Dynamics of Success for Early-Stage Mobile Game Startups

Maryam Roshan Kokabha
Dynamics of Success for Early-Stage Mobile Game Startups

Maryam Roshan Kokabha
Supervising professor
Professor Virpi Kristiina Tuunainen, Aalto University School of Business, Finland

Thesis advisor
Assistant Professor Riitta Hekkala, Aalto University School of Business, Finland

Preliminary examiners
Professor Kieran Conboy, National University of Ireland Galway, Ireland
Professor Tero Päivärinta, Luleå University of Technology, Sweden

Opponent
Professor Kieran Conboy, National University of Ireland Galway, Ireland
Mobile platforms offer an opportunity for creating and distributing apps through app stores. Many developers, in the form of startups, have joined these platforms with economic motivations. However, they face several challenges on the way to creating a business. Firstly, the startup nature of the businesses implies a scarcity of financial resources and a dearth of business experience in developing and selling apps. Secondly, the popularity of the platforms has resulted in fierce competition among apps, hence difficulty in bringing an app to the attention of users. Thirdly, the startups are compelled to operate within the app store structures under the market-making rules of the platform owner. Finally, the contextual factors related to the type of an app may create additional difficulties. For example, game developers must cope with the heterogeneity of users’ expectations, users’ reluctance to wait for a game to be fixed in the case of problems, and the mainstream freemium monetization model which makes revenue-making more complex.

This dissertation explains how early-stage mobile game startups operate in order to succeed. It uses the qualitative classic Grounded Theory Methodology for both analyzing and theory building. The data is collected through semi-structured interviews with 20 international game startups that were participating in a business accelerator – founded by a platform-owner – in Finland. The main research question asks ‘What are the dynamics of success for early-stage mobile game startups?’ The two more specific research questions ask how startups utilize the facilities of an accelerator, and how they excel in the market after publishing their games.

The theoretical contributions of the dissertation are threefold. Firstly, it contributes to the literature on the role of the platform owner in the activities of startups. It discusses the role of the accelerator as a social boundary resource in improving the capabilities of startups in terms of game design, access to resources, and improved market activities, as well as providing an opportunity for the startups to impact the platform owner’s decisions and to access resources from other members of the ecosystem. Secondly, it contributes to the literature on experimentation in software startups by explaining how mobile game startups follow an experimentation approach and use game analytics to discover areas of improvement and new potential markets. Thirdly, the developed theoretical model of ‘dynamics of success for early-stage mobile game startups’ explains that by utilizing the accelerator’s facilities and taking an experimentation approach, startups aimed to iteratively raise game quality, visibility and user engagement in the pursuit of initial success for future expansion to other platforms. Moreover, the findings provide practical implications for mobile game developers, platform owners and business accelerators.

Keywords Accelerator, App development, Boundary resource, Experimentation, Grounded theory, Mobile game success, Startup.
Acknowledgements

‘As you start to walk the way, the way appears.’ (Rumi, 1207-1273)

During my doctoral research, the journey of life has been full of adventures and ups and downs. I would like to dedicate the next few paragraphs to the wonderful people whose assistance and encouragements made this journey as pleasant as it was.

First and foremost, I would like to express my deepest gratitude to my supervisors Professor Virpi Kristiina Tuunainen and Assistant Professor Riitta Hekkala. Virpi has been a source of motivation, guidance and trust since the beginning of my doctoral research journey and has provided me with all the supporting facilities and freedom to explore the world of research. Her amazing personality, patience and priceless feedback created an inspiring environment for me to delve into academic research and build up my confidence as a researcher. I would also thank Riitta who inspired me in selecting my research methodology and supervised me to follow it properly. Riitta never stopped reminding me of the light at the end of the tunnel at a time when I was in a state of total confusion and doubt. In addition, I thank both my supervisors for co-authoring with me, a process through which I gained invaluable learning. Co-authorship was truly a joyful and professional experience with Virpi and Riitta.

I would like to sincerely thank my two pre-examiners, Professor Kieran Conboy from NUI Galway, and Professor Tero Pääväranta from Luleå University of Technology for taking the time to read my dissertation and share their insightful comments for further improvement of my work in future. I appreciate Professor Conboy for agreeing to act as my opponent in my public defense as well.

I would like to express my gratitude to all the AppCampus managers and staff for welcoming me warmly at their premises. Especially, I would like to thank Paolo Borella, Mike Bradshow and Chris Bouret for their time and knowledge in our discussions and interviews. I would sincerely appreciate all the startup interviewees for their generous sharing of information and thoughts. Conducting this research would have not been possible without their precious data.

I would like to thank my brilliant colleagues in Information Systems Science at Aalto University School of Business. My thanks to Professor Matti Rossi for his dedication in facilitating a fabulous and supportive research environment.
I would like to thank Dr. Antti Salovara for his great suggestion of the study context, and Dr. Wael Soliman for the very interesting debates that we had in crafting the earlier stages of my dissertation. I am grateful to Dr. Johanna Bragge for offering me the opportunity to act as a teaching assistant in her Management Information Systems course throughout the years of conducting this research. Many thanks to Dr. Negin Banafshehjahromi who quickly became a friend and shared tons of practical tips for finalizing my dissertation. I would also like to thank my peer doctoral students, Jussi Nykänen for his excellent comments on the first draft of my dissertation and Sanna Tiilikainen for sharing information on many topics throughout all these years. Also, I would like to thank Merja Mäkinen for all her administrative support.

Focusing on full-time research was solely possible by financial aids that I received from Marcus Wallenberg Foundation, Foundation for Economic Education, the HSE Foundation, and the Aalto University School of Business. I appreciate them all for their support.

I sincerely thank my precious friends, Dr. Kalevi Euro, Maria Leppälä, and Maria Appelberg for their practical and emotional support throughout all these years and for making Finland a second home for me. I appreciate Mikki Jahnukainen and Kaarlo von Freymann, my dear neighbors in Lauttasaari, who have energized me with their heartfelt kindness and interesting discussions. I would like to thank my dear friend, Dr. Akram Dehnokhalaji for all the useful tips about academic life. Furthermore, my most gratitude to my mentor - the Shams for my inner Rumi - Naser Mohammadi who skillfully broadened my horizon to reach the inner peace and become a much stronger person in accomplishing my goals. I also thank Naser for his marvelous painting on my dissertation cover.

Last but not least, I would like to thank my family for their unconditional love and support. Words cannot express my deep gratitude to my late father, my role-model in life, who raised me as a critical thinker and encouraged me to seek the highest levels of education, wisdom and humanity. I have written a considerable amount of this dissertation at his bed at the hospital while he was bravely yet silently fighting cancer. I wish I could share this moment with him as he always wished for me to pursue a doctoral degree; yet, at the bottom of my heart I feel his broad smile right now. My greatest appreciation and love to my mother, Dr. Fatemeh Ghorashi, a perfect example of a strong and successful woman, who along with my father, wisely provided me with the support, guidance and motivation to follow my dreams. Especially, her strength and dedication during my father’s illness allowed me to carry out my research. Finally, I would like to thank my beloved sisters, Mina and Setareh, and my dear brother-in-law, Nasir, for all their love and care, and for bearing so much of the family responsibilities while I was on this journey. I am also grateful to Mina for all our long-distance scientific discussions.

Helsinki, September 18th, 2018
Maryam Roshan Kokabha
Contents

List of Abbreviations and Symbols ............................................................. 1

1. Introduction ..................................................................................... 3
  1.1 Mobile Application Industry ...................................................... 3
  1.2 App Stores ................................................................................... 4
  1.3 Challenges for App Developers .................................................. 5
  1.4 Research Problem and Approach .............................................. 6
  1.5 Dissertation Outline ............................................................... 8

2. Research Methodology .................................................................... 9
  2.1 Research Paradigm of the Study ................................................ 9
  2.2 The Grounded Theory Methodology (GTM) ................................ 11
  2.3 The Glaserian (Classic) vs. Straussian GTM ............................ 12
  2.4 The Classic GTM Procedure ...................................................... 13
  2.5 Research Setting ....................................................................... 14
  2.6 Data Collection ......................................................................... 16
  2.7 Literature Review Procedure .................................................... 21
  2.8 Data Analysis Procedure .......................................................... 22
    2.8.1 Open Coding ................................................................. 22
    2.8.2 Selective Coding ............................................................ 23
    2.8.3 Theoretical Coding ......................................................... 25
  2.9 Quality of the GTM Research ................................................... 25

3. Literature Review .......................................................................... 27
  3.1 Preliminary Literature Review .................................................. 27
    3.1.1 Mobile Application Development on Digital Platforms ......... 27
    3.1.2 Software Startups ........................................................... 33
    3.1.3 Business Accelerators ..................................................... 35
  3.2 Extended Literature Review .................................................... 39
    3.2.1 Boundary Resources ....................................................... 39
    3.2.2 Iterative Approach in Software Startups ............................ 41
    3.2.3 Success of Mobile Games ................................................ 43

4. Utilizing the Accelerator’s Facilities ............................................. 49
## List of Abbreviations and Symbols

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>application programming interface</td>
</tr>
<tr>
<td>App</td>
<td>application</td>
</tr>
<tr>
<td>BML</td>
<td>build-measure-learn</td>
</tr>
<tr>
<td>GT</td>
<td>grounded theory</td>
</tr>
<tr>
<td>GTM</td>
<td>grounded theory method</td>
</tr>
<tr>
<td>IS</td>
<td>information systems</td>
</tr>
<tr>
<td>MAAC</td>
<td>mobile app acceleration camp</td>
</tr>
<tr>
<td>MVP</td>
<td>minimum viable product</td>
</tr>
<tr>
<td>RQ</td>
<td>research question</td>
</tr>
<tr>
<td>SDK</td>
<td>software development kit</td>
</tr>
<tr>
<td>WOM</td>
<td>word-of-mouth</td>
</tr>
<tr>
<td>WP</td>
<td>Windows phone</td>
</tr>
</tbody>
</table>
1. Introduction

In this opening chapter, I present an introduction to the mobile application (app) industry, the concept of app stores, and the challenges facing mobile app developers. Then, I introduce the research problem and methodology, and finally close the chapter with an outline of the dissertation.

1.1 Mobile Application Industry

Alongside the widespread use of the Internet and the digital connectivity in businesses and among people, technological devices themselves have become more advanced and pervasive. Smart devices have been introduced to the market with modular architectures that allow the creation of additional services and content on them. Among the smart devices are the widely accepted smartphones, which allow both data and voice communications. In addition to attracting mobile phone users, smartphones have encouraged providers of data-driven services to get on board to make financial benefits (West and Mace, 2010). They offer various software applications (or apps) to be installed on smartphones. These apps help the users to perform specific tasks on their phones, such as browsing the web, listening to music, watching videos, and playing games.

Until about 10 years ago, the major players on the supply side of application services were mobile operators who had a strong relation with mobile manufacturers and a strict control on mobile services (Holzer and Ondrus, 2011). Application developers had to go through the operators’ channels to offer their applications to users. Thus, a limited number of developers were able to get approved by the operators and offer applications for mobile devices (Bergvall-Kåreborn and Howcroft, 2013a).

However, in 2008, Apple opened up its iOS platform code sources and allowed developers to create apps for its iOS-based devices (i.e. iPhone, iPod and iPad). Consequently, developers started to work with Apple in an arm’s length revenue-sharing contract (Ghazawneh and Henfridsson, 2013) where Apple took 30 percent of the revenue made by their apps. Within this setting, app developers could use the Software development Kits (SDKs) and Application Programming Interfaces (APIs) provided by Apple to develop apps. They could distribute their apps directly to users in a digital marketplace called the App Store. Mobile users could search and download apps from the App Store to their mobile devices seamlessly. Apple’s initiative transformed the mobile service industry and soon after, other large software companies and mobile manufacturers followed a similar model, either on their own or in partnerships. Among them were Google, Research in Motion, Nokia, and later Microsoft and Amazon. However, some of these companies ceased operation or stayed at a small scale, and to the date of writing this dissertation (2018), the market is mainly dominated by the Apple iOS and Google Android platforms.
The transformation of the mobile industry resulted in the evolution of the app economy, comprising all the economic activities related to the creation, management and distribution of mobile apps. It created new opportunities for app developers and entrepreneurs by allowing them to access the global markets (Tiarawut, 2013). In recent statistics, the global revenue made from the purchase of apps on the major platforms of Apple and Google was 60 billion dollars in 2017, a 35 percent increase compared with the 2016 figure (Perez, 2018). The revenue is expected to reach 189 billion dollars by 2020 (Dogtiev, 2018). At the heart of the app economy sit the app stores.

1.2 App Stores

App stores work as the interface between the platform owner and all the other actors of the ecosystem shaped around the platform, i.e. developers, users, advertisers, and other service providers (Ghazawneh and Mansour, 2015). Each mobile platform has its own app store, such as the iOS App Store, Android Google Play, and Microsoft Store (formerly Windows Store). In addition, some independent firms may create their own app stores on a platform. For instance, the Amazon Appstore has been created on Android to provide apps for Amazon Kindle Fire devices.

The apps offered on these stores have various attributes in terms of types and pricing. They have different types such as games, health apps, utility apps, educational apps. Game apps comprise the largest category on major platforms (Merikivi, Tuunainen and Nguyen, 2017). The apps have different monetization models, mainly offered as paid or free (Liu, Au and Choi, 2012). Paid apps are sold at a price at the time of download. Free apps are offered at no price at the time of download but may encourage users to purchase upgrades and add more features to the app – a model referred to as freemium or in-app-purchase. In addition, some apps may make revenue by showing advertisements inside them, referred to as in-app-advertisements. The same app may have different prices or monetization models on different platforms, decided by its developer (Bergvall-Kåreborn and Howcroft, 2011).

