Defining App Stores: The Role of Curated Marketplaces in Software Ecosystems

Slinger Jansen and Ewoud Bloemendal

Utrecht University
slinger@slingerjansen.nl,webloem@gmail.com

Abstract. The app store is a novel concept in the software business, that has changed the way in which customers perceive software and its day-to-day use. The concept, however, is poorly understood, which can be observed by lack of a comprehensive definition and relatively little literature on the topic. This paper provides a definition of app stores, provides a conceptual model of the concept, and supplies typical features and policies that are observed in app stores, using six case studies. The increased understanding that the research provides, aims to help practitioners make their app store more successful and provides researchers with a frame for defining and analyzing app stores.

Key words: App stores, software ecosystems, comparative multiple case study

1 Introduction

The product software business is a fast changing business. One of the most powerful changes that the software business currently is experiencing is the introduction of app stores, which are marketplaces for applications that are available for instant download. App stores are influencing the industry in the following major ways. First, people are becoming more aware of the software business: with so much software available in everyone's pocket, increasing numbers of people are exposed to the app business. Furthermore, due to the low prices of apps in the app store, business models are radically changing to constantly add value to the product such as content, as to generate equal amounts of revenue from complex software systems as was possible in the "old days" of license and maintenance models. Thirdly, app stores appear to be the method of choice to build up a healthy software ecosystem [10, 8, 6]. It is surprising that this topic has not received more attention over the past years. In this paper we aim to fill that gap, by providing an exhaustive overview of the features and policies that determine the structure of an app store. Furthermore, we provide the following definition of app stores.

> **App store:** *An online curated marketplace that allows developers to sell and distribute their products to actors within one or more multi-sided software platform ecosystems.*

Jansen, Finkelstein, and Brinkkemper define a software ecosystem as 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 [9]. An app store can be seen as a catalyst in such a software ecosystem. The services it offers are part of the common platform of the ecosystem and it can have a pivotal role in creating the common market. App stores allow developers to monetize their software and bring consumers new functionality. A successful app store is beneficial to the success of a software ecosystem which consequently can be beneficial to the company that owns it, or according to a Deutsche Bank analyst, Apple's app store is the "the competitive moat which competing handset vendors cannot replicate" [3].

The literature on app stores is 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 received the most attention in literature. The success of Apple's app store has led to a fair amount of literature [2, 10] specifically about the iOS ecosystem. A broader perspective is given by Holzer and Ondrus [5] when they take a developer view on the significant structural changes in the mobile application market. 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 ecosystems 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 [5] propose that a fully integrated platform facilitates the flow through the distribution process.

At the moment no literature on app stores has been found that combines a broad software ecosystem scope with the perspective of an (aspiring) app store owner and therefore this study will try to fit in this gap. The objective for this study is help app store owners gain more insight in the app stores by creating an app store definition and performing a multiple case study on app store features and policies.

In order to provide more insight into app stores and their features and policies, this paper uses the following structure. In Section 2 an app store definition is introduced together with a conceptual model of an app store. In Section 3 the case study method is explained, the cases are introduced, and we provide a brief discussion on validity of the research. In Section 4 the models resulting from the case studies are listed with a table of feature and policy descriptions. The study concludes with the key findings, limitations and opportunities for further research. The highlights of the research are the app store definition, the poli-

cies and features, and the finding that most app store coordinators are aiming to cover all features and policies instead of consciously deciding to leave out features and policies.

2 Conceptual Model: Defining App Stores

We introduce the following definition of an app store: *An online curated marketplace that allows developers to sell and distribute their products to actors within one or more multi-sided software platform ecosystems.*

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 of this element was added to distinguish the app store from a brick and mortar stores. *Software ecosystems* are part of the definition to emphasize the relation between an app store and its ecosystem. It the following part two groups of users are identified: *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 is made to limit the scope of what is considered an app store in this research. To be considered an app store a system should: (1) be available using the internet, (2) be curated by an organization, typically but not necessarily the platform owner, (3) allow for the selling and buying of software products, (4) take care of the financial transactions involved in selling the software products, (5) have two distinct user groups: developers and users, (6) be serving one or more software ecosystem, and (7) implement a platform that takes care of the distribution of the software products. Please note that an app store can apply to one ecosystem, such as Google Play serving the Android ecosystem, or multiple ecosystems, such as the BinPress app store, where code can be purchased for multiple platforms and ecosystems.

