. After a download other relevant components are shown.	12, 23
lists	yes	New components, popular components.	3, 8, 9
recommendations	no		
search	yes		47
app listing	yes	Components are listed with their name, description, set of screenshots, demo page, integration time, development hours saved, rating, tags and release date.	12, 19
app store filters			
content rating	no		
device compatibility	na	No device integration	
geographic targeting	no		
platform compatibility	yes	Components can be labeled as compatible with 12 languages with 16 frameworks and 18 platforms.	3. 4. 33
app suggestions	yes	Users can add component suggestions to the app store. Developers can browse the suggestions for inspiration.	7, 27, 40
app support forums	yes	Users can ask support questions on the component listing page.	11
app test driving	no		
data API	yes	Component data is accessible through an API	46, 67
developer			
app management	yes		39
beta testing	no		
contract management	no		
multi-user developer login	no		
sales statistics	yes		45
transaction list	yes		43
developer listing			
app list	yes		17
developer profile	yes	Developers are listed with a name, description, location, member date, social connect, contact information, website, list of proficiencies and recent activities.	17
feature suggestion	yes		14, 16
globalization			
multi-currency	no		
multi language	no		
issue tracking	yes	Users can report and track support issues on the component listing page.	13
pay out methods			
check	no		59
credit card	no		59
payment service provider	yes	Payouts are handled through PayPal.	59

wire transfer	no		59
payment methods			
app gift codes	no		
carrier billing	na	Not a mobile platform	
credit card	yes		18
store credits	yes		21, 28, 58
third party payment service	yes	Payment through PayPal is available.	20
personalization			
app list	yes	A list of all downloaded and purchased component is available.	25, 26
subscription list	na	Subscriptions not available	
transaction list	yes	A list of transactions to the balance of the account is available.	28
user profile	yes	Users can fill a profile with information on themselves to be used in the comments and review sections.	30
platform integration			
deployment integration	yes	Binpress connects with Github to allow code published on Github to be imported in Binpress.	32, 76
device integration	na	Not applicable, source code only.	
distribution integration	partial	Components can only be downloaded.	22, 23
licensing integration	no		
multichannel distribution	no		
update integration	no		
refunds			
automated refunds	no		
developer refunds	no		
revenue models			
affiliate program	yes	Binpress hosts an affiliate program which can pays up to 10% of the sales price.	48, 49, 50, 56
component based	no		
discounts	no		
free	partial	Free licensing is partially available but generally only as part of a volume licensing scheme.	10, 36, 53, 54
in-app advertising	na	Source code only	
in-app billing	na	Source code only	
paid	yes	Pricing levels can be set to any number but should end at .99.	10, 36, 53, 54
social share	yes	Access to component can be given after a user has tweeted or left his email address.	10, 36, 44, 53, 54
subscriptions	no		36, 53, 54
volume pricing	yes	Volume pricing is available by creating multiple license options.	10, 36, 53, 54, 55
reviews / ratings			
ratings	yes	Users can rate components on a five star scale for four categories: Source-code quality, documentation, easy to integrate and overall impression.	12, 24, 52
reviews	yes	Users can leave review text about their overall impression and suggested improvements.	12, 23, 24, 25, 52
user review curation	yes	Users can flag reviews and rate the helpfulness.	12, 52
security			
app reporting	na	Source code only	
app security	na	Source code only	
remote application removal	na	Source code only	
store curation			

categories	yes	Components are categorized in 15 categories.	3, 4, 33
featured	yes	Binpress has featured components on the homepage. Featured components have a label elsewhere on the site.	3, 8
tags	yes		5, 33
tax support	no		

Policies

Policies			
curation			
content			
code quality	yes	Components should adhere to the quality guidelines outlined by Binpress.	61
competing functionality	no		
functional quality	yes	Components are checked for functional quality.	61
interface quality	na		
processes			
approval before publish	yes		61
automated monitoring	no		
entry barriers			
developer verification	no		68
fees			
initial fee	no		68
recurring fee	no		68
geographical availability	Worldwide	No limitations on geographical availability, except for proscribed countries.	60
licensing models			
custom	no	Custom license terms are allowed but only additional to the guided licensing.	70, 71, 72, 73, 74, 75
guided	yes	A comprehensive guided licensing system is used.	70, 71, 72, 73, 74, 75
open source	yes	Several open source licenses are supported.	75, 69
pay-out			
delay	14		59, 60
schedule	monthly		59, 60
threshold	$50		59, 60
price control	developer		60
app store refunds	yes	App store refunds are negotiable, a 14 day money back guarantee for components that do not function as advertised.	21, 57, 60
revenue share	70%		65
review posting			
after purchase	yes	Reviews can only be posted by users that bought or downloaded the component	23
verified poster	no		
third party integration			
app stores	yes	Binpress does not require exclusivity, so other app stores are allowed.	63
in-app advertising	na	Source code only	
in-app billing	na	Source code only	

Document database

ID	Type	Document title	Year
1	Obs	Screenshot1	2012
2	Obs	Screenshot2	2012
3	Obs	Screenshot3	2012
4	Obs	Screenshot4	2012
5	Obs	Screenshot5	2012
6	Obs	Screenshot6	2012
7	Obs	Screenshot7	2012
8	Obs	Screenshot8	2012
9	Obs	Screenshot9	2012
10	Obs	Screenshot10	2012
11	Obs	Screenshot11	2012
12	Obs	Screenshot12	2012
13	Obs	Screenshot13	2012
14	Obs	Screenshot14	2012
15	Obs	Screenshot15	2012
16	Obs	Screenshot16	2012
17	Obs	Screenshot17	2012
18	Obs	Screenshot18	2012
19	Obs	Screenshot19	2012
20	Obs	Screenshot20	2012
21	Obs	Screenshot21	2012
22	Obs	Screenshot22	2012
23	Obs	Screenshot23	2012
24	Obs	Screenshot24	2012
25	Obs	Screenshot25	2012
26	Obs	Screenshot26	2012
27	Obs	Screenshot27	2012
28	Obs	Screenshot28	2012
29	Obs	Screenshot29	2012
30	Obs	Screenshot30	2012
31	Obs	Screenshot31	2012
32	Obs	Screenshot32	2012
33	Obs	Screenshot33	2012
34	Obs	Screenshot34	2012
35	Obs	Screenshot35	2012
36	Obs	Screenshot36	2012
37	Obs	Screenshot37	2012
38	Obs	Screenshot38	2012
39	Obs	Screenshot39	2012
40	Obs	Screenshot40	2012
41	Obs	Screenshot41	2012
42	Obs	Screenshot42	2012
43	Obs	Screenshot43	2012
44	Obs	Screenshot44	2012
45	Obs	Screenshot45	2012
46	Obs	Screenshot46	2012