Mobile users can find apps on the app stores by using the ranking lists and/or the app store search engine. In addition, they can see apps that are highlighted in one way or another by the app store staff in the form of featured apps. Examples of featured apps captions used in the iOS App Store include ‘new games we love’ and ‘what we are playing’.

Despite the existence of different app stores, their structures are rather similar. Every app store has several ranking lists where apps are listed according to certain aspects, such as top free, top paid, popular, and top-grossing. Users can see the apps on each list, but the screen sizes of mobile phones allow only a few apps to be seen without the need for scrolling. Consequently, app developers would like to have their apps in the top places in these ranking lists for more visibility (Bergvall-Kåreborn, Howcroft and Chincholle, 2010). The algorithms used to locate apps on top-ranking lists have not been officially announced by any of the platform owners to date (e.g. Yi, Lee and Kim, 2017). However, developers agree on the importance of factors such as the volume of recent and aggregated downloads and revenue, frequency of app usage, number and value of reviews by users, keyword relevance, and updates (Duarte and Picoto, 2016).

Aside from the platform owner and app developers, other types of companies have complementary roles on mobile platforms (Basole, 2009). Among them are the providers of advertisement networks that supply the advertisements embedded in the apps (e.g. AdMob or AdDuplex). Some others provide cross-promotional networks to help developers promote
their apps (e.g. AdDuplex). Some companies have created a pool of users and promote apps to them (e.g. My App Free). Other types of companies help with cross-platform development tools and environments (e.g. the Unity game engine). Such services help developers in reducing their compatibility challenges when they intend to offer their apps on multiple platforms. Moreover, some companies provide analytics services so that developers can trace the performance of their apps on the market (e.g. Google Analytics). In spite of all these services, the platform business setting creates some challenges for developers as well.

1.3 Challenges for App Developers

Despite the low barrier to entry and existence of the opportunity for job creation and entrepreneurship in the domain of app development, the markets are quite complex for the app developers. The notion of negative same side externalities is valid on the platforms, by which the existence of a large number of developers on the supply side of the market results in intense competition among them (Rochet and Tirole, 2003). Consequently, it is very difficult to make an app stand out among the crowd.

Moreover, developers suffer from a lack of power in the face of the platform owners (Bergvall-Kåreborn and Howcroft, 2011; Bresnahan, Orsini and Yin, 2014). The platform owners have a centralized control and gatekeeping role (Qiu, Gopal and Hann, 2017) and hesitate to communicate openly about their strategies with developers (Bergvall-Kåreborn and Howcroft, 2014). Through the app store structures, platform owners have become the market-makers by restricting the developers to following their rules (Qiu et al., 2017). For example, the decision to feature an app as well as the design of the ranking lists and the mechanism for getting on the lists are simply decided by the platform owners. Given that the number of downloads may affect the position of an app on these lists, the apps need to already be receiving a large number of downloads before being positioned on these lists. Therefore, initial success leads to more success, and if developers fail to succeed in the first place, they may lose their chance of success in future. Further, research shows the incapability of the ranking lists to capture the target users for a specific app (Bresnahan et al., 2014). Thus, the market structure works more in favor of established firms with sufficient budgets that can invest in marketing their apps to their target users, compared with young firms with tight budgets.

In addition to the challenges imposed by the market structure and the intensive competition, app developers may face some challenges in their tasks in developing and publishing their apps. Unlike traditional software development roles, app developers’ responsibilities are not limited to the technical development phase, but include the full life-cycle of the app, from concept generation to technical development to sales and customer relationship activities (Bergvall-Kåreborn et al., 2010; Bergvall-Kåreborn and Howcroft, 2011). Consequently, app developers have a broad range of responsibilities including making revenue. All these activities are taking place in an intensively competitive market with a mainstream trend of offering freemium apps. Consequently, most app developers make little or no financial revenue from their apps (Bergvall-Kåreborn and Howcroft, 2013b; Lim and Bentley, 2012).

Among the app developers, mobile game developers face some context-specific challenges as well, due to the nature of their product and the more intense competition in the game category compared with other apps (Yin, Davis and Muzyrya, 2014). On the development side, game development has a lower barrier to entry compared with other types of apps because of the existence of vast numbers of development tools and libraries. This results in many devel-
opers entering the platforms; hence the negative same side externalities result in fierce competition among games. The level of competition creates problems even for well-established companies. For example, the creator of the famous Angry Birds game, Rovio, lost as much as 50 percent of its share value in early 2018 after not being able to meet its expected profit (Ram, 2018). This may result in different market strategies for game developers compared with other types of app developers (Davis, Muzrya and Yin, 2014). On the use side, game players are more heterogeneous in terms of demands and expectations. While a non-game app may serve a specific purpose with clear specifications, games serve the purpose of entertainment and enjoyment (Okazaki, Skapa and Grande, 2008), which may have a different meaning for different users. In fact, the demands of the game players are unpredictable and users may not be able to define exactly what they want in advance (Koskenvoima and Mäntymäki, 2015). In addition, the existence of many alternative games results in the impatience of game players for a game to be fixed if there are existing problems (Yin et al., 2014). Meanwhile, games have the highest chance of attracting users and getting on top-ranking lists immediately upon their release time (Yi et al., 2017). Therefore, creating a game with appeal to the mass market and the ability to not only attract users, but also to encourage them to continue playing, is critical for game developers.

1.4 Research Problem and Approach

Many app developer firms are independent startups (Qiu et al., 2017). The commonly agreed characteristics of a startup include having limited resources in terms of knowledge, finance, and network, and struggling to promptly react and adapt to market changes, and finally to scale up (Paternoster et al., 2014). Startups face challenges in defining their product, market, and business strategy for making a profit, as well as team working and operating inside the firm in order to scale-up before running out of money (Giardino, Bajwa, Wang and Abrahamsson, 2015). App developer startups face some additional challenges related to the platform-based business environment, as explained in the previous section, such as issues related to diversity in handling multiple markets and customers, lack of knowledge about the business and the platform strategies, the market structure under the heavy control of the platform owners, and the intense competition (Bergvall-Kåreborn and Howcroft, 2014).

In general, startups use various strategies to cope with scarce resources and market challenges. In recent years, a popular initiative has been to participate in business accelerators’ programs to expedite their market adaptation and accelerate their growth (Cohen, 2013). Accelerators were first introduced in 2005 in the US and can now be found around the world. They recruit a cohort of startups through open applications, and over a short period of time, provide them with seed funding, intensive training, mentorship, and networking opportunities (Cohen and Hochberg, 2014). The accelerators can be either independent (and benefit from an equity stake in the startups) or be owned by a company with the aim to create an ecosystem for the company (Pauwels, Clarysse, Wright and Van Hove, 2016). Previous studies show some positive impacts of accelerators on startups, including getting access to capital, customers, experience, and networks (e.g. Hoffman and Radojevich-Kelley, 2012; Kohler, 2016). However, the existing literature does not provide an in-depth understanding of how the accelerator’s facilities are utilized by the startups to support their activities and strengthen their position in the competitive market.

To cope with the uncertainties in the market, startups may also take an experimenting approach based on lean production principles (Ries, 2011). This method suggests startups de-
velop products by getting user feedback along the development process to avoid investing in a product with little or no market appeal. Accordingly, the startups decide whether to stay in their existing direction or pivot to a new one. Some recent studies have provided practical guidelines according to the lean principles in the form of experimentation for startups to help them find a product-market fit (e.g. Fagerholm, Sanchez Guinea, Mäenpää and Münch, 2017). However, these studies have not considered the platform-based environment of the business with its highly competitive setting.

In the platform setting, games need to take off fast to get a large number of downloads in a very short period of visibility due to the intense competition on the app stores (Bergvall-Kåreborn and Howcroft, 2013b). In this context, startups gain access to their users only after publishing the app on the platform (Bergvall-Kåreborn and Howcroft, 2013a), and that coincides with the time of the highest chance of receiving publicity and consequently, the app getting downloaded (Yi et al., 2017). Therefore, defining what needs to be offered at the launch time so as not to lose the already attracted users, and what can be experimented with and adapted to the market becomes critical for the app developers.

Despite the economic importance of app developer startups, their activities are not yet comprehensively understood in the literature. The extant literature discusses the challenges of app developers and the factors that can lead to the success of mobile apps (e.g. Bergvall-Kåreborn and Howcroft, 2014; Bresnahan et al. 2014; Li, Goh and Cavusoglu, 2013; Qiu et al., 2017). However, it does not provide a comprehensive picture of how app developers actually work in coping with challenges. I aim to shed light on the activities of mobile app developer startups by targeting those working on developing games. Particularly, my focus is on early-stage startups that are working on developing and publishing their first commercial games on a platform, because they have just entered the field of commercial app development and are more likely to face problems due to their lack of business experience.

To serve this purpose, I used a qualitative classic grounded theory method (GTM), both as the method of data analysis and as the technique for theorizing (Glaser, 1992). The research data is made up of interviews with 20 international mobile game startups (named alphabetically) from 13 countries. They were working on their first commercial game on the platform. My aim is to understand how game startups operate on mobile app platforms to develop and publish their games in their early stage of business. Applying the classic GTM resulted in the emergence of two core categories from the empirical data. Accordingly, I defined two specific research questions that each address the findings of one of the core categories.

**RQ 1:** How do early-stage mobile game startups utilize the accelerator’s facilities in developing and publishing their games?

**RQ 2:** How do early-stage mobile game startups excel in the market?

In the final stage of the study, I related the two core categories and scaled up to a higher level of abstraction to build the core theme and theorize it. Accordingly, I finalized my main research question as follows.

**What are the dynamics of success for early-stage mobile game startups?**
1.5 Dissertation Outline

I have organized this dissertation in seven chapters. In Chapter 1, I present an overview of the mobile application industry, the concept of app stores, and the challenges facing app developers including game developers, and finally present the research questions and approach.

In Chapter 2, I outline an overview of the research paradigm and methodology. I explain the research setting and the data collection and analysis procedures in detail.

In Chapter 3, I present the literature review. It consists of both a preliminary and extended literature review following the GTM. The preliminary literature review (Section 3.1) includes the streams of literature on mobile app development, software startups, and business accelerators. The extended literature review (Section 3.2) includes an in-depth review of the literature related to boundary resources, the iterative approach in software startups, and the success of mobile games.

Chapters 4 and 5 contain the findings of the study in the two emergent core categories. In Chapter 4, I discuss the first core category, ‘utilizing the accelerator’s facilities.’ In Chapter 5, I discuss the second core category, ‘ Excelling in the market.’

In Chapter 6, I explain the creation of the grounded theory in the form of a theoretical model by integrating the two core categories and scaling up.

Finally, in Chapter 7, I present the summary and conclusions of the study by portraying the theoretical and practical contributions as well as the limitations of the study and avenues for future research.
In this chapter, I explain the details of my research methodology. I will provide an overview of research paradigms and my research philosophy. Subsequently, I will explain in detail the Grounded Theory Methodology, how I have applied it, and how I have met the research quality criteria.

2. Research Methodology

The research paradigm characterizes the researcher's assumptions toward 'the nature of the world, the individual's place in it, and the range of possible relationships to the world and its parts' (Guba and Lincoln, 1994, p. 107). In other words, the paradigm sets the researcher's approach to the ontology, epistemology and methodology (Healy and Perry, 2000; Guba and Lincoln, 1994). The **ontology** explains the researcher's point of view about the nature of reality, in terms of whether the world exists objectively and independent of humans, or subjectively and made by humans' actions. **Epistemology** indicates the relationship between the researcher and reality, and the criteria used for the creation and evaluation of knowledge. The **methodology** is the appropriate tools and techniques that the researcher uses to understand reality and create acceptable evidence. (Healy and Perry, 2000; Orlikowski and Baroudi, 1991.)

The three main paradigms in the field of information systems are positivist, interpretivist, and critical. Positivism is rooted in the natural sciences and believes in the existence of an objective reality independent of humans. (Guba and Lincoln, 1994; Orlikowski and Baroudi, 1991.) Positivist researchers just observe the phenomenon under investigation without having any effect on it (Healy and Perry, 2000), and their role involves discovering the existing reality (Guba and Lincoln, 1994; Orlikowski and Baroudi, 1991). The phenomenon is objectively given and is completely separate from the researcher and the research tools (Myers, 1997; Orlikowski and Baroudi, 1991). Positivist researchers aim at testing a theory in order to draw conclusions from a representative sample to the stated population (Klein and Myers, 1999) by creating hypotheses for a predictive understanding of the phenomena (Orlikowski and Baroudi, 1991).

Interpretivism has its roots in ethnographic research in anthropology (Walsham, 1995). In contrast to positivism, interpretivist researchers believe in the social construction of reality and find it impossible to separate social reality from the human actors who create it (Orlikowski and Baroudi, 1991; Walsham, 1995). To understand social reality, the interpretivist researcher uses social constructions to rely on the research participants' sense-making on the phenomena (Myers, 1997). The subjectivity of reality in the interpretivist research demands that the researcher considers various viewpoints of the participants and interprets reality by
Creating a relatively shared understanding with the participants about the phenomenon (Orlikowski and Baroudi, 1991). So, for the matter of sense-making between the researcher and the participants’ constructed world, the researcher engages with the participants to understand and explain a social phenomenon by using qualitative data (e.g. interviews, field notes, documents) (Myers, 1997).