App stores would not exist if it were not for platforms: a set of solutions to problems that is made available to the members of the ecosystem through a set of access point or interfaces [7]. Furthermore, Hagiu and Wright speak of a multi-sided platform as being an organization that creates value primarily by enabling direct interactions between two (or more) distinct types of affiliated customers [4]. Three platform types are distinguished by Baldwin and Woodard [1]: 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.". According to Baldwin and Woodard, 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.

In order to create a better understanding of how the app store acts within its software ecosystem a conceptual model of app store mechanics is proposed in Figure 1. In the ovals the different actors within the ecosystem are modeled. The first one on the bottom is the owner, which is not necessarily the owner of the ecosystem but rather the owner of the app store. The other two ellipses both represent a set of actors rather than one actor. The first set of actors are the end 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 represented by the arrows from and to the app store triangle.

Within the triangle three concepts are depicted: two bottom squares *features* and *policies* and at the top the *characteristics* square. Features represent individual parts of the software systems that the actors can interact with. Policies represent the rules, regulations and governing processes that limit the functional reach of the features. Features and policies together form the part of an app store that the owner can directly influence. The characteristics square represents a set of app store characteristics that cannot be directly influenced by the app store owner. An example of a such characteristic would be the total number of apps available in an app store. Other examples of these characteristics are the number of developers, the number of end users, the quality of the apps or the usability of the app store software. The arrows going from the features and policies to the characteristics represent the assumption that app store owners try to influence the characteristics by implementing a certain set of features and policies.

3 Research Approach: Identifying Features and Policies

In order to identify features and policies in existing app stores, case studies were performed using a multiple case study method based on the case study methods of Yin [11]. At first, a case study protocol was created to ensure a consistent research execution amongst the cases. Secondly, a long-list of existing app stores was created based on a set of web search queries, resulting in a list of 81 app stores. The following criteria were used to create a convenient sample for the research. The first criterion checked whether the researcher had access to the app store and could get a complete overview of the features and policies. The second set of criteria aims to improve the generalization of the sample by selecting app stores from multiple vendors and only allowing app stores with a minimum of 1000 transactions and existing longer than 6 months. This resulted in the following six app stores that were selected for the case study: Google

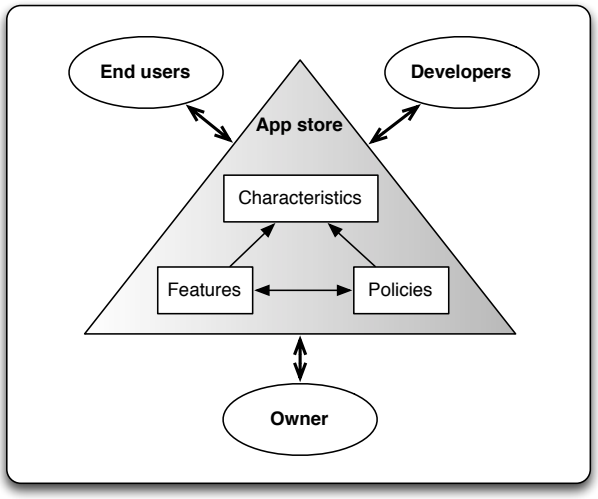

Fig. 1. Conceptual Model of an App Store

Research Step	**# Features**	**# Policies**	**# Apps**	**Paid (%)**
Initial model (from literature)	14	8		
After Case 1: Google Play	28	10	450,000	28%
After Case 2: SlideMe	35	17	20,000	26%
After Case 3: Apple Appstore	40	21	650,000	66%
After Case 4: Binpress	57	20	30,000	65%
After Case 5: Amazon appstore	64	24	369	100%
After Case 6: Intel AppUp	67	24	3,000	64%

Table 1. Growth of the Data Set in Different Phases of the Research

Play, SlideMe, Apple Appstore, Binpress, Amazon app store for Android and Intel AppUp.

The goal of the six case studies was to create an overview of the features and policies of existing app stores that accurately describes the features and policies of an app store. For each of the six selected app stores data was gathered on its features and policies. Data for the case studies was gathered by gaining access to the app store and making screenshots to of features. Documentation was used and collected from three different sources: by using the support documentation provided by the app store owner, by directly searching tech blogs and news websites, and lastly by blogs referred to by general web searches.