ID	Type	Document title	Year
47	Obs	Screenshot47	2012
48	Obs	Screenshot48	2012
49	Obs	Screenshot49	2012
50	Obs	Screenshot50	2012
51	Obs	Screenshot51	2012
52	Obs	Screenshot52	2012
53	Obs	Screenshot53	2012
54	Obs	Screenshot54	2012
55	Obs	Screenshot55	2012
56	Doc	Binpress Affiliate Program	2012
57	Doc	Contact us	2012
58	Obs	Screenshot56	2012
59	Doc	Payment terms	2012
60	Doc	Terms of use	2012
61	Doc	Guidelines and coding standards for Binpress components	2012
62	Doc	Licensing	2012
63	Doc	Our commission rate	2012
64	Doc	Welcome	2012
65	Doc	Frequently Asked Questions	2012
66	Doc	About	2012
67	Doc	Advanced usage	2012
68	Doc	Publishing on Binpress	2012
69	Doc	About open-source licenses	2012
70	Obs	Screenshot57	2012
71	Obs	Screenshot58	2012
72	Obs	Screenshot59	2012
73	Obs	Screenshot60	2012
74	Obs	Screenshot61	2012
75	Obs	Screenshot62	2012
76	DocW	Source Code Marketplace Binpress Adds Github Integration	2012
77	DocW	Binpress Is a Marketplace for Buying & Selling Source Code	2011
78	DocW	Crunchbase: Binpress	2012

Table 13 – Overview of the case study database for Binpress

Each table row represents a document in the case study database. The documents are given numbers to identify them and are coded to indicate their source: “Lit” stands for scientific literature, “Obs” for direct observation, “Doc” for documentation provided by the owner and “DocW” for documentation retrieved from other places on the web.

Appendix C5 - Case Study Report: Amazon appstore for Android

Data collection

Three sources of data has been used for this case study in accordance with the case study protocol. The first source of data is as always scientific literature. A Google scholar search however did not show papers relevant to our case study. The second data source used are direct observations. Using a HTC Legend android device and a Firefox web browser the Amazon app store was used. All actions taken have been thoroughly documented by screenshots which were stored in the case study database. In order to look at the developer side of the app store and to get access to developer documentation a developer account was created by the researcher.

The third source of data is documentation either as provided by Amazon or as provide by third parties using or discussing the Amazon app store for Android. Documentation from Amazon could be found at the developer support pages at developer.amazon.com as well as the regular Amazon help pages. Third party documentation was found by searching large blog sites such as Techcrunch, Engadget, TheNextWeb and ArsTechnica and also by performing Google web searches with keywords related to the Amazon app store.

All the relevant documents found were put in a case study document database as prescribed by the case study protocol. All the data was collected between December 2011 and July 2012. An overview of the case study database can be found in Table 14.

Refined model

After the case study on the Amazon appstore for Android the model had to be refined once again. Since the Amazon app store is closely connected to the existing Amazon web shop infrastructure a fresh perspective on the features and policies of an app store was required. First of all the feature app gift codes was moved to payment methods. Localization has been renamed to geographic filter, a name that better fits the measured unit. Feature payout methods has been split up into four sub features: check, credit card, payment service provider and wire transfer. The possible payments options have been extended with carrier billing. Personalization now has an extra feature subscription list an user support is renamed to app support forums. Also a feature app test drive has been added to the model.

On the policy side of the model the payout policy was split up into three: delay, schedule and threshold. These three options are generally sufficient to describe the payout policy of an app store. Because of the way Amazon handles app pricing a new policy price control was added to the model. The review posting policy has been made more detailed by adding two sub policies: after purchase and verified poster. The resulting app store classification model can be found in Figure 34 and Figure 35.

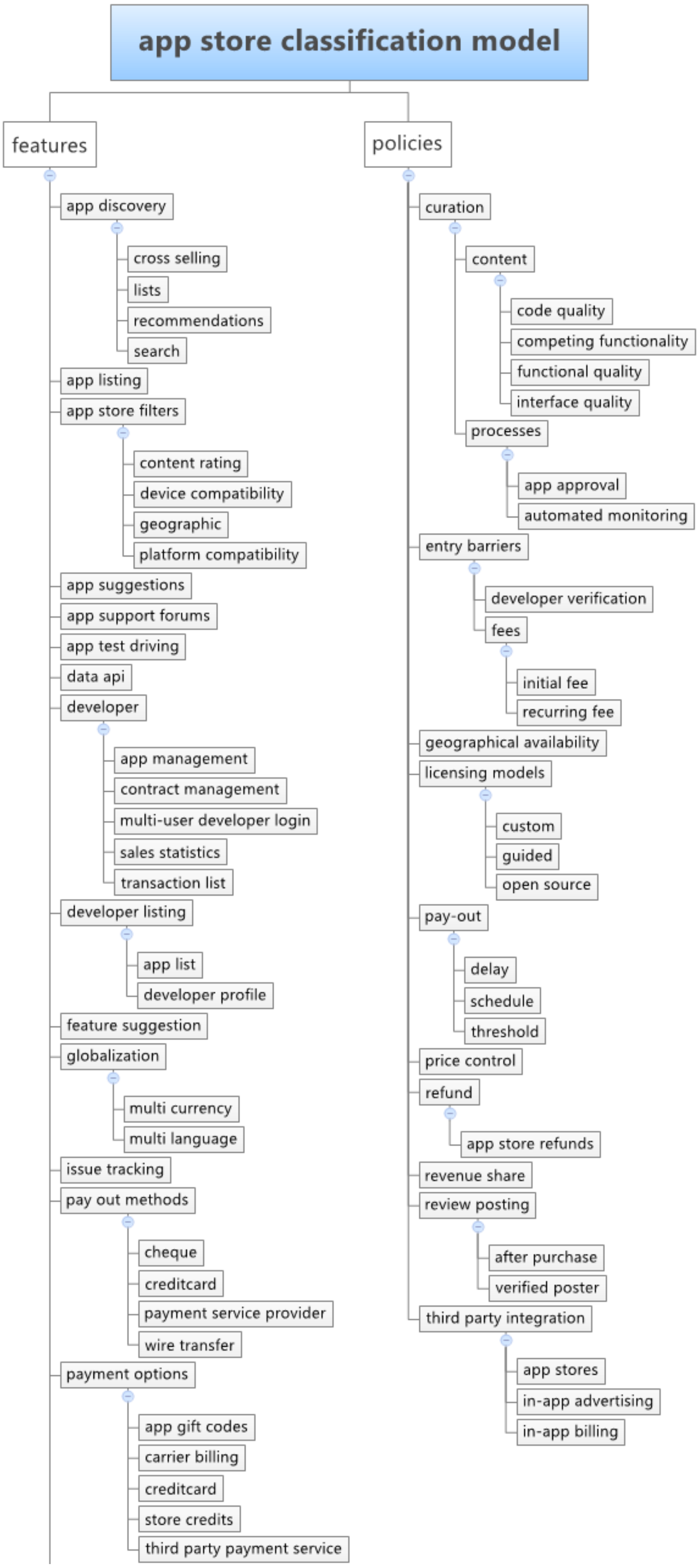


Figure 34 – Amazon appstore for Android: Refined app store classification model (part 1)