Critical research considers social reality to be historically constituted and to be produced and reproduced by humans throughout history (Guba and Lincoln, 1994). Since historical and environmental factors have shaped the social realities, the researcher tries to critically evaluate the reality that is being researched and to transform it, instead of predicting or explaining it, as in positivist or interpretivist research (Orlikowski and Baroudi, 1991). The researcher is interactively involved in the phenomenon under study and attempts to challenge the status quo and transform it according to their values (Guba and Lincoln, 1994; Orlikowski and Baroudi, 1991). Therefore, the critical paradigm requires research methods that are of a longitudinal and dialectical nature (Guba and Lincoln 1994; Healy and Perry 2000; Orlikowski and Baroudi 1991).

The reactions of scholars to the existence of the above three main schools of thought have categorized them into three groups: imperialists, isolationists, and pluralists (Mingers, 2004). Imperialists argue for the dominance of a single paradigm. Isolationists accept the existence of the different paradigms, but as distinctively separate and independent of each other. Pluralists welcome the diversity of paradigms, either in accepting the existence of different paradigms, or the combination of the paradigms to shape trans-paradigmatic research methods. Arguing that life includes both objective and subjective characteristics, Weber (2004) discusses that the differences between the paradigms, particularly the main two competing ones of positivism and interpretivism, do not lie substantively at their metatheoretical level, and the rhetoric to win with one of these paradigms brings no use to the scientific world. Instead, he argues that both paradigms aim to improve our knowledge of some phenomenon, and according to the purpose of the research, the researcher should choose the appropriate research method and analysis techniques, considering their strengths and weaknesses.

In my paradigmatic views on life, I follow Weber's (2004) argumentation, and believe in the existence of both subjective and objective realities. As he exemplifies, the principle of the Earth’s gravity is an objective matter and humans cannot live with their own interpretation of it. However, for example, decision-making about an issue follows a subjective understanding by the decision-maker about that issue. Whether the decision's consequences are supported by the realities or not, still the decision is made subjectively and through the sense-making of the decision-maker. Therefore, I, as a researcher follow a pluralist approach and do not emphasize a specific paradigm in a general sense. I would rather consider the phenomenon under study and accordingly take a suitable paradigm and research method which allow me to understand it in my best capacity and serve my research purpose.

In this particular study, by choosing the grounded theory methodology (GTM) my emphasis was on the strengths of this methodology in coding and theory-building by the empirical data (Urquhart, 2013, p. 32) where existing theories do not exist. Therefore, I allowed the empirical data to talk rather than imposing a predefined framework on them. GTM in its nature does not have any ‘identifiable inherent philosophy’ (Urquhart, 2013, p. 59). However, applying it in this study, I focused on sense-making by the qualitative data and let the findings emerge without deploying any pre-set dependent or independent variables, as in an interpretivist study (Klein and Myers, 1999).
I attempted to understand the phenomenon of the early-stage startups’ developing and publishing activities through the comprehending and sense-making of the startups about their world (Orlikowski and Baroudi, 1991). Choosing a qualitative grounded theory method and collecting rich data through interviews facilitated this process. During the interviews, I became closer to the world of the startups and to understanding their concern, challenges and resolutions. I let them openly share their ideas, resulting in the emergence of new themes and discussion points to create a shared understanding of the phenomenon. I visited their premises during the accelerator program and observed most of the activities taking place through trainings and other interactions between peer-startups as well as between startups and accelerator staff. When possible, I had informal talks with the participants and accelerator staff to familiarize myself with their world to a greater extent. Alongside the coding process, I was also careful to view the data with an open mind without any pre-assumptions and to interpret what my data revealed (Orlikowski and Baroudi, 1991).

2.2 The Grounded Theory Methodology (GTM)

In this study, I use the GTM for both analyzing the data and theory-building. Developed in the field of sociology, the GTM aims at ‘discovery of theory from data systematically obtained from social research’ (Glaser and Strauss, 1967, p. 1). The discovery refers to a theoretical conceptualization which is done through a rigorous and systematic process of data collection and analysis (Urquhart and Fernández, 2013). In fact, the GTM is a package of research methods for collecting, coding, and analyzing the data (Glaser, 1998, p. 12) and establishes ‘the infrastructure, the skeleton and a process by which data is gathered and analyzed’ (Birks, Fernández, Levina and Nasirin, 2013). The GTM heavily emphasizes theory-building. It does not impose any priori theoretical idea or prior knowledge to the data, but instead allows the theory to emerge from the empirical data (Urquhart, Lehmann and Myers, 2010).

The main characteristics of the GTM are defined as allowing for the emergence of the theoretical concepts through theoretical sampling and constant comparison that leads to saturation of the concepts (Birks et al., 2013; Glaser and Strauss, 1967; Urquhart and Fernández, 2013). Theoretical sampling refers to the process of data collection where the emerging theory guides the researcher to know where to collect the next data and continues until the saturation of the theory, instead of using a defined sample size (Glaser, 1992). In other words, the researcher collects slices of data and analyzes them (Glaser and Strauss, 1967, p. 65). Theoretical sampling enables the researcher to justify the emerging concepts by showing more evidence for them (Urquhart, 2013, p. 8), and is accompanied by constant comparison. Constant comparison is the iterative process of data analysis where each incident found in the slices of data is compared with other existing incidents for similarities and differences, as well as with the emerged concepts, to see if it enriches them (Glaser, 1992). The constant comparison continues until saturation, which means when the researcher starts to see ‘similar instances over and over again’ (Glaser and Strauss, 1967, p. 65).

The GTM is a suitable method of research to study emerging areas and phenomena where there is a scarcity of theoretical understanding (Orlikowski, 1993). By using the GTM, the researcher discovers what is actually happening in the data rather than what should happen (Urquhart and Fernández, 2013; Glaser, 1978, p. 159). In the context of this study, the GTM is very useful due to the dearth of research on understanding how early-stage mobile game startups operate on the platforms (e.g. see Ghazawneh and Henfridsson, 2013) and to discov-
er more insights into the role of the platform owner (e.g. see Qiu et al., 2017) and accelerator (e.g. see Mian et al., 2016) by their impact on the activities of startups.

Moreover, the GTM has been used as an appropriate method of research in the field of information systems (Lehmann, 2001; Levina, 2005; Orlikowski and Iacono, 2001), either alone or in a mixed method such as combined with case studies (Matavire and Brown, 2013). The GTM has been used in two distinct forms: 1) as a coding method to make sense of a large amount of data, or 2) as a method of analysis and theorizing to build a theory from the empirical data (Mattarelli, Bertolotti and Macri, 2013; Urquhart et al., 2010). Most IS researchers have used the GTM for data analysis without aiming for theory-building, and sadly some researchers have mislabeled their research as using GTM without actually following its principles (Matavire and Brown, 2013; Urquhart and Fernández, 2013). However, the GTM has much more to offer when it is used as a theory-building approach, particularly when it allows the researcher to point to much empirical evidence that has been used to build the consequent theory (Urquhart et al., 2010).

2.3 The Glaserian (Classic) vs. Straussian GTM

Since the introduction of the GTM, it has been progressed in two distinct directions by its co-founders and their followers – referred to as Glaserian (aka classic) versus Straussian. The deviation started when Strauss and Corbin (1990) tried to provide more guidelines for conducting a GTM. This caused the negative reaction from Glaser (1992), who believed that the guidelines were endangering the essence of the GTM by preventing the data from speaking for itself any more (Glaser, 1992, p. 123).

The versions have some fundamental differences in various aspects (Matavire and Brown, 2013; Niekerk and Campus, 2009). Overall, the Glaserian version allows for more flexibility and parsimony, while the Straussian version offers more operational procedures (Locke, 2001). The classic GTM relies totally on the data and what emerges from the data, and strictly prohibits the use of any pre-conceptions, assumptions, or priori theory coming from either the researcher’s experiences or from the literature. It does not require the researcher to have a specific research question in mind before entering the field, but it is enough to have a concern or a research problem. In contrast, in the Straussian GTM, the researcher must start with a research question. It requires the researcher to think of other possible aspects that can bring a broader view to the data. Therefore, the researcher should ask questions based on their experiences and/or the existing literature and verify them with the data. As a matter of fact, in the Straussian GTM, a priori theory can be used as a sensitizing device. While the classic GTM is fully based on induction, the Straussian GTM is based on both induction and deduction.

The two versions are also different in their coding process. The classic GTM uses three phases of open coding, selective coding, and theoretical coding. The Straussian GTM uses three phases of open coding, axial coding, and selective coding. Even though they share the same names in some phases, they are fundamentally different in essence. Furthermore, the classic GTM allows for much more flexibility in theorizing by offering many different ways of relating the concepts. Glaser (1978) provides 18 different families of codes that the researcher can use to relate the codes to each other, and emphasizes the flexibility in doing so (Glaser, 1992). In contrast, the Straussian GTM recommends the use of one model which is based on causal relationships.
Given the fundamental differences between the two methods, researchers must decide which method they want to follow from early on (Niekerk and Campus, 2009). I decided to follow the classic GTM, for two reasons. First, I entered the research field without having a clear research question in mind; instead, I had a broad research question and concern in regard to understanding how the early-stage startups operate in the intensely competitive app markets. I was interested in understanding their concerns and their tactics and strategies, rather than imposing any personal thoughts and assumptions on the data. In fact, the rich data collected through the interviews offered a golden opportunity to have ample data to find out what was happening (Glaser 1992; 1998). Second, I found the flexibility and stronger attention to the conceptualization in the classic GTM, more appealing because I had the freedom to relate the concepts in any way that made sense to create a theory grounded in the data. The classic GTM gave me the freedom to conceptualize creatively (Niekerk and Campus, 2009) and to avoid imposing any existing knowledge on my data analysis, but rather be fully original.

2.4 The Classic GTM Procedure

I used the classic GTM in the ‘full’ version for the purpose of both data analysis and theorizing (Matavire and Brown, 2013). I followed the recommendations of Glaser (1992) to take an open approach and allow the concepts to genuinely arise from the data. Meanwhile, I also conducted a preliminary literature review to permit theoretical sensitivity (Urquhart and Fernández, 2013).

The coding procedure has three phases: open coding, selective coding, and theoretical coding, which is accompanied by a process of constant comparison and memo-writing (Glaser, 1992; 1998; 2004; 2007). Open coding refers to ‘coding the data in every way possible’ (Glaser, 1978, p. 56). Selective coding is delimiting the coding to the concepts that relate to the core categories (Glaser, 1978, pp. 61-72). In other words, the open codes are organized into selective codes which will then contribute to the creation of the core categories (Urquhart, 2013, p. 49). Theoretical coding is conceptualizing how the selective codes relate to each other, to be eventually integrated with the literature (Glaser, 1978, p. 72). Glaser suggests 18 families for how codes can relate to each other, among which I followed the ‘model’ family (Glaser, 1978, p. 81). The models depict the core categories (or core variables) (Glaser, 1978, p. 61), which are built by the selective codes and their relationships. A study can have a few core categories, but it is advised to have only one or two (Urquhart, 2013, p. 116; Glaser, 1978, p. 94). During the coding process, through constant comparison and memo-writing (Glaser, 1978, p. 83), the researcher enriches the emerged concepts (Urquhart, 2013). The memos can be written in a free format (Glaser, 1978, p. 85) from a few words to more complicated diagrams. By the iterative process of coding, constant comparison, memo-writing, and theoretical sampling, the researcher tries to saturate the core categories (Glaser, 1978, p. 95).

The saturation of the core categories results in the creation of a substantive and, further on, a formal theory. The core category with its selective codes and their relationships creates a substantive theory, which is a ‘theory developed for a substantive or empirical area’ (Glaser, 1978, p. 144). The substantive theory is backed up with many incidents of empirical data; thus, it is quite rich and detailed (Urquhart, 2013, p. 30). Subsequently, the researcher scales up the substantive theory to take it to a higher level of abstraction (Urquhart, 2013, p. 130). In doing so, the researcher groups the core categories and relates them together with more
abstract concepts and themes (Urquhart, 2013, p. 31). Hence, the substantive theory is lifted to a formal theory, which is ‘developed for a formal or conceptual area’ (Glaser 1978, p. 144). Both substantive and formal theories can be referred to as ‘middle-range’ theories, falling somewhere between the ‘all-inclusive’ grand theories, and the ‘minor working hypotheses’ of everyday life Glaser (1978, p. 144). There are several ways to increase the scope and generalizability of the developed theory, such as 1) rewriting the phrases and constructs in a more abstract language than the substantively specific words, 2) comparing the developed theory with other same-level theories from the literature, and/or 3) including more data to broaden the scope of the theory (Glaser, 1978, pp. 145-146).

2.5 Research Setting

I collected the research data through interviews with early-stage mobile game startups who were participating in the AppCampus training program in Finland.

AppCampus was a three-year (May 2012-May 2015) seed-accelerator program for attracting mobile app developers to the Windows Phone (WP) platform and making them into successful entrepreneurs. It was a partnership project between Microsoft (and formerly Nokia) investing 18 million euro, and Aalto University investing 3 million euro in the operation. The funding for each app ranged from 20,000 to 70,000 euro depending on the level of the app’s complexity as decided by the AppCampus staff. The funding was provided in exchange for 90 days exclusivity of the app on the WP Store. The initial objective of AppCampus was to populate the WP platform. So, its early concentration was to provide funding for novice app developer startups who had innovative apps which used key features of the WP platform (such as Nokia Maps, in-app-purchase, push notification).