For each case a case study database was created in which all documents were stored. The data in the database was then analyzed and the observed features and policies were filled in and if needed features or policies were added, removed or renamed. Whenever possible data from multiple convergent sources was used to determine the existence of a feature or policy. As Yin suggested the case

studies were 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 model. In order to create a starting point for the first case study a preliminary model was created using blog posts comparing app stores. In Table 1 the growth of the model through this process is shown. After the completion of the model all previous cases were revisited in order to complete the dataset.

3.1 Validity

First of all **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 feature or policy was 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 1 show that the model did not need many changes between the fifth and the sixth case suggesting that the model is approaching 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.

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 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.

3.2 Case Descriptions

The first case study was done on Google Play, launched in 2008 and now the biggest app store in the Android ecosystem. It serves the Android ecosystem, an open source operating system for mobile devices and tablet computers. 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 from Android Market to Google Play and it became a digital multimedia content service that also sells books, music and movies.

The second case is SlideMe app store, which launched in April 2008 and introduces itself as the third app store in the Android ecosystem, after Google

Play and the Amazon Appstore. SlideMe provides an alternative app store for devices that for some reasons cannot install Google Play and is used by over 120 OEM devices as their alternative app store. SlideMe does not charge transaction fees to developers other than the payment processing fees because SlideMe earns its money by providing white-label curated app stores for OEM devices, i.e., manufacturers can use the SlideMe software and rebrand it as their own for a service fee.

The third case is the Apple app store, launched in July 2008. The Apple app store 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. 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

The fourth researched app store is Binpress, founded in 2011 as a marketplace for source code. Binpress hosts source code for a variety of twelve programming languages with 16 frameworks and 18 platforms. Because Binpress is a marketplace that sells and buys from developers to developers it offers features tailored to developer needs, such as issue tracking and component support forums.

The fifth app store used as case is the Amazon app store for Android which opened in March 2011. It is the second most important app store in the Android software ecosystem and was founded by online retail giant Amazon. 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 app store for Android from Google Play are the availability of the Deal of the Day promotion and the possibility to test drive apps before buying.

The last case study is the app store introduced in January 2010 by semiconductor manufacturer Intel called Intel AppUp. The initial focus of the Intel AppUp store was on applications for the at that time popular netbook devices using the Intel Atom processor platform. Nowadays the Intel AppUp store hosts apps for platforms Adobe Air, Microsoft Silverlight, the Linux-based MeeGo operating system for mobile devices and native Windows XP and Windows 7 applications.

4 Case Results

After all features and policies were identified and the data for each of the case studies was available a set of common features shared by all app stores could be created. This corresponds with the fact that all app stores were selected using the same definition and set of criteria. The fifteen core features are shown in Table 2. Each core feature can be mapped to a part of the app store definition: ‘app categories’, ‘app listing’, ‘app lists’, ‘featured apps’, ‘ratings’, ‘reviews’ and

Core feature	Descriptions
app categories	Apps are listed in categories and subcategories
app listing	Apps are listed with full description, images, etc.
app lists	apps are listed, e.g. top selling lists or latest additions
dev app management	Devs can manage their apps in a developer console
dev transaction list	Devs can manage their transactions
distribution integration	Distribution and installation happens through platform
featured apps	Apps can be featured to receive more attention
free revenue model	Apps can be offered for free
paid revenue model	Apps can be sold
pay out methods	Number of pay out methods
payment methods	Number of payment methods
platform comp. filter	Apps have information on their platform compatibility
ratings	Apps can be rated by the user
reviews	Users can read and write reviews of an app
search	Users can search for apps using search keywords

Table 2. Core app store features

the 'search' feature can be mapped to the 'online curated marketplace' part, 'developer app management', 'developer transaction list', 'distribution integration', 'free revenue model', 'paid revenue model', 'pay out methods' and 'payment methods' map back to 'allows software developers to sell and distribute their products to users' and 'platform compatibility filter' maps back to the 'software platform' part. In Table 2 the descriptions for each of the core features are listed.

To give the elements of the model a meaningful categorization, the app store characteristic that is mainly influenced by each feature or policy was determined. This resulted in a categorization based on the following nine app store characteristics: app store usability to the user, app findability, app quality, developer quality, app store usability to the developer, app visibility, monetization potential, user interaction, openness. These characteristics could further be divided into user focused characteristics and developer focused characteristics. The accompanying feature and policy descriptions can be found in Tables 3 and 4. The models and the descriptions give a complete overview of the observed features and policies in the six case studies.