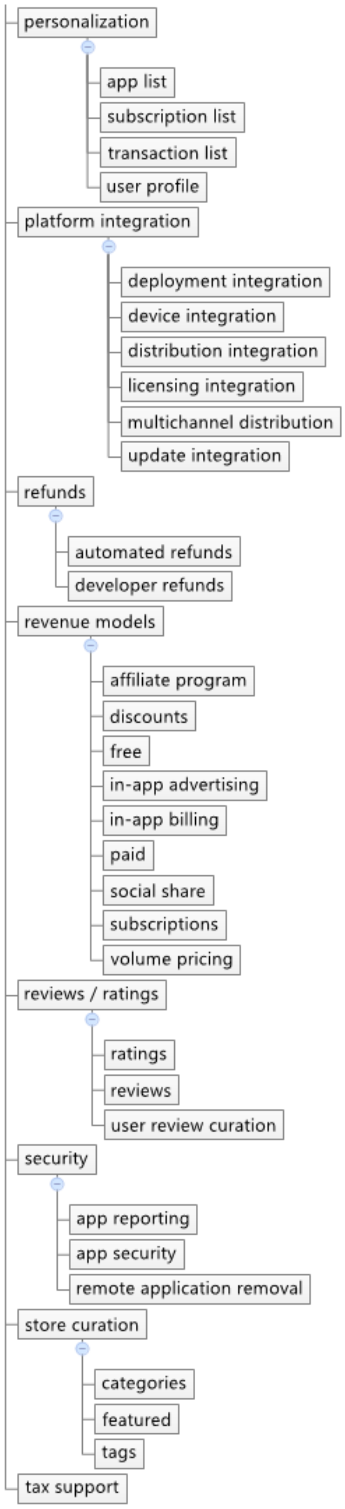


Figure 35 - Amazon appstore for Android: Refined app store classification model (part 2)

Revisited model

After the app store classification model has been made final this case study had to be updated to reflect those changes. The final list of features and policies can be found below.

Features

Features

Name	**Available**	**Notes**	**Reference**
affiliate stores	no		
app discovery			
cross selling	yes	List of apps customers ultimately bought after viewing a certain app. Also bought list.	35, 98
lists	yes	Top free, top paid, new, popular apps, popular games, featured	7, 8, 9, 10, 89, 90, 92
recommendations	yes		20
search	yes		46, 47, 48, 99
app listing	yes	Apps are listed with name, developer, price, description, reviews, rating, release date, screenshots, size, version, age rating, android platform, permissions	33, 34, 35, 36, 37, 38, 39, 62, 63, 64, 98
app store filters			
content rating	yes	Content is rated on a scale: children -9, all ages, 9+ 13+, 17+	65, 98, 102
device compatibility	partial	Apps are filtered for compatibility with the used device capabilities.	104, 105
geographic targeting	yes	Apps can be targeted to specific countries.	61, 107, 109
platform compatibility	yes	Apps are checked for minimum platform version.	98, 105
app suggestions	no		
app support forums	no		
app test driving	yes	Both on the website and on the native device.	106, 119, 120, 121
data API	yes	Amazon Marketplace Web Services provides an API.	123
developer			
app management	yes		80
beta testing	no		
contract management	no		
multi-user developer login	yes		85, 86, 106
sales statistics	yes		81, 82, 83, 106
transaction list	yes		81, 82, 83
developer listing			
app list	yes		42, 110
developer profile	partial	Description and contact information only.	42, 55, 110
feature suggestion	no		
globalization			
multi-currency	yes		62, 107
multi language	yes		63, 107
issue tracking	no		
pay out methods			
check	partial	Only outside USA	57, 109, 111
credit card	no		

payment service provider	no		57, 109, 111
wire transfer	partial	Only inside USA	57, 109, 111
payment methods			
app gift codes	no		
carrier billing	no		
credit card	yes		112
store credits	yes		26, 112
third party payment service	no		
personalization			
app list	yes		50, 93
subscription list	yes		94
transaction list	yes	connected to amazon account transaction list.	114
user profile	yes	connected with general amazon account profile.	113
platform integration			
deployment integration	no		
device integration	partial	Some devices come with amazon app store installed by default, most notably the Kindle.	1, 2, 3, 4, 5
distribution integration	yes	Apps are directly installed using the native app.	49, 50
licensing integration	yes		66, 106, 109
multichannel distribution	yes	Distribution both on website and native client.	7, 89, 115
update integration	yes	Updates are distributed using the native client.	53
refunds			
automated refunds	no		
developer refunds	no		
revenue models			
affiliate program	no	App store products are excluded from amazon affiliate program.	
component based	no		
discounts	yes	Daily deal where paid apps are free.	7, 61, 89, 91
free	yes		8, 61
in-app advertising	no		
in-app billing	yes		67, 68, 69, 70, 71, 72, 73, 74
paid	yes		8, 61
social share	no		
subscriptions	yes		75, 76, 77, 78, 79
volume pricing	no		
reviews / ratings			
ratings	yes	Five star system.	34, 116
reviews	yes		39, 40, 116
user review curation	yes	Reviews marked for helpfulness	98
security			
app reporting	yes		36
app security	yes	Android role based security is shown into app store listing.	98
remote application removal	no	Only if DRM is activated and code is signed using amazon generated certificate.	117, 118
store curation			

categories	yes	28 categories and 15 subcategories for games.	14, 15, 16, 17, 18, 19, 63, 89
featured	yes		21, 89
tags	partial	Search keywords can be added to an app.	63
tax support	yes		109

Policies

Policies			
curation			
content			
code quality	yes		106, 122
competing functionality	no		106
functional quality	no		106
interface quality	no		106
processes			
approval before publish	yes	Apps need to be approved and tested before they are published.	54, 106
automated monitoring	no		
entry barriers			
developer verification	no		
fees			
initial fee	no		
recurring fee	partial	$99 but currently waived.	106
geographical availability	1 (6)	USA only. UK, Italy, France, Germany and Spain later this year.	89, 107
licensing models			
custom	no		
guided	yes	Custom EULA can be provided but subject to default EULA	109
open source	no		109
pay-out			
delay	30	30 days after last day of calendar month	106, 109
schedule	monthly		106, 109
threshold	$10 / $100	$10 (US) or $100 (worldwide)	106, 109
price control	owner	Price is controlled by Amazon.	109, 134
app store refunds	partial	Negotiable through support.	109, 118, 126
revenue share	70% - 20%	70% of sales price, or 20% of list price.	109
review posting			
after purchase	no		
verified poster	yes	Reviews can be posted by verified user accounts , older than 48 hours, with confirmed transactions.	41
third party integration			
app stores	no		127, 128
in-app advertising	yes		109
in-app billing	no		106