Over time, with the full acquisition of the Nokia phone by Microsoft, and with more developers joining the platform, the focus of AppCampus shifted toward funding those apps with the potential for mass market appeal and a higher chance of accomplishing the performance targets set by AppCampus for download volume, revenue volume, and user star-rating (see Table 1). Accordingly, AppCampus shifted away from its initial goal of populating the platform with junior startups and moved toward selecting startups with some previous experience in developing apps and services (even as hobby development), who could show competence in developing apps with potential. However, these companies still needed to offer an app with some innovation; for example, the app should not exist on other platforms or if it did, the AppCampus app should offer some differentiated features for the WP platform, and still use the key features of the WP platform.

Table 1. AppCampus success criteria for apps.

<table>
<thead>
<tr>
<th></th>
<th>Successful app</th>
<th>Hero app</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume of downloads in a year</td>
<td>150,000</td>
<td>500,000</td>
</tr>
<tr>
<td>Revenue in a year</td>
<td>10,000</td>
<td>50,000</td>
</tr>
<tr>
<td>User star-rating</td>
<td>&gt; 4.3</td>
<td>&gt; 4.3</td>
</tr>
</tbody>
</table>

With the shift of focus in AppCampus, the apply process and the process for selection of the apps were changed as well. Originally, applying were open to the public on the AppCampus website, and selection was done only by the AppCampus staff. Later, AppCampus decentralized its application operation to its local partners across the world, such as Microsoft and Nokia offices and local accelerators. This modification was due to several factors such as 1) AppCampus’s gain in popularity and the increasing number of applications, 2) the require-
ment to have a stricter selection process due to the change of focus toward potentially successful apps, and 3) the strategy of Microsoft to provide more training opportunities for its app developers across the globe through its local partners.

The new process of applying and selection worked through AppCampus partners in each country filtering the startups. The local partner screened the applications and selected ten startups to participate in a two-day Mobile App Acceleration Camp event (MAAC). The selection criteria were mainly around the app having a good design, value, and features. So, the startups were required to provide a prototype to meet these criteria rather than having barely an idea. During the MAAC events, the startups went through pitching and mentoring sessions with coaches. Those approved by the MAAC staff were allowed to apply to AppCampus.

After being selected for AppCampus, the startups went through a four-phase process: 1) vetting, 2) design, 3) release, and 4) post mortem. In the vetting phase, the startups did the administrative formalities (i.e. filling out the payment information form, signing the AppCampus terms and conditions, and registering with the WP developer center). In the design phase, they submitted their project definition including the target users and structure of the app, user interface components, and navigation in the app. During this phase, the project definition was reviewed by the AppCampus Quality Assurance (QA) team and if approved, the startups received 30 percent of their funding. In the release phase, the startups released the app to the WP store in a hidden format available to selected users to get QA approval for store certification. After receiving the QA certification, the developers could publish the app publicly. During this phase, the startups received the remaining 70 percent of the funding, and the 90-day exclusivity period started. In the post mortem phase, AppCampus provided more support for the startups to help with gaining visibility and promotion for the apps.

In addition to the funding of the selected apps, AppCampus invited some of the startups with the most promising apps to participate in AppCademy. AppCademy was an intensive acceleration program held on the Aalto University premises in Finland. A dedicated AppCampus space was established that provided shared working facilities, private meeting rooms, and a lecture hall. Initially, AppCademy was a four-week program to train the teams on app development and broader entrepreneurship areas such as design, development, pitching, monetizing, marketing, communication, branding, and positioning of the app on the store. It was held twice a year, and each AppCademy hosted approximately 20 teams. Later, with the change of focus to more experienced teams, AppCademy also changed into a shorter two-week program with an emphasis on Go to Market training. It then took place more frequently, four times a year. Through AppCademy, the startups became acquainted with people from Nokia, Microsoft and other companies providing relevant services such as publishers and promoters. Every AppCademy ended with an Open House Day, where the startups could show their apps to the public. Apps graduating from AppCademy received an AppCademy watermark on their app logo on the AppCampus website.

AppCampus provided continuous monitoring of its published apps and offered support for them in several ways, such as introducing the app on the AppCampus website, Facebook page, Twitter, press releases, public events, and its contacts in advertising companies. It also supported its startups by running an alumni group on Facebook where they could contact their peers and AppCampus staff to discuss and share topics of interest. Moreover, it published videos of the relevant courses in the AppCademy Online space (in Eliademy) with open access for anyone interested.

By the end of the AppCampus operation period, 315 funded apps had been published by 305 startups. According to statistics released by AppCampus, its apps were generating nearly
seventy times more downloads and twice the revenue compared with the average rate on the WP platform.

2.6 Data Collection

I entered the field with a research concern in mind. I knew that the app markets were intensely competitive, and I was aware that young firms usually do not have many resources to invest in marketing. I was personally interested in how developers of new apps find their way into the competitive app market and create a business. Therefore, I decided to study early-stage app developers and performed semi-structured interviews with different app developers who were working on their first commercial app on the platform.

The semi-structured interview is neither a free nor a very structured discussion, and rather than asking fixed questions, focuses on certain themes (Järvinen, 2004, p. 141). In conducting semi-structured interviews, the researcher has prepared some questions, but allows for improvisation according to the responses of the interviewee (Myers and Newman, 2007). My interview themes were related to background information about the startups (their personnel, experiences, and roles), and business model elements about their current and, if any, previous applications (description, target audience, revenue model, success objective), and their interaction with AppCampus (how they had found it, experiences, learning).

My data included 47 interviews with game (28) and non-game (19) app developer startups. After each interview, I took field notes to reflect my understanding of the discussed topics. Besides conducting the interviews, I was present on the AppCampus premises and observed most group training sessions. I interacted with participants and talked with the staff as well to get a more in-depth insight into the activities during the AppCademy. I used this additional data for triangulation of my understanding about what was actually going on.

Gradually, by interviewing more startups, I realized that the non-game developers had a different nature and were following a different path than the game developers. Some had sponsors, for whom they were creating an additional service through the mobile app. Others were developing a mobile app for an already existing service on the Internet and wanted to port it to mobile phones. Most importantly, non-game app developers were trying to solve a specific problem for users, therefore had a specific target audience which would affect their marketing activities. In contrast, game developers were all independent, were creating a new concept from scratch, and finally were targeting mainly entertainment without any clear-cut problem to solve. Consequently, I decided to narrow down the scope of my study to game developers, especially when games are the largest category on all popular app stores to date. The rationale for this decision was further supported by the relevant literature that highlighted the differences of game and non-game apps in terms of market competition and user demands (Yin et al., 2014).

Later, after realizing the lack of existing theories and considering the richness of my data, I decided to use the classic GTM (Glaser, 1992). Deciding to use the GTM was doable, as I had the contact details of the interviewees in case of a need for follow-ups. I was aware that I should have followed the theoretical sampling principle to conduct a proper GTM. Consequently, after discussion with GTM experts, I decided to follow a ‘lighter’ theoretical sampling procedure as advised by Urquhart (2013, pp. 64-65). I reviewed the collected data, did a preliminary analysis of the emerging themes as I went along, and followed the next stages of analysis according to those themes. Going through the interviews one by one, and through the constant comparison of the data with previous emerged themes, I was becoming more
focused on the emerging concepts rather than a general-level analysis. I had conducted 28 semi-structured interviews with game developers, but through applying the theoretical sampling method, my findings became saturated after analyzing the data for 20 startups, referring to the point where I started to see ‘similar instances over and over again’ (Glaser and Strauss, 1967, p. 61).

The final 20 startups came from 13 different countries. The interviewees were mostly the founders of the startups, except for two firms where the project leader and designer and the public relation officer were interviewed. In these two cases, the interviewees had been with the firm since its early days and were informative about the firm’s operations. I considered the interviewees as the representatives of their startups (Churchill and Lewis, 1983), as in the early stages of a firm, the owner and the business are highly linked and the founder significantly invests themselves into the business (Cope, 2003). Details of the startups and their games are presented in Table 2.

The interviews lasted between 30 and 60 minutes, and I recorded them with the permission of the interviewees. Then, I transcribed the interviews partly by myself, and partly by outsourcing. Subsequently, I checked the transcriptions in comparison to the voice records for accurateness. Glaser (1998) discusses that recording of the interviews should be avoided to reduce unnecessary work, and to focus only on key concepts reflected in the researcher’s interview notes. However, despite my not knowing I was going to conduct a GTM at the time of the interviews, I would still opt for recording the interviews in any case, because it allowed for more in-depth interpretation of the data. Considering my lack of experience in conducting interviews as a novice researcher, as well as language-related issues related to English being a second language for myself and most of the interviewees, there was a high chance of missing some important points if I had relied solely on my field notes. Instead, I complemented the transcription of the interviews with the field notes to have a more in-depth perspective toward the study. As a matter of fact, technically, I revisited the data and re-coded it over many iterations, which would not have been possible without the transcriptions. In addition, upon deciding to use the GTM, the mass of data and its richness allowed me to apply the methodology and its principles of constant comparison and coding approaches without violating the theoretical sampling principle (Birks et al., 2013; Urquhart, 2013, pp. 64-65).
Research Methodology

Table 2: Details of the interviewed startups.