5 Analysis of the Results

In Figure 2 a sample of the table is found that lists the features and policies of the app stores under study. The main finding from these data, is that most of the app stores are on their way to include most, if not all features and policies in their app stores. Furthermore, due to the sheer size of the effort of implementing an app store we expect to see third parties offering white-label app store platforms[1], thereby reducing effort for platform owners in orchestrating the ecosystem.

[1] Such as the start-up www.appgalleries.com

User focused: app findability	
recommendations	Apps are recommended based on user profile
store curation tags	Developers can tag and categorize their apps
User focused: app quality	
app security integration	An app platform security system is provided
app security reporting	Harmful apps and security threats can be reported
app test driving	Apps can be test driven before purchase
content rating filter	Apps are rated with a content rating
device compatibility	Apps can be filtered on device compatibility
remote app remove	Harmful apps can be removed by owner from device
user review curation	Users can curate each other's reviews
User focused: app store usability	
automated refunds	Users can apply for refunds
developer refunds	Developer can initiate refunds
device integration	Devices have the app store installed by default
multi language	app store is internationalized
multichannel dist.	Users can use multiple channels to acquire apps
multi-currency	Multiple currencies are supported
update integration	Automated updates are possible for the app
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: feedback potential	
app suggestions	Users can leave suggestions for new apps
app support forums	Each app has its own support forum
beta testing mgmt	Developers can invite beta testers for their apps
feature suggestion	Users can suggest features to apps
issue tracking	Users can report issues and track their progress
user profile	Users have extended user profiles
Developer focused: monetization potential	
affiliate program	Users can make money directing "traffic" to apps
affiliate stores	Apps are offered through other channels
component offering	Developers can also offer separate components
discounts	Apps can be temporarily offered at a discount
in-app advertising	Monetization can also be done through advertising
in-app billing	Users can purchase extra features and content in-app
licensing integration	License checking for illegally installed or acquired apps
social media sharing	Apps can be shared through social media
subscriptions	Developers can offer content subscriptions to users
volume pricing	Developers can offer volume pricing
Developer focused: app store usability	
data API	Developers can get data from the app store using an API
deployment integration	Developers can automate the deployment to the app store
dev contract mgmt	Developers can manage contracts with the app store owner
dev multi-user login	Multiple users can be added to a developer account
dev sales statistics	Developers have access to sales statistics
geographic targeting	Apps can be targeted to geographic regions
tax support	The app store applies legally required taxation
Developer focused: visibility	
cross selling	Associated apps are shown to the developer
developer app list	A list of apps made by each developer is available
developer profile	Developers have profile pages with details

Table 3. User and developer centric app store features

Some other observations can be made, however, about the data. To begin with, the core features are supported by all app stores, which is one of the reasons why they made the short list. The data differs greatly, however, for all other features and policies. The largest outlier is Binpress (the source code app

User focused: app quality	
approval before publish	Apps are checked by the store owner for compliance manually
automated monitoring	The app store uses an automated system to check for apps that do not comply
code quality curation	The quality of the code of apps is checked
functional quality curation	The functional quality of apps is curated by the owne
interface quality curation	The owner checks apps for compliance with interface guidelines
review after purchase	Reviews for an app can only be posted by users that have downloaded or purchased the app
review poster verified	Users that are verified by the app store owner can post reviews
User focused: developer quality	
developer verification	Developers have to prove their identity to the app store owner before receiving payments
recurring fee	A recurring fee is required to be a developer at an app store
Developer focused: 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
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
revenue share	The percentage revenue share that goes to the developer
third party app stores	Apps are allowed to reference other app stores
third party in-app advertising	Apps are allowed to use third party in-app advertising
app store refunds	The app store owner has a clear refund policy and provides refunds on request of a user
third party in-app billing	Apps are allowed to use a third party system for in app purchases
Developer focused: openness	
competing functionality curation	Apps that have features that compete with the app store owner are not approved
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
Developer focused: visibility	
geographical availability	The number of countries an app store is available in

Table 4. User and developer centric app store policies

store), since it supports many different features and policies than the other app stores.