Case study database

ID	Type	Document title	Year
1	Obs	Device screenshot 1	2012
2	Obs	Device screenshot 2	2012
3	Obs	Device screenshot 3	2012
4	Obs	Device screenshot 4	2012
5	Obs	Device screenshot 5	2012
6	Obs	Device screenshot 6	2012
7	Obs	Device screenshot 7	2012
8	Obs	Device screenshot 8	2012
9	Obs	Device screenshot 9	2012
10	Obs	Device screenshot 10	2012
11	Obs	Device screenshot 11	2012
12	Obs	Device screenshot 12	2012
13	Obs	Device screenshot 13	2012
14	Obs	Device screenshot 14	2012
15	Obs	Device screenshot 15	2012
16	Obs	Device screenshot 16	2012
17	Obs	Device screenshot 17	2012
18	Obs	Device screenshot 18	2012
19	Obs	Device screenshot 19	2012
20	Obs	Device screenshot 20	2012
21	Obs	Device screenshot 21	2012
22	Obs	Device screenshot 22	2012
23	Obs	Device screenshot 23	2012
24	Obs	Device screenshot 24	2012
25	Obs	Device screenshot 25	2012
26	Obs	Device screenshot 26	2012
27	Obs	Device screenshot 27	2012
28	Obs	Device screenshot 28	2012
29	Obs	Device screenshot 29	2012
30	Obs	Device screenshot 30	2012
31	Obs	Device screenshot 31	2012
32	Obs	Device screenshot 32	2012
33	Obs	Device screenshot 33	2012
34	Obs	Device screenshot 34	2012
35	Obs	Device screenshot 35	2012
36	Obs	Device screenshot 36	2012
37	Obs	Device screenshot 37	2012
38	Obs	Device screenshot 38	2012
39	Obs	Device screenshot 39	2012
40	Obs	Device screenshot 40	2012
41	Obs	Device screenshot 41	2012
42	Obs	Device screenshot 42	2012
43	Obs	Device screenshot 43	2012
44	Obs	Device screenshot 44	2012
45	Obs	Device screenshot 45	2012
46	Obs	Device screenshot 46	2012

ID	Type	Document title	Year
47	Obs	Device screenshot 47	2012
48	Obs	Device screenshot 48	2012
49	Obs	Device screenshot 49	2012
50	Obs	Device screenshot 50	2012
51	Obs	Device screenshot 51	2012
52	Obs	Device screenshot 52	2012
53	Obs	Device screenshot 53	2012
54	Obs	Web screenshot 1	2012
55	Obs	Web screenshot 2	2012
56	Obs	Web screenshot 3	2012
57	Obs	Web screenshot 4	2012
58	Obs	Web screenshot 5	2012
59	Obs	Web screenshot 6	2012
60	Obs	Web screenshot 7	2012
61	Obs	Web screenshot 8	2012
62	Obs	Web screenshot 9	2012
63	Obs	Web screenshot 10	2012
64	Obs	Web screenshot 11	2012
65	Obs	Web screenshot 12	2012
66	Obs	Web screenshot 13	2012
67	Obs	Web screenshot 14	2012
68	Obs	Web screenshot 15	2012
69	Obs	Web screenshot 16	2012
70	Obs	Web screenshot 17	2012
71	Obs	Web screenshot 18	2012
72	Obs	Web screenshot 19	2012
73	Obs	Web screenshot 20	2012
74	Obs	Web screenshot 21	2012
75	Obs	Web screenshot 22	2012
76	Obs	Web screenshot 23	2012
77	Obs	Web screenshot 24	2012
78	Obs	Web screenshot 25	2012
79	Obs	Web screenshot 26	2012
80	Obs	Web screenshot 27	2012
81	Obs	Web screenshot 28	2012
82	Obs	Web screenshot 29	2012
83	Obs	Web screenshot 30	2012
84	Obs	Web screenshot 31	2012
85	Obs	Web screenshot 32	2012
86	Obs	Web screenshot 33	2012
87	Obs	Web screenshot 34	2012
88	Obs	Web screenshot 35	2012
89	Obs	Web screenshot 36	2012
90	Obs	Web screenshot 37	2012
91	Obs	Web screenshot 38	2012
92	Obs	Web screenshot 39	2012
93	Obs	Web screenshot 40	2012
94	Obs	Web screenshot 41	2012

ID	Type	Document title	Year
95	Obs	Web screenshot 42	2012
96	Obs	Web screenshot 43	2012
97	Obs	Web screenshot 44	2012
98	Obs	Web screenshot 45	2012
99	Obs	Web screenshot 46	2012
100	Doc	MOBILE APP DISTRIBUTION AGREEMENT	2012
101	Doc	Amazon Appstore Gift Cards	2012
102	Obs	Web screenshot 47	2012
103	Obs	Web screenshot 48	2012
104	DocW	Amazon App Store hits v2.2.0 with new notification settings, improved app compatibility checks	2012
105	Doc	Amazon Appstore for Android Troubleshooting	2012
106	Doc	FAQs	2012
107	Doc	Get Ready for International App Distribution	2012
108	Doc	Introducing In-App Purchasing for Kindle Fire and other Android devices	2012
109	Doc	App Distribution Agreement	2012
110	Obs	Web screenshot 47	2012
111	Obs	Web screenshot 48	2012
112	Doc	Getting Started with Amazon Appstore for Android	2012
113	Obs	Web screenshot 49	2012
114	Obs	Web screenshot 50	2012
115	Doc	Purchasing apps	2012
116	Obs	Web screenshot 51	2012
117	DocW	Amazon App Store Requires Security Compromise	2011
118	Doc	Amazon Appstore for Android Troubleshooting	2012
118	Doc	Amazon Appstore for Android Terms of Use	2012
119	Obs	Web screenshot 52	2012
120	DocW	Amazon Appstore's Test Drive try before you buy feature now available on Android phones	2012
121	DocW	Amazon.com lets you play with an Android virtual machine, try apps before you buy them	2011
122	DocW	Amazon Appstore problems: why one developer pulled its game	2012
123	Doc	Amazon Marketplace Web Service	2012
124	DocW	Amazon Appstore Postmortem: A Guide to Android Developers	2012
125	Doc	Additional Terms Relating to Appstore Software	2012
126	DocW	How to get a refund from Amazon for an Android App Store purchase	2012
127	DocW	Supporting Amazon and Android market links inside application	2012
128	DocW	Feedback links for each market app	2011
129	DocW	Amazon’s AppStore hits 31,000 apps in one year, offers a week of app deals to celebrate	2012
130	DocW	Amazon's Android App Store Launches: Test Drive Apps Directly From Your Browser	2011
131	DocW	Amazon's Disruptive Android App Store Now Open To Developers — Full Details	2011
132	DocW	Amazon’s Appstore For Android Turns One: 31K Apps, Millions Sold	2012
133	DocW	Amazon’s Appstore: You’ll Make $0 When We Give Your App Away, And You’ll Like It	2012
134	DocW	Video game developers group slams Amazon pricing	2011
135	DocW	Distimo Publication - February 2012: The Amazon Appstore: Show Me the Money	2012

Table 14 – Overview of the case study database for the Amazon app store for Android

Each table row represents a document in the case study database. The documents are given numbers to identify them and are coded to indicate their source: “Lit” stands for scientific literature, “Obs” for direct observation, “Doc” for documentation provided by the owner and “DocW” for documentation retrieved from other places on the web.