<table>
<thead>
<tr>
<th>Startup</th>
<th>Game</th>
<th>Country</th>
<th>Company and team details</th>
<th>Interviewee</th>
<th>Participation in AppCampus</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An adventure game for casual players. The game was going to be released shortly.</td>
<td>South Africa</td>
<td>Founded in 2002, but the focus had been on contracted client work, and not on developing their own apps. The company had two members involved in the game development. They met through a gathering for a technological project. They had wide experience of working for Nokia. Also, they had developed some apps for iOS and Android, but as client work or just for testing and learning. They had a previous game but they had not commercialized it.</td>
<td>Founder</td>
<td>Applied to AppCampus through the advice of a networking contact.</td>
</tr>
<tr>
<td>B</td>
<td>An adventure game for casual players. The game was launched 4 months previously.</td>
<td>Finland</td>
<td>Founded in 2013. The company had three founders working part-time due to studying at university. They met in their projects at university. The current game was their first commercial game. Previously, they had developed other apps, but only as hobby or school projects without commercialization.</td>
<td>Founder</td>
<td>Applied to AppCampus in open application.</td>
</tr>
<tr>
<td>C</td>
<td>An adventure game for casual players. The game was going to be released shortly.</td>
<td>Canada</td>
<td>Founded in 2013. The team had two members and were spending about 50 percent of their time on this game and the rest on client projects. They knew each other from art-related studies at university. They had developed another non-game app as a team, for the personal interest of a previous investor in the company.</td>
<td>Founder</td>
<td>Learned about AppCampus through their community of developers in their region and decided to develop a game for it.</td>
</tr>
<tr>
<td>D</td>
<td>A platformer game for casual players with the aim to also educate players about the environment through messages. The game was going to be launched shortly.</td>
<td>Belgium</td>
<td>Founded in 2008 as a consultancy company, but the actual work of app development started with the introduction of the Windows Phone in 2013. A one-man company with full-time dedication. The founder had developed several classic tile games on Windows Phone, mainly for learning and experimenting, but he had implemented in-app-advertisements in them. With the current app, he wanted to work on commercialization by trying the in-app-purchase model.</td>
<td>Founder</td>
<td>Learned about AppCampus on a blog and decided to apply for it.</td>
</tr>
<tr>
<td>E</td>
<td>A physics and platform game for casual players. The game had been BAFTA-nominated for a 'New Talent Award'.</td>
<td>UK</td>
<td>Founded in 2010. The company had seven members. Five were working full-time in the company and two were working part-time. In addition to developing the current game, they developed apps as did design work for clients. The current game was their first commercial game as a whole team.</td>
<td>Founder</td>
<td>Learned about AppCampus through their former business contacts at the platform owner. They applied directly to AppCampus through their website.</td>
</tr>
<tr>
<td>F</td>
<td>A racing game for casual players. The game was published about two months previously.</td>
<td>Canada</td>
<td>Founded for about five years. The company had three members taking care of business, art and development. Two of them were part-time workers and one was full-time. Their location was dispersed, and they worked virtually. They had five previous games on the BlackBerry, Android and iOS platforms, but had not worked on commercialization and only used in-app-advertisements. The current game was their first on the Windows Phone Platform. They wanted to offer a commercialized game by using a different revenue model and no advertisements.</td>
<td>Founder</td>
<td>Learned about AppCampus through their community in their region.</td>
</tr>
<tr>
<td>G</td>
<td>A racing game for mid-core players. The game was going to be released shortly.</td>
<td>Italy</td>
<td>Founded in 2011 and located in an incubator. It was founded by two members and had grown to nine. They all worked full-time and did all the work in-house. They had two other games in niche markets, one as a browser game for hard-core players and another on mobile platforms for mid-core players on Android, Amazon, and iOS. Their previous games were in a niche market, but they had created a community of dedicated users through them. The current game was their first game that they wanted to commercialize on the Windows Phone platform.</td>
<td>Founder</td>
<td>Learned about AppCampus through a neighboring company in their incubator and presented their idea at a MAAC event.</td>
</tr>
<tr>
<td>H</td>
<td>A shooter game for casual players. The game was launched three weeks previously.</td>
<td>Brazil</td>
<td>Founded in 2010. It had two founders who were both university students working part-time in the company. One founder was responsible for the game mechanics and the other for Facebook integration and the online experience of the game. They outsourced the art and music. They were game players and got to know each other through a game-playing website and met at university after some years of knowing each other. The current game was their first commercialized game.</td>
<td>Founder</td>
<td>They were invited to AppCampus by the platform owner because of their high downloads for their previous game on Windows Phone.</td>
</tr>
<tr>
<td>Startup</td>
<td>Game</td>
<td>Country</td>
<td>Company and team details</td>
<td>Interviewee</td>
<td>Participation in AppCampus</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>---------</td>
<td>--------------------------</td>
<td>-------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>I</td>
<td>A shooter game for casual players. The game was going to be released shortly.</td>
<td>Finland</td>
<td>Founded in 2012. The six-person team included three programmers, two artists, and one project lead and designer. They had another game on the market which started as a hobby project and was launched on iOS and Android. A simple version of the current game had been launched on Android, but they decided to develop and commercialize a Windows Phone version with additional features.</td>
<td>Project lead and designer</td>
<td>Learned about AppCampus through an event by the platform owner. The platform contacts saw the game, and advised the team to apply to AppCampus.</td>
</tr>
<tr>
<td>J</td>
<td>A puzzle game for casual players. The game was launched a month previously.</td>
<td>UK</td>
<td>Founded in 2013. It had two members, a game inventor and a programmer. They both worked part-time and had other jobs as well, but they hoped to be able to work full-time if they could make revenue. They had known each other for about 12 years. The current game was their first commercial game on a mobile platform. The game had a browser version as well.</td>
<td>Founder</td>
<td>Invited by the platform owner because of the interesting game idea as a browser game.</td>
</tr>
<tr>
<td>K</td>
<td>A role-playing game as an interactive fiction novel. The game was released three months previously.</td>
<td>US</td>
<td>Founded in 2013. It had one member. He took care of the creative and business side of his stories and outsourced the technical side. The founder was a former Microsoft employee as a leader of a writing team, and five months previously, he had decided to quit and work full-time on his own games. The game was a novel sequel to its previous iteration.</td>
<td>Founder</td>
<td>Advised to apply to AppCampus by a platform contact, because of the download levels of his previous games.</td>
</tr>
<tr>
<td>L</td>
<td>A platformer game for casual players. It was released four months previously.</td>
<td>Italy</td>
<td>Founded in 2012. There were eight team members in the company who were working full-time. There were three programmers, two designers, and a marketing person, and the rest were artists. It was the first of the company’s games to be published on the Windows Phone platform.</td>
<td>Founder</td>
<td>Learned about AppCampus funding opportunity by a platform contact and applied.</td>
</tr>
<tr>
<td>M</td>
<td>An adventure game for casual players. The game was going to be released shortly.</td>
<td>Chile</td>
<td>Founded in 2010. It had three members. The company was creating sequel games by keeping the main character, and they were published on iOS, but they wanted to start publishing on Android and Windows phones too. They did work for clients, too, but they wanted to focus on their own games. The current game was their first game on the Windows Phone. It was a sequel to a previous game using the main character.</td>
<td>Founder</td>
<td>Got invited to participate in MAAC because of being known in their country through the press.</td>
</tr>
<tr>
<td>N</td>
<td>An action and adventure game for casual players. The game was released four months previously.</td>
<td>Finland</td>
<td>In the process of registration. Located at a university premises. It had four members with two programmers, a graphic artist, and a designer. They had developed three small games prior to this one, and one of them had won a game competition, but they did not commercialize them. The current game was their first commercial game.</td>
<td>Founder</td>
<td>Learned about AppCampus through a local event when they were searching for funding. They showed their prototype to the AppCampus team at the event and were encouraged to apply.</td>
</tr>
<tr>
<td>O</td>
<td>A platformer game for casual players.</td>
<td>Finland</td>
<td>Founded in 2014. It had one founder and he used his life-partner for some user interface design. They outsourced art-related tasks when necessary. The company had another game launched on other platforms but it was not commercialized and did not succeed in receiving a high download volume. The current game was the first commercial game.</td>
<td>Founder</td>
<td>He did not remember how he got to know about AppCampus, but he applied to AppCampus directly through their website.</td>
</tr>
<tr>
<td>P</td>
<td>A racing game for casual players. It was going to be released shortly.</td>
<td>Finland</td>
<td>Founded in 2013. The team had five members: two programmers, one artist, one CEO and one person to take care of the server and back-end tasks. The current game was their first game.</td>
<td>Founder</td>
<td>Learned about AppCampus through information and news as they were in Finland.</td>
</tr>
<tr>
<td>Q</td>
<td>A racing game for casual players. It was going to be released in about four months.</td>
<td>Spain</td>
<td>Founded in 2010, with ten members: the CEO, art director and graphic designer, game designer, two artists, and five programmers. There were some family relations in the company. They were all working full-time. They were focusing on B2B business, but they wanted to start publishing their own games. They had three games on other platforms, but they had not managed to commercialize them. The current game was their first on the Windows Phone.</td>
<td>Founder</td>
<td>Learned about AppCampus through advice from a previous AppCampus alumnus whom they met at an event.</td>
</tr>
</tbody>
</table>
## Research Methodology

<table>
<thead>
<tr>
<th>Startup</th>
<th>Game</th>
<th>Country</th>
<th>Company and team details</th>
<th>Interviewee</th>
<th>Participation in AppCampus</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>An adventure game for casual players. The game was going to be released shortly.</td>
<td>Russia</td>
<td>In the process of registration. The company had five members who had been friends, among which four were the founders. The roles in the company were artists, programmers, and marketing staff. They were working part-time because of having other duties such as studying. They had a previous game but they had not managed to commercialize it. The current game prototype had won the Microsoft Imagine Cup 2014 competition.</td>
<td>Founder</td>
<td>Invited by the platform owner after winning the Imagine Cup.</td>
</tr>
<tr>
<td>S</td>
<td>A strategy game for casual players. The game was released three months previously.</td>
<td>Czech Republic</td>
<td>Founded in 2013. The company had eight members: two game designers, two graphic designers, two developers, one public relations officer, and one analyst. They had three previous games on iOS and Android. For them it was important to see that they could develop a game and publish it on the market. They wanted to develop the current game for the Windows Phone and commercialize it.</td>
<td>Public Relation officer</td>
<td>Learned about AppCampus through the website and participated in a MAAC event in their region.</td>
</tr>
<tr>
<td>T</td>
<td>A platformer game for casual players. The game was released about four months previously.</td>
<td>Czech Republic</td>
<td>Founded in 2012. The company had two members who used to be colleagues in a former company. They both worked full-time and used freelancers for art and music. They had two previous games, but they encountered problems with publishers and in the company and had not managed to commercialize them. Their focus was on their current game which they wanted to commercialize at full capacity.</td>
<td>Founder</td>
<td>Learned about AppCampus through a networking contact.</td>
</tr>
</tbody>
</table>
2.7 Literature Review Procedure

I followed the classic GTM principles in the process of reviewing the literature. Firstly, I conducted a ‘preliminary literature review’. The preliminary literature review helped to avoid being a ‘blank slate’ and have theoretical sensitivity around the existing theories and studies concerned with the research problem (Urquhart and Fernández, 2013, p. 1). In the preliminary phase, I considered the literature on mobile app development on platforms, software startups, and business accelerators, since I found them relevant to my context of study. However, I did not impose any literature on the analysis of my empirical data.

Subsequently, I conducted an ‘extended literature review’ to ground my concepts in the extant body of knowledge (e.g. Urquhart, Lehmann and Myers, 2010) when my emergent concepts and models had been developed through the empirical data. There were three phases in which I grounded my empirical findings: 1) The emergence of the first core category, 2) The emergence of the second core category, and 3) The emergence of the final core theme. To ground my first core category, I used the extended literature by digging into the literature on boundary resources on platforms. To ground the second core category, I went more in-depth into the literature on the iterative approach in software startups. Finally, to ground the final core theme as a result of relating the two core categories and scaling-up, I used the literature on the success of mobile games. Figure 1 illustrates the use of literature in theorizing the core categories and the core theme.

![Figure 1. Use of literature in theorizing the categories and the core theme.](image)

I will present the preliminary literature review in Section 3.1, and the extended literature review in Section 3.2.
2.8 Data Analysis Procedure

I loaded the interview transcriptions into the Atlas.ti software package before starting the process of coding. The reason for using the Atlas.ti was its facilitation for the data management and coding process, given that working with it was easy and did not take a considerable amount of cognitive load (Urquhart, 2013, pp. 101-102). However, to avoid getting trapped by the technology (Glaser, 1998, pp. 185-186), I used Atlas.ti only for open and selective coding, and to have a network view of the codes and their relationships. Alongside that, I used hand drawings and note-taking and writing to complement the software package. My main purpose of using a software package was to facilitate the data analysis process as long as it was not an obstacle in the way of my creative work. Then, I started the coding process through three phases of open coding, selective coding and theoretical coding.

2.8.1 Open Coding

Before starting the open coding of each interview, I listened to the interview recording once more to remember the story in more detail and to sense the atmosphere of the interview again. Subsequently, I went through the text carefully (Glaser, 1978) for sense-making of the interviewee’s world. To assign codes, I considered the guiding questions defined by Glaser (1978, p. 57) as: 1) What are these codes a study of? 2) What category does this incident indicate? 3) What is actually happening in the data? Although I went through the text line-by-line to get close to the data (Urquhart, 2013, p. 24), the answers to these questions were not necessarily met through the line-by-line coding. Therefore, in assigning the codes, I considered the adequate length necessary for sense-making without limiting myself to the line-by-line procedure (Glaser, 1992). Table 3 includes some examples of my open codes.

Table 3. Examples of open coding.

<table>
<thead>
<tr>
<th>Piece of data</th>
<th>Open code</th>
</tr>
</thead>
<tbody>
<tr>
<td>“It just happened that we found some amazingly talented people just on Craig's list or up and coming people coming out of university who are just happy, they just desperately need experience because with that experience they'll eventually get into the union.” (Startup C)</td>
<td>Accessing low-cost resources</td>
</tr>
<tr>
<td>“So, our community is very involved with our games and of course we are very proud of that. We're happy about that. But we will probably also leverage on that for our beta-testing and fine-tune.” (Startup G)</td>
<td>Monitoring user behavior in the game</td>
</tr>
<tr>
<td>“Most important thing is their movement in-game, where they go, etc. And if they return, what part in the game, that's kind of very important to know that they get annoyed, etc.” (Startup T)</td>
<td></td>
</tr>
<tr>
<td>“We're going to start analyzing some basic stuff like how much they play? Are they still playing, see like how much time when the people install the game maybe he did that the third day maybe he doesn't delete it. So, we're going to see like the live value of the user, like how much time he plays, how much does he spend in terms of monetization.” (Startup M)</td>
<td>Focusing on downloads and building a reputation</td>
</tr>
<tr>
<td>“I like the downloads, largely in part because it builds my brand. And so, it's kind of like, if you can get the numbers, you can make the money later. Or at least you have the potential to make the money later.” (Startup K)</td>
<td></td>
</tr>
</tbody>
</table>
From the start of the coding process, I was careful to follow the principle of theoretical sampling, as explained in Section 2.6. In doing so, my initial codes directed me toward future coding (Glaser, 1978, p. 56). Consequently, going through the next interviews, I was considering the instances of the earlier codes, as well as keeping an open mind about the emergence of new concepts. Following this procedure of coding, the number of open codes reduced comparatively in the later interviews. The number of open codes in the first interview was over 190, while the 20th interview contained only 36 open codes.

Initially, a sum of more than 500 open codes was identified from all the interviews, but subsequently – through many iterations and merging and refining, the number of open codes was reduced to 245. Table 4 illustrates a few examples of the merging of codes along with memos to explain the logic.

Table 4. Examples of open code merging and renaming.

<table>
<thead>
<tr>
<th>Initial codes</th>
<th>New larger codes</th>
<th>Memo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outsourcing localization; benefiting from local resources; outsourcing; exploiting community; using family and friends; using professional network; using interns</td>
<td>Accessing low-cost resources</td>
<td>The startups use all these ways to get the cheapest resource available to them</td>
</tr>
<tr>
<td>Finding challenge level different after launch; receiving unfair reviews</td>
<td>Getting insight into user behavior in the game</td>
<td>I merged the codes into a higher-level code because I understood that the startups are learning the users’ perspective differently than their assumptions prior to launch of game</td>
</tr>
<tr>
<td>AppCampus as facilitator with external network</td>
<td>Benefiting from external network in AppCademy</td>
<td>I renamed the code to make it more compatible with other types of codes relating to benefiting from AppCampus</td>
</tr>
<tr>
<td>Benefiting from peer network in AppCademy; Benefiting from external network in AppCademy</td>
<td>Benefiting from networking in AppCademy</td>
<td>I merged the codes in order to make it to a higher-level code, including all aspects of networking in AppCademy</td>
</tr>
</tbody>
</table>

2.8.2 Selective Coding

In the next phase, I did the selective coding. I organized the open codes into larger selective codes which contributed to the creation of the core categories (Urquhart, 2013, p. 49). This was facilitated by many iterations of open coding and constant comparison, through which I was realizing the emergence of patterns and commonalities that helped me to group them under umbrella terms, i.e. selective codes. Table 5 shows some examples of the selective coding by presenting the open codes that comprise them, and my memos to rationalize their organization. Altogether, I had 11 selective codes.
Subsequently, I started the iterative process of interpreting how the selective codes could contribute to my understanding of the research concern (i.e. how these young firms operate in the intensely competitive market).

The iterative process was done by constant comparison and memo-writing to help in understanding the emerging concepts, relationships and stories. I did many iterations of coding, re-coding, merging, moving open codes from one selective code to another, and moving the selective codes from one core category to another. My memos included both process memos indicating how I was making the change, as well as analytical memos indicating my thought process and conceptualizations (Gasson and Waters, 2013). Some examples are presented in Tables 4 and 5. Figure 2 is also an example of an analytical memo where I was trying to understand the steps that the startups were taking, their concerns and the facilitations received by the accelerator.