	Intel AppUp	Binpress	Amazon appst	Apple appstor	SlideMe	Google play
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

Fig. 2. Sample of the feature and policy evaluation for each app store case

At the time of writing only one app store (Amazon) out of six enables end-users to test drive an app before purchasing. Secondly, besides Binpress, all app stores include a content rating filter, since Binpress is the only source code app store, in comparison to the others. Only Google Play at this point supports automated developer and end-user refunds, suggesting that the Google Play billing system may be the most advanced, which we do not find surprising when considering Google's e-payment strategy and product portfolio.

In regards to feedback management, very little features are supported. Only one app store supports app suggestions, and only one other supports beta testing management. Yet another app store supports issue tracking and feature suggestions, which shows that developers could be supported much more extensively by the app store owners than they are now. In regards to monetization there is a varied set of app stores supporting the monetization features, and no generalizations can be made about those features at this time. It is interesting, however, to see that only one app store enables component (i.e., supporting app development) sharing.

When looking at the specific app stores it can be observed that Google Play and Amazon's app store are the most complete in supporting developers and end-users. SlideMe supplies the least features and could be considered the most immature. Surprising is that both in terms of developer features and end-user features the Apple app store is not the most mature, even though it is the 'benchmark' app store that significantly increased popularity of app stores.

6 Conclusions and further research

The objective for this study was to help app store owners gain more insight in the app stores by creating an app store definition and performing a multiple case study on app store features and policies. To achieve this objective, first, the definition of an app store was given. Section 4 shows that the common features observed in the case studies could be mapped to the different parts of this definition. Secondly, in section 2, the conceptual model features and policies were defined: features represent individual parts of the app stores software systems that the actors can interact with. Policies represent the rules, regulations and governing processes that limit the functional reach of the features. The description tables of these features and policies can be used to identify a feature or policy in an app store.

The app store definition combined with the feature and policy models provide app store owners or organizations considering becoming an app store owner with insight into the concept. One possible problem with the external validity might be the number of features and policies that were not applicable to the source code app store Binpress. In order to improve the external validity of the model for source code app stores it would be useful to add more case studies on source code marketplaces, or even classify app stores along their content, features, and policies. These explorations into domain specific app stores are seen as future work.

The possible relation between the strategy of app store owners and the app store characteristics is a topic for further research. Possibly, one could formulate sets of policies and features based on typical strategies followed by platform owners. However, for such an exploration, more insight into platform owner strategy and success is required. One challenge in such analysis is that a platform strategy, of which an app store is a small part, may be much more influential to the success of the platform than the app store.

References

1. C. Baldwin and J. Woodard. The architecture of platforms: A unified view. In A. Gawer, editor, *Platforms, Markets and Innovation*. Edward Elgar Pub, 2010.
2. B. Eaton, S. Elaluf-Calderwood, C. Sørensen, and Y. Yoo. Dynamic structures of control and generativity in digital ecosystem service innovation: the cases of the apple and google mobile app stores. *LSE, London Report*, 44(183):1–25, 2011.
3. P. Elmer-DeWitt. 6 ways iphone and android users differ. *Fortune. Feb*, 25, 2010.
4. A. Hagiu and J. Wright. Do you really want to be an ebay? *Harvard business review*, 2013.
5. A. Holzer and J. Ondrus. Mobile application market: A developers perspective. *Telematics and Informatics*, 28(1):22–31, 2011.
6. S. Hyrynsalmi, T. Mäkilä, A. Järvi, A. Suominen, M. Seppänen, and T. Knuutila. App store, marketplace, play! an analysis of multi-homing in mobile software ecosystems. *Proceedings of the International Workshop on Software Ecosystems*, page 59, 2012.
7. M. Iansiti and R. Levien. Strategy as ecology. *Harvard bus. rev.*, 82(3), 2004.
8. A. Idu, T. van de Zande, and S. Jansen. Multi-homing in the apple ecosystem: why and how developers target multiple apple app stores. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pages 122–128. ACM, 2011.
9. S. Jansen, A. Finkelstein, and S. Brinkkemper. A sense of community: A research agenda for software ecosystems. In *Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 187–190. IEEE, 2009.
10. J. West and M. Mace. Browsing as the killer app: Explaining the rapid success of apple's iphone. *Telecommunications Policy*, 34(5):270–286, 2010.
11. R. K. Yin. *Case Study Research: Design and Methods, Third Edition, Applied Social Research Methods Series, Vol 5*. Sage Publications, Inc, 3rd edition, December 2002.