Appendix C6 - Case Study Report: Intel AppUp

Data collection

Three sources of data has been used for this case study in accordance with the case study protocol. The first source of data is as always scientific literature. A Google scholar search however did not show papers relevant to our case study. The second data source used are direct observations. Using a Windows 7 pc and a Firefox web browser the Intel AppUp store was used. Also the Intel AppUp center for Windows 7 client application was installed. All actions taken have been thoroughly documented by screenshots which were stored in the case study database. In order to look at the developer side of the app store and to get access to developer documentation a developer account was created by the researcher.

The third source of data is documentation either as provided by Intel or as provide by third parties using or discussing the Intel AppUp app store. Documentation from Intel could be found at the developer support pages at appdeveloper.intel.com. Third party documentation was found by searching large blog sites such as PCMag and Huffingtonpost. Other documentation was found by performing Google web searches with keywords related to the Intel AppUp program.

All the relevant documents found were put in a case study document database as prescribed by the case study protocol. All the data was collected between December 2011 and July 2012. An overview of the case study database can be found in Table 15.

Refined model

To fit the Intel AppUp app store into the classification model a few changes had to be made. The first change was the addition of a affiliate stores feature. Intel AppUp sells its content also using custom branded affiliate stores. Also an beta testing feature was added to the developer features. AppUp provides developers with the possibility to invite beta testers to test their apps before publishing. The feature payment options was renamed to payment methods to be consistent with the naming of the payout methods feature. A component based revenue model was added as a feature. Intel AppUp allows developers to publish software components to other developers. When other developers use these software components and sell their apps the component developer receives part of the revenue. On the policy side of the model the app approval policy was renamed to approval before publish. This name was deemed a bit more descriptive.

The resulting app store classification model can be found in Figure 36 and Figure 37.

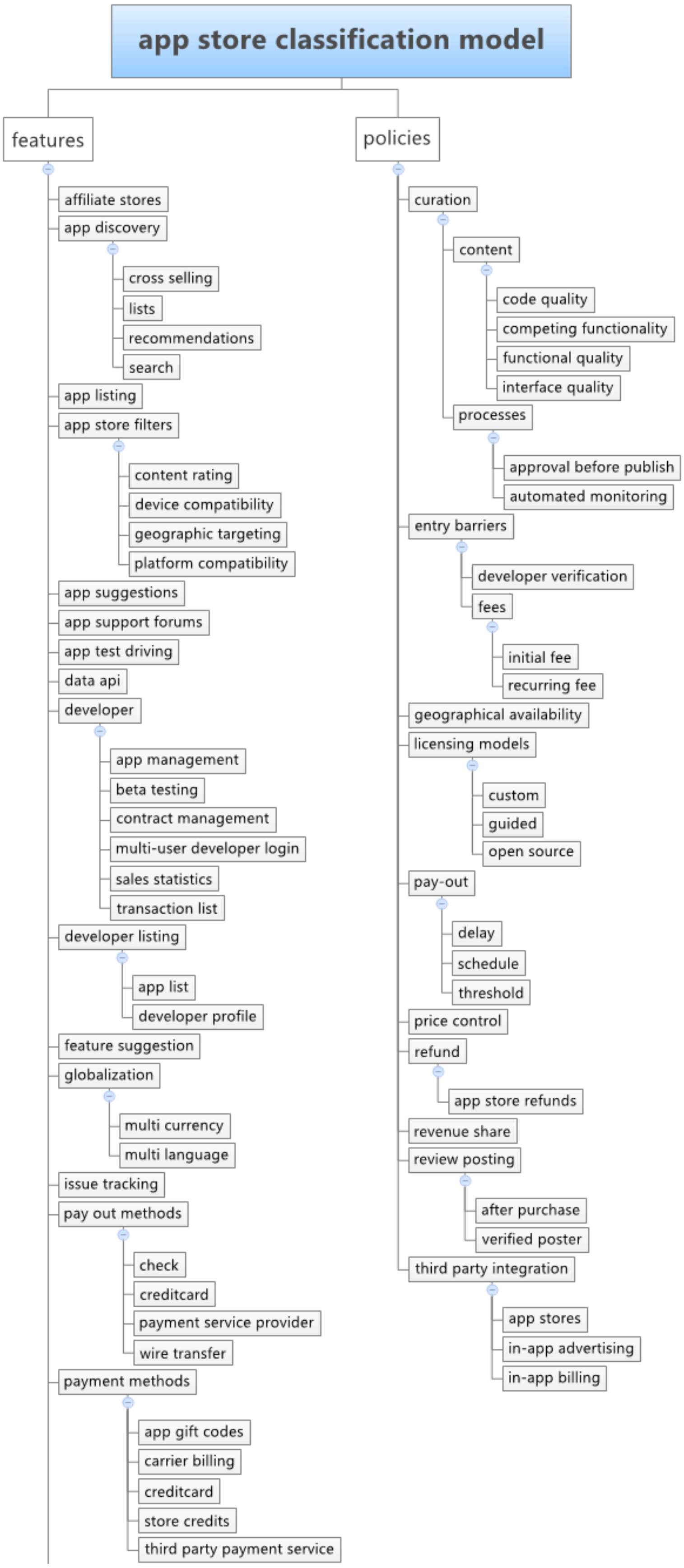


Figure 36 – Intel AppUp: Refined app store classification model (part 1)

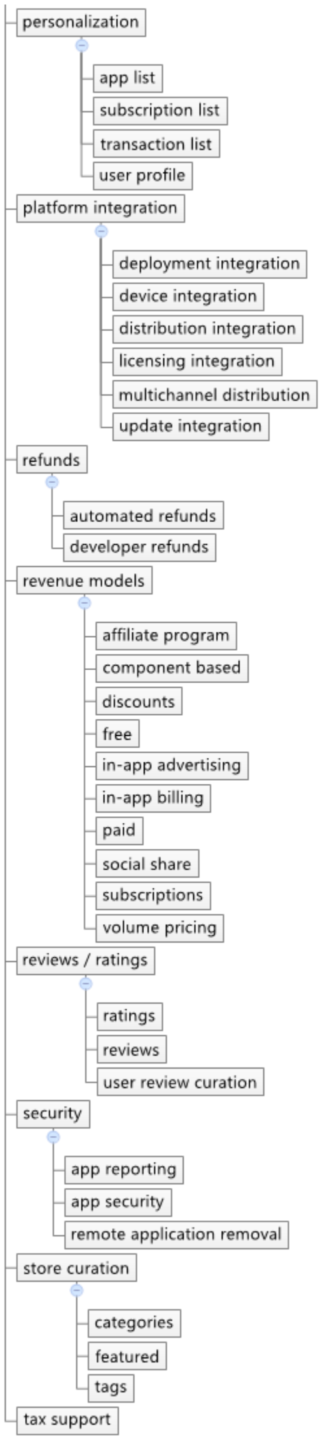


Figure 37 - Intel AppUp: Refined app store classification model (part 2)