Gradually, my two core categories were emerging and being refined. They were: 1) Utilizing the accelerator’s facilities (comprised of six selective codes), and 2) Excelling in the market (comprised of five selective codes). In each of these two core categories, I drew the relationships between the selective codes to conceptualize what was happening in my data. These two categories helped me to construct my first and second research questions as follows.

<table>
<thead>
<tr>
<th>Open codes</th>
<th>Selective code</th>
<th>Memo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Using low-cost marketing; 2) focusing on downloads and building a reputation; 3) getting featured to get downloads; 4) expanding the market step-by-step; 5) contacting users; 6) Using a publisher to launch games; 7) making a portfolio</td>
<td>Market-related activities</td>
<td>All these open codes relate to what the startups do when they are dealing with the marketing aspect of their games.</td>
</tr>
<tr>
<td>1) Exploiting team members’ capabilities; 2) accessing low-cost resources</td>
<td>Resources</td>
<td>These open codes relate to how the startups acquire their needed resources.</td>
</tr>
<tr>
<td>1) Differentiating the game, 2) integrating social aspect in the game, 3) Developing the game, avoiding complex game design, 4) designing for monetization, 5) caring about getting user star ratings and reviews, 6) paying attention to game quality</td>
<td>Game design activities</td>
<td>All these open codes relate to the various aspects that developers consider when they design their games.</td>
</tr>
</tbody>
</table>

![Figure 2. Example of analytical memos.](image-url)
RQ 1: How do early-stage mobile game startups utilize the accelerator’s facilities in developing and publishing their games?

RQ 2: How do early-stage mobile game startups excel in the market?

2.8.3 Theoretical Coding

In each of the two core categories, I conducted the theoretical coding by conceptualizing how the selective codes related to each other to be able to integrate it with the literature (Glaser 1978, p. 72). In both cases, I followed the coding family of the ‘model’ type (Glaser, 1978, p. 81). Then I conducted the extended literature review to go more in-depth into some aspects that my models were implying, such as research on boundary resources and on the iterative approach in the startups, to ground my findings.

Subsequently, I moved toward the big picture – the core theme. I drew the relationships between the two core categories and theorized at a higher level through the help of the memos and a creative process of conceptualizing. Glaser allows for flexibility in relating the categories for data integration and theorizing (Glaser 1998). In a similar manner to theorizing the core categories, I followed the ‘model’ type of coding family for the core theme as well (Glaser 1978, p. 81). Consequently, I developed a theoretical model called ‘Dynamics of Success for Early-Stage Mobile Game Startups’ that I will present in Figure 9 (in Section 6.2). In the end, I chose the extended literature relevant to the success of mobile apps and mobile games according to the emergent concepts (e.g. Glaser 1992; Urquhart et al. 2010). With the help of the extended literature, I grounded my findings.

2.9 Quality of the GTM Research

To evaluate the quality of a GTM research, Glaser and Strauss (1967, pp. 237-250) introduced four criteria for the developed theory: fit, understandable, general, control. These criteria were later reaffirmed by Glaser (1978, pp. 4-6) and renamed fit, work, relevance, and modifiability of the theory. In the following, each criterion is explained and I discuss how they have been met in this study.

Fit refers to the emergence of the categories and concepts from the data, rather than using any preconceived codes or concepts. I met this criterion by using the classic GTM and being fully dedicated to allowing the emergence of concepts from the data. As already mentioned, I did not impose any idea, codes or concepts from the literature and/or other sources. Instead, I looked closely at the data by asking the guiding questions defined by Glaser (1978, p. 57), and was fully reliant on what the data was showing by a constant comparing (Urquhart and Fernández, 2013). Although the verification of coding is not required in an interpretivist GTM (Urquhart, 2013, p. 61), I have tried to clearly illustrate how I conducted the coding and the steps I took throughout the study. The purpose was to show the ‘chain of evidence’ to indicate the quality
of my research. I performed a rigorous iterative process of creating the substantive theory (Urquhart and Fernández, 2013) by coding and re-coding to refine, conceptualize, and to reach saturation in the emergence of the substantive theory and scaling up. I used the existing literature only in the phase of grounding the concepts in my theoretical models.

Work refers to the ability of the grounded theory to explain and interpret what has happened and predict what will happen. The criterion of work is met through getting the ‘facts’ through systematic research (Glaser, 1978, p. 4). To meet this criterion, throughout this study, I focused on understanding ‘What is going on?’ in the data, enabled by the rigorous analysis of the data led by theoretical sampling and constant comparison. The emergent concepts come from the data, possessing the nature of being both analytical and sensitizing, following Glaser and Strauss (1967, p. 240). The audience can ‘understand’ the substantive theoretical models in each of the two core categories, and eventually the scaled-up theoretical model of ‘dynamics of success for early-stage mobile game startups’. Accordingly, they can use the model to either explain a current problem or use it for potential problems and situations.

Relevance refers to the focus of the theory on the core processes and concerns referred to in the data. Glaser (1978, p. 5) explains that relevance is reached by allowing the core problems and processes to emerge from the data and not imposing any preconceived concepts on the data. Thus, the researcher does not need to convince why the particular focus and core concepts were chosen, and instead should show how the core concepts are relevant to the data. I followed the relevance principle by taking a classic GTM in this study, and using theoretical sampling and focusing on emerging concepts to reach saturation. Consequently, all the emerged concepts are well-grounded in the data, and there exists much empirical evidence to support each emerged concept.

Modifiability refers to the ability of the theory to be modified for preservation as new data emerge over time. As long as reliance on data is the essence of the grounded theory, naturally there is room for modification of the theory when new data appears (Glaser, 1978, p. 5). Therefore, the theory preserves its continuous relevance by predicting and allowing for modification rather than being outdated (Niekerk and Campus, 2009). The obvious deduction is that my emerged theory in this study is built on the basis of the substantive data, and in the case of applying it to a new context, I should have an open mind to seek any possible modifications.
3. Literature Review

In this chapter, I firstly present the ‘preliminary literature review’ in Section 3.1 and its subsections (3.1.1–3.1.3). Then, I will present the extended literature in Section 3.2 and its relevant subsections (3.2.1-3.2.3).

3.1 Preliminary Literature Review

This study is located at the intersection of three research streams: 1) mobile app development on digital platforms, 2) software startups, and 3) business accelerators. The review of the first stream contains a general overview of digital platforms, mobile application platforms, and mobile game applications (Section 3.1.1). The review of the second stream addresses the definition of software startups and the concept of lean startups (Section 3.1.2). Finally, the review of the third stream introduces business accelerators and a form of them, corporate accelerators (Section 3.1.3).

3.1.1 Mobile Application Development on Digital Platforms

This section is comprised of three subsections. It starts by introducing digital platforms, narrows down to mobile application platforms and the development of apps on them, and finally offers a review of mobile game applications and the relevant issues in their development.

Overview of Digital Platforms

The concept of platform is defined based on the idea of having a common core that is used in the creation of multiple derivatives at a lower cost than creating separate services and/or products from scratch (Baldwin and Woodard, 2009). Gawer (2009, pp. 46-58) classifies platforms into four types: 1) An internal platform, where a platform is within a firm and is used to increase efficiency. An example is product development platforms in car manufacturing, where various brands are produced by modifications made to a core design. 2) A supply chain platform, which is an extension of the internal platform across the supply chain; for example, a shared platform between members of an alliance (e.g. Nissan-Renault alliance). 3) An industry platform, where the owner of the platform allows members of the industry to make complementary services or products on the platform. Consequently, the platform becomes a precious point of control for the platform owner, such as the Windows platform by Mi-
4) A two (or multi)-sided platform, where the platform facilitates interaction between two (or multiple) distinct groups of users, and works as a point of transaction, for example eBay. Unlike industry platforms, multi-sided platforms do not necessarily allow for the creation of complements.

A popular form of industry platform that has gained much attention is the digital platform. These platforms have created new grounds of doing business and connecting people within and across industries. Digital platforms are defined as ‘the extensible codebase of a software-based system that provides core functionality shared by the modules that interoperate with it, and the interfaces through which they interoperate’ (Tiwana, Konsynski and Bush, 2010, p. 676). They have been designed to serve various purposes which are not mutually exclusive, such as social media to connect people (e.g. Facebook, Instagram), sharing spaces for content (e.g. YouTube, Pinterest), and a space for peer-to-peer connection (e.g. Uber, Lyft, Airbnb, Bitcoin). Joining these platforms and using their services has turned into a lifestyle element for people, and consequently competition among businesses has turned into platform-centric competition with the aim of bringing a large number of platform users on the supply and demand sides together. An example is the transformation of the mobile industry from device-based competition to platform-based competition, where users not only use the devices, but also use the accompanied apps on the platforms.

The emergence and rapid growth of digital platforms is enabled by their modular layered architectures. Their modular architectures are designed to have two sets of components: 1) a component with low variety and high reusability (i.e. platform infrastructure), and 2) a set of components with high variety and low reusability (i.e. complements) (Baldwin and Woodard, 2009). For example, mobile platforms are comprised of an infrastructure in the form of a mobile operating system (e.g. iOS) and complements as the apps that users can install. The modularity means that the digital components are embedded into physical products. For example, mobile devices are physical products that allow the digital platforms to be embedded into them. This layered modularity mandates the complements to be compatible with the specification of the physical devices where the platform is running (Yoo, Henfridsson and Lyttinen, 2010; Yoo, Boland, Lyttinen and Majchrzak, 2012). For instance, a software application can be developed and run on a platform only if it matches the physical attributes of the computer or mobile device on which it is going to be executed. Software applications are ‘executable pieces of software that are offered as applications, services or systems to end-users of the platform’ (Ghazawneh and Henfridsson, 2013, p. 175), such as mobile apps.

Benefiting from the layered modular architecture, these platforms are dynamic (Yoo et al., 2010; 2012). They allow heterogeneous actors to add capabilities and create contents, services and applications (Tilson, Lyttinen and Sorensen, 2010); a characteristic known as generativity (Yoo et al., 2012). Generativity is defined as the ‘overall capacity to produce unprompted change driven by large, varied and uncoordinated audiences’ (Zittrain, 2006, p. 1980). Accordingly, a generative digital platform is built on the notion of incomplete-
ness and evolves by the creation of new uses and contents by heterogeneous actors (e.g. Hanseth and Lytyinen, 2010; Yoo, 2013; Zhu and Iansiti, 2012). So, the platform owner takes advantage of the high innovativeness by using global experts without paying the direct costs of employing them (Bergvall-Kåreborn and Howcroft, 2011). Meanwhile, the platform owner finds a balance between letting the actors fully utilize the platform generativity while controlling the actors to stay in line with its strategies (e.g. Eaton, Elaluf-Calderwood, Sørensen and Yoo, 2015; Elaluf-Calderwood, Eaton, Sørensen and Yoo, 2011; Ghazawneh and Henfridsson, 2013; Lytyinen, Yoo and Boland, 2016). The paradox of generativity and control and its resolution is done through both technical (e.g. API, SDK) and social (e.g. rules and regulations) controls. On the technical side, it provides application programming interfaces (API), software development kits (SDK) and other tools that allow the development of the complements (e.g. apps), and on the social side it provides incentives and rules (Ghazawneh and Henfridsson, 2010).

Ghazawneh and Henfridsson (2013) refer to the notion of controls as resourcing and securing through the use of boundary resources (more details in Section 3.2.1). In resourcing, the platform owner extends the scope and diversity of the platform, easing the development restrictions for the developers. In securing, the platform owner controls the platform against the applications and the activities that may infringe the platform by applying new rules and regulations or by tightening existing ones. Resourcing can be done by the developers as well (referred to as self-resourcing), in their response to a heavy securing by the platform owner. The case of iPhone jail-breaking is an example of self-resourcing through which developers have enabled themselves to run unauthorized third-party applications on iPhone devices against the platform owner’s wish.

However, it should be noted that the relationship between generativity and control does not always pull in opposite directions, and the increase or decrease of one does not necessarily lead to the opposite change in the other. In a study on the iOS and Android platforms, it is outlined that control can both increase and decrease generativity (Remneland Wikhamn, Ljungberg, Bergquist and Kuschel, 2011). When this is through providing standards and guidelines by the platform owner, it can actually ease the creation of applications for developers, hence increasing the generativity of the platform. On the other hand, when the control considers issues such as permission requirements, it can hinder generativity.

**Mobile Application Platforms**

In the past ten years, the mobile industry has gone through a fundamental transformation where the locus of control and power has changed from the mobile network operators to the platform owners. Formerly, mobile services were controlled mostly by the network operators, mobile device manufacturers, and a few mobile content providers and application developers (Holzer and Ondrus, 2011), each responsible for only part(s) of the process. Mobile operators hold the gatekeeping roles by having application providers on the one hand and the mass of consumers on the other. Hence, the mobile platform...
sources were accessible by only a limited number of application developers (Bergvall-Kåreborn et al., 2010; Le Bodic and Lannartz, 2009). The structure started to change with the launch of open mobile platforms where third-party developers could access the source codes of the device and the platform with less restriction (Le Bodic and Lannartz, 2009).

In July 2008, Apple opened up its iOS platform to third-party developers and offered a digital marketplace for the distribution of apps to Apple devices. Third-party developers are those developers who create and distribute their apps on the platform under an arm’s length revenue-sharing contract with the platform owner (Boudreau and Lakhani, 2009; Ghazawneh and Henfridsson, 2013). The digital marketplaces are generally referred to as app stores, and hold bridging roles between the app developers and the app users with the seamless delivery of mobile apps (Basole, 2009; Basole and Karla, 2011).