Refined model details

In the following table each feature and policy found in the Intel AppUp app store is described using yes, no and partially to describe their availability. If needed additional information on the availability of a certain feature or policy is added to the notes column. One or more references to the document database are added to ensure each description is underpinned with sufficient evidence from the case study documents. Features marked with n/a means that the feature is not applicable. This can be because of the platform constraints or because of other missing features that are a prerequisite to that feature.

Features

Features

Name	Available	Notes	Reference
affiliate stores	yes	There are over 20 affiliate stores with their own branding.	82
app discovery			
cross selling	yes	Related apps for each app listing	19
lists	yes	Top Paid, Top Free, Featured, New Arrivals	4, 5, 6, 7, 36, 37
recommendations	no		
search	yes		73, 74
app listing	yes	Listing with name, description, rating, reviews, screenshots, audience rating, release date, version, size, developer and license.	19, 20, 58, 59, 60, 61
app store filters			
content rating	yes	Available categories are 3+, 6+, 10+, 13+ and 17+	59, 80
device compatibility	partial	Only on capability level.	60
geographic targeting	yes		59
platform compatibility	yes		60
app suggestions	no		
app support forums	no		
app test driving	no		
data API	no		
developer			
app management	yes		47, 48, 75
beta testing	yes	The developer portal has a special section to invite beta testers. Beta testers will see the app in their app list.	61, 77
contract management	yes		49
multi-user developer login	yes		46
sales statistics	partial	Monthly generated report only	53
transaction list	yes		53
developer listing			
app list	no		
developer profile	no		
feature suggestion	no		
globalization			
multi-currency	yes		59
multi language	yes		58
issue tracking	no		
pay out methods			
check	no		44

credit card	no		44
payment service provider	yes	PayPal	44
wire transfer	no		44
payment methods			
app gift codes	yes		17
carrier billing	na		
credit card	yes		14
store credits	no		
third party payment service	no		
personalization			
app list	yes		11
subscription list	na		
transaction list	yes		15
user profile	no		
platform integration			
deployment integration	no		
device integration	partial	Not by default available on devices in the ecosystem, except for certain certified devices.	1, 2, 3, 79
distribution integration	yes		23, 24, 25, 29, 30
licensing integration	yes	Through Intel AppUp™ SDK	81
multichannel distribution	no		
update integration	yes		12
refunds			
automated refunds	no		
developer refunds	no		
revenue models			
affiliate program	no		
component based	yes	AppUp allows developers to sell components to other app developers and take a share their revenue.	66, 78
discounts	no		
free	yes		37, 59
in-app advertising	no		
in-app billing	yes	Through a partnership with Urban Airship	60
paid	yes		36, 58
social share	no		
subscriptions	no		
volume pricing	no		
reviews / ratings			
ratings	yes		32
reviews	yes		32
user review curation	no		
security			
app reporting	yes		35
app security	no		
remote application removal	no		
store curation			
categories	yes	22 categories with several subcategories.	8, 59
featured	yes		5, 7
tags	partial	Keyword system used for search purposes, no tag filtering.	60

tax support	yes		59, 85, 86

Policies

Policies			
curation			
content			
code quality	yes		80
competing functionality	no	E.g. self-updating or over-the-air updating applications are not allowed.	80
functional quality	no		
interface quality	yes		
processes			
approval before publish	yes		80, 83
automated monitoring	no		
entry barriers			
developer verification	no		
fees			
initial fee	no		41
recurring fee	partial	Normally $99 but currently waived	41
geographical availability	47	47 countries.	78, 84
licensing models			
custom	no	Custom terms should not conflict with standard EULA	80
guided	yes	default license available, custom terms should not contradict.	58, 80
open source	yes		58
pay-out			
delay	0		86
schedule	monthly		86
threshold	0		86
price control	developer		86
app store refunds	partial	Sales generally are final, few exceptions can be negotiated.	87
revenue share	70%		86
review posting			
after purchase	yes		25, 31
verified poster	no		
third party integration			
app stores	no		80
in-app advertising	yes		78
in-app billing	no		80

Case study database

ID	Type	Document title	Year
1	Obs	Screenshot 1	2012
2	Obs	Screenshot 2	2012
3	Obs	Screenshot 3	2012
4	Obs	Screenshot 4	2012
5	Obs	Screenshot 5	2012
6	Obs	Screenshot 6	2012

ID	Type	Document title	Year
7	Obs	Screenshot 7	2012
8	Obs	Screenshot 8	2012
9	Obs	Screenshot 9	2012
10	Obs	Screenshot 10	2012
11	Obs	Screenshot 11	2012
12	Obs	Screenshot 12	2012
13	Obs	Screenshot 13	2012
14	Obs	Screenshot 14	2012
15	Obs	Screenshot 15	2012
16	Obs	Screenshot 16	2012
17	Obs	Screenshot 17	2012
18	Obs	Screenshot 18	2012
19	Obs	Screenshot 19	2012
20	Obs	Screenshot 20	2012
21	Obs	Screenshot 21	2012
22	Obs	Screenshot 22	2012
23	Obs	Screenshot 23	2012
24	Obs	Screenshot 24	2012
25	Obs	Screenshot 25	2012
26	Obs	Screenshot 26	2012
27	Obs	Screenshot 27	2012
28	Obs	Screenshot 28	2012
29	Obs	Screenshot 29	2012
30	Obs	Screenshot 30	2012
31	Obs	Screenshot 31	2012
32	Obs	Screenshot 32	2012
33	Obs	Screenshot 33	2012
34	Obs	Screenshot 34	2012
35	Obs	Screenshot 35	2012
36	Obs	Screenshot 36	2012
37	Obs	Screenshot 37	2012
38	Obs	Web Screenshot 1	2012
39	Obs	Web Screenshot 2	2012
40	Obs	Web Screenshot 3	2012
41	Obs	Web Screenshot 4	2012
42	Obs	Web Screenshot 5	2012
43	Obs	Web Screenshot 6	2012
44	Obs	Web Screenshot 7	2012
45	Obs	Web Screenshot 8	2012
46	Obs	Web Screenshot 9	2012
47	Obs	Web Screenshot 10	2012
48	Obs	Web Screenshot 11	2012
49	Obs	Web Screenshot 12	2012
50	Obs	Web Screenshot 13	2012
51	Obs	Web Screenshot 14	2012
52	Obs	Web Screenshot 15	2012
53	Obs	Web Screenshot 16	2012
54	Obs	Web Screenshot 17	2012