Apple’s mobile application platform, and the concept of the app store became so popular that soon after, other large software and mobile manufacturer companies started to imitate Apple’s initiative. Examples include but are not limited to Google Play (formerly Android Market) by Google in August 2008, BlackBerry App World by Research in Motion (RIM) in April 2009, Ova Store by Nokia for Symbian-based phones in May 2009, and later the Windows Phone store by former Nokia-Microsoft. However, over the years, some of these platforms ceased operation or stayed at a small scale, and Google and Apple have become the major platforms.

Mobile application platforms connect multiple distinct groups of app users to service providers (Bergvall-Kåreborn and Howcroft, 2014; Rochet and Tirole, 2003) such as app developers, device manufacturers, operators, payment handlers, and advertisers. On the supply side, app developers use development tools (e.g. SDK, API) to create and publish their apps to the platform, and on the demand side, mobile users download the apps to their devices. The increase in the number and diversity of apps attracts more users to the platform, and the increase in the number of users attracts more developers to the platform. Transactions made on the platform generate revenue for the platform owner, developers, and other service providers (Bresnahan et al., 2014; Garg and Telang, 2013).

The mobile application platforms have brought various consequences for mobile app developers, which have become topics of interest for IS researchers. A stream of research focuses on the changes in the market, at both the macro and micro level. The macro level aims to understand how the platform owners manage the platforms. The micro level focuses on the app developers and how they cope with the new structure. A pioneer study on this domain is that of Holzer and Ondrus (2011) where they discuss the market trends in mobile application development. They anticipate the centralization of application portals, the openness of technology, and diversity of devices and platform integration, and hypothesize that these changes will have several effects, such as increasing job opportunities for developers, lowering development and distribution costs, easing the development process and access to consumers. In the same vein, Bergvall-Kåreborn and Howcroft (2011) introduce the MADD (mo-
bile application and distribution model) to discuss the activities of the developers on the platforms.

The change in the market structure has been received by developers with various motivations of economic or intrinsic nature, varying from the technological improvement of their skills to making profit (Bergvall-Kåreborn and Howcroft, 2011; Ghazawneh and Mansour, 2015). For example, the availability of a large-scale distribution channel, development tools, and the review process by the platform owner provides intrinsic value for developers in terms of ease of development. Further, the developers can make economic value in various ways, such as the ease of payment handling on the platform, users’ positive reviews leading to more downloads, and the reputation gained by distributing their apps on authorized platforms.

Despite the existing opportunities, the developers also face challenges. They must cope with diversity (dealing with multiple platforms, markets, customers, etc.), lack of knowledge (business knowledge, platform strategies, etc.), and market structure (high control by platform owners, high competition in the market, unreliable income, etc.) (Bergvall-Kåreborn and Howcroft, 2014). The intense competition makes revenue creation very difficult, and requires the developers to find a balance between technical orientation and market orientation in order to succeed (Qiu et al., 2017). Developers go through excessive work hours and deal with inconsistency in their revenue-making (Bergvall-Kåreborn and Howcroft, 2013a). In another study on the Apple and Google platforms, Bresnahan et al. (2014) show that despite the low barrier to entry in the app market, making the apps visible is a serious challenge for developers. They discuss that the top-ranking lists in the app stores allow for the rapid adoption of the apps, but fail to capture the right customers. Consequently, the ranking lists cause more advantage for established firms with larger marketing budgets, rather than entrepreneurial firms with insufficient budgets. Therefore, due to the limitation of the ranking list structures, the crucial task of gaining visibility for the app through marketing effort is borne by the developers (Bergvall-Kåreborn and Howcroft, 2013a).

The more competitive categories of apps, such as games, creates more challenges for developers. Yin et al. (2014) compares games with non-game apps on the Apple App Store and finds that games have a lower barrier to entry because of the vast number of existing development tools and libraries, which leads to more intense competition. Meanwhile, game users have more heterogeneous demands and expectations, and they are not patient enough to wait for a game to be fixed. Subsequently, the study advises game developers to start making new games rather than fixing old ones if they want to be in a top position on revenue-generating ranking lists. In another study on games in the App Store, it is found that game developers can have high revenue generation if they release a portfolio of games simultaneously to the market as opposed to non-game developers that can work on multiple version of the same product over substantial time intervals (Davis et al., 2014).
Mobile Game Applications

A mobile game application (hereinafter mobile game) is a video game that is installed and played on a mobile device, such as a smartphone (Liang and Yeh, 2011). Mobile games can already be embedded in the mobile operating system by the device manufacturer or be installed on the device by the user. The first mobile game embedded in the device was the famous game of Snake in the late 90s by Nokia (Rajala, Rossi, Tuunainen and Vihinen, 2007). Similar to video games, mobile games can be designed as a single player or a multi-player game and can have either original ideas (e.g. Angry Birds) or be designed as a digital version of a physically available game (e.g. backgammon). The games can be played either as native games on the device where a connection to the Internet is not required after installation, or as online games (Liang and Yeh, 2011).

The introduction of mobile application platforms for the creation and distribution of games has made it easier for third-party developers to enter the game development domain independently from large game development corporations. Rather than being professional developers, the third-party developers might include hobbyists, fans, artists and amateur developers (Bowen and Deuze, 2009). The ease of entry into the market has made games the largest category on all platforms (Merikivi et al., 2016; Yin et al., 2014). In 2017, 80 percent of the revenue from mobile apps was made by games, equaling 48 billion dollars (Chan, 2018). Despite the large number of games, many can fail (Merikivi et al., 2016) due to the high competition. The intense competition has resulted in the domination of the game market by freemium games (Chan, 2018).

Many scholars have considered the factors that may lead to the adoption of games by users. Some factors relate to the game quality in terms of content, design aesthetic, navigation and mechanism and game size for download (e.g. Bowen and Deuze, 2009; Kleijnen, De Ruyter and Wetzels, 2004; Okazaki, Skapa and Grande, 2008; Penttinen, Rossi and Tuunainen, 2010). Users also care about aspects related to the game purchase, such as payment, installation, support (Kleijnen et al. 2004; Penttinen et al., 2010), and having no security risk (Okazaki, 2008). In addition, the social aspect of communicating with other players (Kleijnen et al. 2004; Penttinen et al., 2010), gaming experience, enjoyment, and novelty, as well as perceived ease of use are important in game adoption (Lee and Quan, 2013; Merikivi et al., 2016; Zhou, 2013). It has been found that the complexity of the game (i.e. ‘intellectual calculation and motor activity consumption during game play’) and the familiarity of the users with the game (i.e. ‘the extent to which the game demands human cognitive capacity’) can increase the level of user engagement with the game (Li, Jiang, Tan and Wei, 2014). Finally, contextual factors such as the possibility to try the game before purchasing it (Penttinen et al., 2010) or the lifestyle of the users (Liu and Li, 2011) have been noticed as factors affecting the adoption of games. Although different researchers have examined various influential factors on game adoption, these factors are not mutually exclusive. For example, Merikivi et al. (2016) show that the design aesthetic, perceived ease of use and novelty of a game can encourage the perceived enjoyment, thus resulting in the users’ continuous playing of the game.
In general, mobile game developers need to address three aspects in their games: humanistic (i.e. the effect of the game on users, society and the gaming community), construction (i.e. the actual making of the game from a technical perspective), and business (i.e. the economics around computer games in terms of markets and competition) (Hakonen, Mäkilä, Smed and Best, 2008). Therefore, game development requires collaboration among various areas of expertise such as artists, animators, game designers, level designers, musicians and programmers, while traditional software development only includes programming and testing. In addition, factors related to finding the market, revenue-making and scaling-up of the game users must be considered by developers if they want to create successful games (Järvi, Mäkilä and Hyrynsalmi, 2013)

3.1.2 Software Startups

Startups have been vaguely defined through considering various factors in the extant literature (Paternoster et al., 2014). What defines a startup is the status of flux the firm is in due to the temporality of the business model and its organizational process, and the rapid evolution that it expects to happen (Giardino et al., 2014; Standing and Mattsson, 2016). In the same vein, a software startup is a startup that is working in the digital high-tech context by offering software products and/or services to the market. This study borrows the definition of software startups by Giardino et al. (2016, p. 585) as:

Those organizations focused on the creation of high-tech and innovative products, with little or no operating history, aiming to aggressively grow their business in highly scalable markets.

Startups have scarce resources in terms of knowledge, finance and networks, while working in uncertain markets and with new technologies (Abrahamsson et al., 2016). They struggle with resource limitations when aiming to promptly react to uncertain markets, adapt to market changes, and finally to scale up (Giardino et al., 2016; Giardino et al., 2014; Paternoster et al., 2014). Startups are at danger of facing a long lead-time for developing a suitable product, which may consume their financial resources and lead to them failing before they can create a cash flow (Hilmola, Helo and Ojala, 2003). Consequently, they have to scale-up before running out of money.

For scaling-up, startups offer market-driven products that are designed for high-potential markets and are appealing to the mass of customers (Giardino et al., 2016; Paternoster et al., 2014). Creating a product and/or service that meets customer needs (Paternoster et al., 2014) and the requirement to adapt quickly to the market demands (Giardino et al., 2016) can be a challenge for startups due to their limited resources. In fact, a major reason for failure in software startups is failure in finding the product-market fit (Giardino, Wang, et al., 2014). In addition, they have little operating history (Sutton, 2000), a lack of structure, and a lack of validated learning when it comes to understanding customers’ real problems (Giardino et al., 2015). Consequently they face
problems in defining the product, finding the fit between the product and market, defining their business strategy for profit-making, and technological uncertainties (e.g. Giardino et al., 2014; Crowne, 2002; Xiaofeng et al., 2016). Startups may also suffer from isolation from business networks (Raz and Gloor, 2007). As startups grow and learn, the focus moves from product-related challenges (e.g. building the product and finding funding) to market-related challenges (e.g. customer acquisition and scaling-up) (Xiaofeng et al., 2016).

The flux nature of startups requires them to be in continuous learning process with regard to their product-market fit and the evolution of their business model. Getting meaningful customer feedback from the right customers and through the right method of data collection are important factors in the early stages of startups (Hokkanen and Leppänen, 2015). They need to be both efficient (i.e. minimize development costs and maximize gained value), and systematic (i.e. validate whether customer value is generated) (Bosch, Olsson, Björk and Ljungblad, 2013). They are encouraged to avoid over-investment in technical solutions, and instead build a workable product and focus on getting user groups for it (Giardino et al., 2016; Paternoster et al., 2014; Standing and Mattsson, 2016).

From the technical point of view, software development methods of an agile nature are common in software startups and aim for the better organization of the software development to meet the dynamic and unpredictable conditions of the business environment (Fagerholm, Guinea, Mäenpää and Münch, 2014). These methods focus on optimizing the software development process; therefore, they may miss focusing on the optimization of the whole product or service in terms of its market success (Poppendieck and Cusumano, 2012). However, two approaches exist that take a broader perspective than agile methods, taking into account the market success of the product and creation of a business: 1) the customer development model (Blank, 2006), and 2) the lean startup (Ries 2011). In the existing literature, agile development methods have been addressed as part of these more comprehensive approaches (e.g. Fagerholm et al., 2017; Bosch et al., 2013).

The Customer Development Model (Blank, 2006) involves the customers from the very beginning of the process of conceptualizing and idea generation to the final stages of creating a business. The model is comprised of four stages: customer discovery, customer validation, customer creation, and company building. The first two stages are about searching to understand who the customers are, what they need, and what they are going to pay for. The next two stages cover execution to scale the business and turn it into an efficient and effective one. In other words, the search aims to find the product-market fit and the sales model, while the execution aims at creating demand and building a business (Fagerholm et al., 2017). A factor worth highlighting in the model is its nonlinear nature, and that iteration is allowed for refinements between the stages. Based on this model, another model of iterative nature, named lean startup has been developed and is widely recognized.
Inspired by Toyota lean production system, Ries (2011, p. 20) introduced the concept of the 'lean startup' and explains:

The Lean Startup is a new way of looking at the development of innovative new products that emphasizes fast iteration and customer insight, a huge vision, and great ambition, all at the same time.

The lean philosophy is to create the right thing for the customers (Ries, 2011, p. 20) while eliminating waste (i.e. anything that does not create value for the customer) (Ibid, p. 48). Value is what the customer is willing to pay for (Chorev and Anderson, 2006). It follows a 'hypothesis-driven' approach to pursuing entrepreneurial opportunities (Eisenmann, Ries and Dillard, 2012) by creating a product and/or business hypothesis and testing it with the customers. Lean startups work according to the cycle of build-measure-learn (BML) to have validated learning. In BML, the product/service is built, delivered to customers, and according to early feedback, learning is gained which is translated into adjustments to the product/service.

To fulfill the BML cycle, the lean approach advises startups to offer a Minimum Viable Product (MVP) to the market to receive feedback. An MVP is a version of the product with minimum features and minimum development time, allowing a complete round in the BML cycle (Ries 2011, p. 77). Startups have an initial hypothesis when they offer the MVP to the market, and are prepared to pivot if it does not fit the market (Giardino, Wang, et al., 2014). The MVP is designed using characteristics that would encourage users to use it and to provide feedback (Hokkanen and Leppänen, 2015). An important factor in the scope of an MVP is that it must be complete enough to demonstrate the value that it is supposed to offer (Moogk, 2012; Münch et al., 2013) and not disappoint the users (Hokkanen and Leppänen, 2015). According to the results of the BML, the startup may preserve the current direction, or pivot to a new one.