ID	Type	Document title	Year
55	Obs	Web Screenshot 18	2012
56	Obs	Web Screenshot 19	2012
57	Obs	Web Screenshot 20	2012
58	Obs	Web Screenshot 21	2012
59	Obs	Web Screenshot 22	2012
60	Obs	Web Screenshot 23	2012
61	Obs	Web Screenshot 24	2012
62	Obs	Web Screenshot 25	2012
63	Obs	Web Screenshot 26	2012
64	Obs	Web Screenshot 27	2012
65	Obs	Web Screenshot 28	2012
66	Obs	Web Screenshot 29	2012
67	Obs	Web Screenshot 30	2012
68	Obs	Web Screenshot 31	2012
69	Obs	Web Screenshot 32	2012
70	Obs	Web Screenshot 33	2012
71	DocW	Intel's AppUp Boss Swims Against the App Store Tide	2011
72	DocW	Intel AppUp 1 Year Later	2011
73	Obs	Screenshot 38	2012
74	Obs	Screenshot 39	2012
75	Doc	How to Use My Dashboard	2012
76	Doc	Quick start guide	2012
77	Doc	Beta Testing	2012
78	Doc	Frequently asked questions	2012
79	DocW	Asus First to Put Intel's AppUp Store on Netbooks	2010
80	Doc	Intel AppUp Developer Program Application/Component Suitability and Validation Guidelines	2012
81	Doc	Importance of Initialization and Authorization	2012
82	Doc	Affiliate Stores	2012
83	Doc	How to Submit an Application	2012
84	Doc	Intel AppUp(SM) center Supported Countries List	2012
85	Doc	Withholding Taxes (Updated March 2010)	2010
86	Doc	Component and Application Submission Distribution Agreement	2010
87	Doc	Customer Account Registration Agreement	2012

Table 15 – Overview of the case study database for the Intel AppUp app store

Each table row represents a document in the case study database. The documents are given numbers to identify them and are coded to indicate their source: “Lit” stands for scientific literature, “Obs” for direct observation, “Doc” for documentation provided by the owner and “DocW” for documentation retrieved from other places on the web.

Appendix D – Long list

In order to create the long list the following queries were used:

	Site	Date	Keywords	results	Inspected	Url
1	Techcrunch	23-1-2012	app store launch	20600	1000	http://techcrunch.com/search/app+store+launch
2	Techcrunch	26-1-2012	app store -iOS -itunes -apple	12900	500	http://techcrunch.com/search/app+store+-iOS+-itunes+-apple
3	List of mobile software distribution platforms	26-1-2012	none			http://en.wikipedia.org/wiki/List_of_mobile_software_distribution_platforms
4	Distimo's App store info	1-2-2012	none			http://www.distimo.com/appstores/

These four queries allowed the creation of the following list of 81 app stores. If the data was readily available the number of apps on the app store and the number of downloads from the app store are also listed, as well as the introduction date of the app store.

Name	Url	Platform type	Ecosystem	# Apps	# Downloads	Starting date	Query
AllMyApps	http://allmyapps.com/	Desktop	Windows			16-12-2010	1
Amazon Appstore for Android	http://www.amazon.com/mobile-apps/b?ie=UTF8&node=2350149011	Mobile	Android	3.800		22-3-2011	1
Ammapp	http://ammapp.ru/	Mobile	Windows Mobile			1-12-2009	3
AndAppOnline	https://www.andapponline.com/	Mobile	Android			1-3-2010	3
Android Market	https://market.android.com/	Mobile	Android	221.189	5.000.000.000	22-10-2008	1
AndroidPit App Center	http://www.androidpit.com/en/android-market	Mobile	Android			1-10-2010	3
App Catalog	https://developer.palm.com/	Mobile	Palm/HP	7.062	106.000.000	6-6-2009	1
App direct	http://www.appdirect.com/	Web					2
AppBistro	http://appbistro.com	Web	Facebook	101			2
Appcelerator Open Mobile Marketplace	https://marketplace.appcelerator.com/home	Mobile, Desktop	Code, Appcelerator Titanium			19-9-2011	1
Appia	www.appia.com	Mobile	Appia (Android, iOS, Blackberry, Symbian, Java, Palm, Windows Mobile)	140.000	280.000.000	1-6-2008	1

Appitalism	http://www.appitalism.com	Mobile				1-1-2010	3
Apple App Store	http://www.apple.com/itunes/	Mobile	iOS	425.000	14.000.000.000	10-7-2008	1
Appoke	http://appoke.com/	Mobile	Android				2
Appshup	http://api.smsgupshup.com/apps	Mobile	Gupshup			3-3-2010	1
Appslib	http://appslib.com/	Mobile	Android	38771			2
AppsonAvenue	https://www.appsonavenue.com/	Desktop	Windows				2
Appsplit marketplace	http://www.appsplit.com/marketplace/	Mobile, Facebook, Web	Code			12-1-2011	1
Bell Canada Business App Store	https://businessapps.bell.ca/landing	Web					2
Benchprep	benchprep.com	Mobile, Web	Benchprep			31-5-2011	1
BinPress	http://www.binpress.com/	Code		204		4-10-2010	1
Blackberry App World	http://us.blackberry.com/apps-software/appworld/	Mobile	RIM, Blackberry OS	32.367	3000000 / day	1-4-2009	1
Carbyn Store	http://about.carbyn.com	Web	HTML5			Not yet open	1
CBSSports.com App Central	http://developer.cbssports.com/	Web	Sports Fantasy Games			31-1-2012	1
Cellmania	http://www.cellmania.com/	Mobile	Multiple			1-6-1999	3
China Mobile Mobile Market	http://mm.10086.cn/	Mobile					1
Chrome Web App Store	https://chrome.google.com/webstore/category/home	Web	Chrome	18.000	1200000000 / month	1-12-2010	1
Chupa	http://www.chupamobile.com/	Mobile	Code			14-10-2011	1
Cisco AppHQ	https://marketplace.cisco.com/apphq	Tablet	Cius			29-6-2011	4
CloudWrangler	http://verecloud.com/index.html	Web					2
CodeCanyon	http://codecanyon.net/	Code		3128			Other
Cyworld Appstore	http://appstore.cyworld.com/	Web	CYWORLD, Opensocial				2
Dell Mobile application store	http://www.dellmobileappstore.com	Mobile	Multiple			12-8-2010	4
DirectTV App store	http://tvapps.directv.com/index.do	TV					2
Download.com	http://download.cnet.com/	Desktop					2
Fasmicro Android App Store	http://apps.fasmicro.net/	Mobile	Android			1-6-2011	3

Foursquare App Gallery	https://foursquare.com/apps/	Web	FourSquare	500			1
Games for windows Marketplace	http://marketplace.xbox.com	Desktop	Games, Windows OS	260		15-11-2011	Other
GetApp	http://www.getapp.com	Web					1
GetJar	http://www.getjar.com/	Mobile	Java, Android, iOS, Symbian	155.550	1.955.503.545	1-1-2005	1
Google Apps Marketplace	https://www.google.com/enterprise/marketplace/	Web	Google Apps	600		1-3-2010	1
Handango	http://www.handango.com/	Mobile	Multiple			1-1-1999	3
Handmark	store.handmark.com/	Mobile	Multiple			1-8-2000	3
Handster	http://www.handster.com/	Mobile	Multiple			26-5-2005	4
Happtique	http://www.happtique.com/	Mobile	iOS				2
Indiroid	https://indiroid.com/	Mobile	Android			1-9-2011	3
Intel AppUp	http://www.appup.com	Desktop, Netbook, Tablet	Windows OS, Meego, Adobe AIR			14-9-2010	2
Intuit App Center	http://marketplace.intuit.com/	Web	Intuit				Other
Kindertown	http://www.kindertown.com/	Mobile / Tablet	iOS			11-11-2011	1
LG SmartWorld	http://us.lgworld.com/web.main.dev	Mobile	Android, Windows Mobile			29-7-2010	3
MAC app store	http://www.apple.com/mac/app-store/	Desktop	OS X	5.000		6-1-2011	1
Mikandi	http://www.mikandi.com/	Mobile	Android			23-11-2010	1
Mobango	http://www.mobango.com	Mobile	Multiple	100000	1000000000		3
Mobihand	http://www.mobihand.com/	Mobile	Blackberry, Symbian, Android, Windows Mobile	50000			2
Mobspot	http://www.mobspot.com/	Mobile	Multiple				3
Motorola Shop4Apps	http://developer.motorola.com/shop4apps/	Mobile	Android			21-1-2010	4
Nduoa	http://www.nduoa.com/	Mobile	Android			1-3-2010	3
Nokia Ovi Store	http://store.ovi.com/	Mobile	Symbian	71.753	1.800.000.000	26-5-2009	1
Nook Store	http://www.barnesandnoble.com/u/nook-store/379002961/	Tablet	Nook			24-3-2011	1
Openappmkt	http://openappmkt.com/	Mobile	HTML5				2
Opencart Extension Store	http://www.opencart.com/index.php?route=extension/extension	Code	OpenCart	2.142			Other
Opera Mobile Store	http://apps.opera.com/	Mobile	Appia (Android, iOS, Blackberry,			7-3-2011	1

			Symbian, Java, Palm, Windows Mobile)				
PageLines Store	http://www.pagelines.com/store/	Web	Wordpress / Pagelines 2.0			8-12-2012	1
Podio App Store	https://podio.com/store	Web	Podio			24-3-2011	1
Robot App Store	http://www.robotappstore.com/	Robot				Not yet open	1
Salesforce App Exchange	http://appexchange.salesforce.com/home	Web	Salesforce CRM	1.187		1-1-2006	1
Samsung Apps	http://www.samsungapps.com	Mobile, TV	Bada	13.000	1.000.000	1-6-2010	1
Slideme	http://slideme.org/	Mobile	Android			1-1-2008	3
SmartAppFinder	https://www.smartappfinder.com	Mobile	Android			1-10-2010	3
Socio Mall	http://mall.soc.io/Home	Mobile	Android				2
Sony Ericson Playnow Arena	http://www.playnow-arena.com/	Mobile	Multiple			10-8-2009	4
Steam	http://store.steampowered.com/	Desktop	Games, Windows OS, OS X	1.200		1-12-2005	Other
Vcast Webstore	http://products.verizonwireless.com/	Mobile	Android			29-3-2010	1
Verious	http://www.verious.com/	Mobile	Code			12-9-2011	1
Vodafone Mobile Application Store (India)	live.vodafone.in	Mobile	Appia (Android, iOS, Blackberry, Symbian, Java, Palm, Windows Mobile)			31-10-2011	1
WCM Exchange	http://www.wcmexchange.com/	Code		243			Other
Webs App Store	http://www.webs.com/applist.htm	Web	Webs				1
Windows Azure Marketplace	https://datamarket.azure.com/	Web	Azure				Other
Windows Phone Marketplace	http://www.windowsphone.com/en-US/marketplace	Mobile	Windows Phone 7	24.102		21-10-2010	1
WPPlugins	http://wpplugins.com/	Web	Wordpress	132			2
Yahoo Connected TV Store	http://connectedtv.yahoo.com/developer/tvstore/	TV	Connected TV			1-1-2011	1

Appendix E – Mapping features and policies to strategy factors

	User focused	Developer focused	Innovation	Control	Growth	Monetization
User focused features						
app findability						
recommendations	+					
store curation tags	+					
app quality						
app security integration	+					
app security reporting	+					
app test driving	+					
content rating filter	+					
device compatibility filter	+					
remote application removal	+			+		
user review curation	+					
app store usability						
automated refunds	+					
developer refunds	+	+				
device integration	+					
multi language	+					
multichannel distribution	+					
multi-currency	+					
update integration	+					
user app list	+					
user subscription list	+					
user transaction list	+					
Developer focused features						
feedback potential						
app suggestions		+				
app support forums	+	+				
developer beta testing management		+				
feature suggestion		+				
issue tracking		+				
user profile		+				
monetization potential						
affiliate program		+				
affiliate stores		+				
component based revenue model		+				
discounts revenue model	+	+				
in-app advertising		+				
in-app billing		+				
licensing integration		+				
social share revenue model		+				
subscriptions		+				
volume pricing	+	+				
app store usability						
data API		+				
deployment integration		+				
developer contract management		+				
developer multi-user login		+				
developer sales statistics		+				
geographic targeting	+	+				
tax support		+				

	User focused	Developer focused	Innovation	Control	Growth	Monetization
visibility						
cross selling	+	+				
developer app list		+				
developer profile		+				
User focused policies						
app quality						
approval before publish	+			+		
automated monitoring	+		+			
code quality curation	+			+		
functional quality curation	+			+		
interface quality curation	+			+		
review posting after purchase	+			+		
reviews posting verified poster	+			+		
developer quality						
developer verification	+			+		
recurring fee	+					
app store usability						
app store refunds	+					
Developer focused policies						
monetization potential						
pay-out delay		-				
pay-out schedule		-				
pay-out threshold		-				
price control		+				
revenue share		+				-
third party app stores allowed		+	+			
third party in-app advertising allowed		+	+			
third party in-app billing allowed		+	+			
openness						
competing functionality curation		+		+		
custom licensing		+	+			
guided licensing		+		+		
open source licensing		+	+			
visibility						
geographical availability		+				