Pivoting is triggered by internal or external factors which may (Ries, 2011) or may not be in the control of the firm (Bajwa, Wang, Nguyen-Duc and Abrahamsson, 2016). In their study of 49 software firms, Bajwa et al. (2016) found that the majority of pivots happen due to unexpected factors of negative customer reaction (e.g. slow user acquisition, low user retention rate, no growth) and/or a flawed business model. However, they highlight that by considering the possibility of learning from the market and pivoting, startups can feel happy about early failure and aim for success. An example is the Yelp website, that was first designed to be an automated system for email recommendations to friends, but pivoted into a successful one-stop shop for local business reviews. Consequently, openness to pivoting is seen as an essential element for startups success.

3.1.3 Business Accelerators

Business incubators have existed since the late 1950s to assist new ventures in their creation and growth (Mian et al., 2016). They started in their simplest
form offering cheap office services and tangible assets of financial resources (Barrow, 2001; Hackett and Dilts, 2004) to their current complex form offering intangible and knowledge-based services (Smilor, 1987; Allen and McCluskey, 1990). Generally, they offer seed capital, mentoring, and networking opportunities (Bergek and Norrman, 2008). Incubators aim to support the creation and growth of small entrepreneurial ventures (Pauwels et al., 2016) by sheltering them from market forces and helping with their sustainability (Cohen, 2013).

Accelerators are a new generation of business incubators that were introduced in the mid-2000s following the rise of low-investment technological firms after the dot-com bubble (Pauwels et al., 2016). A widely accepted definition of an accelerator is presented by (Cohen and Hochberg, 2014, p. 4) as:

> A fixed-term, cohort-based program, including mentorship and educational components, that culminates in a public pitch event or demo-day.

The first accelerator, *Y Combinator*, was established in 2005 in Silicon Valley to offer seed capital and three months of intensive training to selected startups in return for a minor equity stake in them (Fehder and Hochberg, 2015). The next accelerator, *Tech Star*, was established in 2007, and soon after, accelerators became widespread all over the world. They mainly focused on supporting new Internet and mobile startups to accelerate their growth (Miller and Bound, 2011). Their main characteristic was to have an open application process and a seed-investment offering in exchange for a minority stake in the startup. In return, they offered intensive mentoring and education over a limited period of time (Cohen and Hochberg, 2014; Pauwels et al., 2016; Salido, Sabás and Freixas, 2013).

Despite the fact that accelerators are rooted in incubators, there is a fundamental difference between the two in terms of the nature of their assistance and objective. Incubators help new ventures by sheltering them from market forces and providing them with temporary care (Aernoudt, 2004). In contrast, accelerators expedite the adaptation of ventures to the market through a short-term intensive program (Cohen, 2013). Their aim is to help new ventures to build their initial products or services by providing them with concentrated mentoring and business training, and helping them to secure their financial and human resources (Cohen and Hochberg, 2014). Consequently, accelerators offer a fixed-term period of assistance to startups (usually three months), while incubators provide much longer support (one to five years) (Cohen, 2013).

The components of an accelerator are defined in five building blocks by Pauwels et al. (2016) as: 1) the program package, 2) the strategic focus, 3) the selection process, 4) the funding structure, and 5) the alumni relationships. They are shown in Figure 3. The program package consists of the services that are offered to the ventures, in the form of the designed mentoring and educational packages in various business fields such as finance, marketing and management. These services are offered by experienced mentors alongside consultancy services from the accelerator team. In addition, through the demo-day the
ventures can contact investors and customers and present their product and/or service. Besides these services, through the co-location structure, startups can network with their peers. The strategic focus of the accelerator can be either on a specific industry or sector or on a geographical area, and their decision to either serve locally or internationally. The selection process is designed by an open call and standard screening procedures by the stakeholders where startups can pitch their ideas and get approval. The main factor in selecting the startups has been found to be how promising their team is. The funding structure of the accelerator is through receiving funds from their shareholders who might be private investors, companies, and the public, or sometimes an equity stake in the startups. Alumni relations are designed to be maintained through regular events to which alumni are invited or use of the startups as future mentors.

![Figure 3. Building blocks of an accelerator – adapted from Pauwels et al. (2016).](image)

Understanding how accelerators affect startups carries some challenges for researchers for reasons such as the newness of the accelerators and the lack of comprehensive data sources (Cohen and Hochberg, 2014). Research has shown some effects; however, the results are not always empirically well-grounded and, in many cases, there is no clear line to know whether the research findings are based on the points of view of the participating startups or the accelerator managers.

A key impact measure used in many studies has been the increased chance of access to investors for future capital (e.g. Fehder and Hochberg, 2015; Hallen, Bingham and Cohen, 2016). While future capital is inarguably essential for any startup, other effects in overcoming startup’s challenges must also be considered. Studies show that startups can have better networks and become better entrepreneurs as a result of connecting with future investors, technological experts and peers, as well as being disciplined and focusing more on the startup activities after receiving funding (Miller and Bound, 2011). They can use the bonding opportunity with firms in each cohort (Cohen and Hochberg, 2014) while creating a certain cultural capital (Haines, 2014) to learn from each other, and validate their product and concept on a regular basis (Demp-
wolf, Auer and D’Ippolito, 2014), and so increase the likelihood of achieving commercialization (Scott, Shu and Lubynsky, 2015). In addition, startups can be certified and gain a reputation by participating in an accelerator (Miller and Bound, 2011). Consequently, they can expedite their reaching of milestones in revenue creation and getting customer traction (Hallen et al., 2014), accelerate setting their strategy and its deployment (Cohen, 2013), have a better value proposition, team building, and revenue planning (Sharma, Joshi and Shukla, 2014), and finally expedite their exit through acquisition or quitting (Smith and Hannigan, 2015).

Accelerators can be categorized by their goals. Pauwels et al. (2016) classified them into three groups. 1) Ecosystem builders are affiliated with a company to match customers with startups and build a corporate ecosystem. 2) Deal-flow makers introduce startups to investors and find investment opportunities for them. 3) Welfare stimulators inspire startups’ activities and help with their economic development. The program of an accelerator and its performance is designed and measured according to the objective of its shareholders (Bergek and Norrman, 2008; Pauwels et al., 2016). Recent trends in accelerators have gone in two directions: 1) specialization to offer services for specific industries such as health care, and 2) the growth of corporate accelerators, similar to the above ecosystem builder accelerators (Hochberg, 2015, p. 44).

The term corporate accelerator refers to an accelerator a company has invested in to provide time-limited support program for startups with a product in a certain field of interest to the company (without equity investment in them) (Weiblen and Chesbrough, 2015). Historically, the notion of company assistance to new ventures to create products for the company has been utilized. For example, Ford established its own incubator in the late 1990s to expedite the creation and development of Internet businesses to discover new ways of leveraging its assets (Kohler, 2016). A corporate accelerator can be run by a company itself or on behalf of it (Pauwels et al., 2016).

Similar to independent accelerators, corporate accelerators provide numerous benefits to startups in terms of access to resources, markets, funding, and credibility in acquiring funding from investors (Kohler, 2016). In return, the companies can benefit from the innovations that are created by the startups to respond faster to market changes (Weiblen and Chesbrough, 2015) or create an ecosystem around the company (Pauwels et al., 2016). In fact, accelerators allow the company to think out of the box by taking advantage of the entrepreneurial ideas of the startups and exploiting the talents and resources of these startups (Jackson and Richter, 2017).

Despite the recognizable popularity of accelerators and the increasing trend of using their services, academic research on accelerators in general and corporate accelerators in particular are still in their infancy. The extant literature on corporate accelerators focus on the raising concern on the potential tensions that can happen between the startups and the company due to the underlying differences in work practices, modes of operation and cultural differences between large and small firms, and provide advices on to how to avoid
them (Jackson and Richter, 2017; Kohler, 2016; Weiblen and Chesbrough 2015).

### 3.2 Extended Literature Review

This section presents the extended literature, which has been studied in depth due to the emergence of relevant concepts from the data. Firstly, it presents the extant literature on boundary resources which is used to ground the empirical findings in the first core category in Chapter 4. Secondly, it presents the literature on the iterative approach taken by software startups to ground the emerged concepts in the second core category in Chapter 5. Finally, it provides a review of the success of mobile games which is used for grounding the concepts in the core theoretical model in Chapter 6.

#### 3.2.1 Boundary Resources

The concept of the boundary resource originates from boundary objects (Bergman, Lyytinen and Mark, 2007; Star and Griesemer, 1989). A boundary object is any artifact that is shared between multiple actors at the borders of multiple communities to coordinate activities among them (Bergman et al., 2007). A boundary resource is defined as ‘the software tools and regulations that serve as the interface for the arm’s-length relationship between the platform owner and the application developer’ (Ghazawneh and Henfridsson, 2013, p. 174). Therefore, the boundary resource is a relationship mechanism between the platform owner and third-party app developers, by which the platform owner allows the third-party developers to access the platform’s resources while maintaining its control. The platform owner uses the boundary resources to influence and control the platform ecosystem (Ghazawneh and Henfridsson, 2010). The third-party developers use the boundary resources to create apps for the platform. Boundary resources aim to ‘minimize the coordination effort’ between the platform owner and the third-party developers (Bianco, Myllarniemi, Komssi and Raatikainen, 2014). Consequently, boundary resources can facilitate shared understanding between third-party developers and the platform owner in the design of apps (Mohagheghzadeh and Svahn, 2016b).

Boundary resources can have technical and social attributes due to the socio-technical setting of the platform ecosystems. Technical boundary resources are further divided into two groups: 1) application boundary resources that are used by the apps directly and enable the app to utilize the services of the platform (e.g. API); and 2) development boundary resources that enable the developers to create the applications (e.g. SDK). Social boundary resources relate to organizing the actors by incentives and rights to the intellectual property and transferring knowledge between the developers and the platform owner or between developers to enhance the understanding and/or coordination of the platform. (Bianco et al., 2014; Ghazawneh and Henfridsson, 2010; Rudmark and Ghazawneh, 2011.)
The boundary resource model developed by Ghazawneh and Henfridsson (2013) explains the relationship between the platform owner and the app developers through the processes of design and use of the boundary resources to resource and secure the platform (Figure 4).

**Figure 4.** The boundary resources model developed by Ghazawneh and Henfridsson (2013, p. 177).

The process of resourcing refers to extending the scope and diversity of the apps on the platform. The platform owner resources the platform by providing facilitations (e.g. introduction of new APIs). Developers can also resource themselves by their initiatives when they see limitations in the existing boundary resources (e.g. the jail-breaking of iPhone devices before Apple opened up its platform to developers). Securing is the process of increasing the control to protect the platform sovereignty by regulations (e.g. Apple’s review process before publishing an application on its platform). So, the essence of boundary resources is about attracting third-party developers to the platform while obliging them to act according to the platform rules and objectives (Rudmark and Ghazawneh, 2011).

Designing boundary resources has been a topic of interest for researchers. They have found that the platform owner and the third-party developers may have different opinions regarding the implications of a boundary resource (Mohagheghzadeh and Svahn, 2016a). Therefore, to be effective, the platform owner needs to create the boundary resource according to the developers’ perspectives and for them (Bianco et al., 2014). It has been found that incompatibility and lack of timely communication between the platform owner and third-party developers in setting boundary resources can have a negative impact on the developers’ processes and may cause the developers to fail in running their applications on the platform (Rafiq, Ågerfalk and Sjöström, 2013).

Some studies have considered the dynamic nature of boundary resources. The early definition of the boundary resources by Ghazawneh and Henfridsson (2010) implied that the design of boundary resources was limited to the deci-
sion of the platform owner. However, further studies show that boundary resources have been shaped through interactions between third-party developers and the platform owner. Eaton et al. (2015) refer to the interaction as a tuning process that is performed via each developer accommodating or resisting the existing boundary resources. Furthermore, Mohagheghzadeh and Svahn (2016b) find that in addition to the interaction between the platform owner and the third-party developers, the internal resources of the platform owner organization influence the shaping of the boundary resources.

The existing research on boundary resources mostly focus on the perspective of the platform owners and the technical boundary resources (Schreieck, Wiesche and Krcmar, 2016). However, in the socio-technical context of digital entrepreneurship on the platforms, social boundary resources play a great role in orchestrating innovation as well, especially given that the ecosystem members may pursue conflicting goals (Yoo et al., 2010). Therefore, understanding the social boundary resources as the means of interaction between platform owners and third-party innovation creators can benefit the platform owners in managing this strategic resource for maintaining competitive platform ecosystems.

3.2.2 Iterative Approach in Software Startups

Iterative approach in software startups is closely linked to the deployment of lean approach in the startups. As mentioned in Section 3.1.2, lean approach has a broader perspective than the software development processes, such as agile development methods, and rather suggest a philosophy of work in the entire startup organization.

Scholars and practitioners have been tempted to create step-by-step guidelines or models to work according to lean approach. For example, Maurya (2010) in his book *Running Lean* defines three steps to building web-based software: documenting the initial plan, identifying the most risky parts of the plan, and systematically testing the plan, through which the attention is on the overall business as well as the product. In each stage, he firstly identifies and prioritizes the risks in three areas of product risks, customer risks, and market risks, and then applies the build-measure-learn loop. In the same vein of operationalizing the lean principles, Bosch et al. (2013) introduce the Early Stage Software Startup Development Model (ESSSDM) with the aim to select one product idea among many ideas based on customer feedback. The model provides guidance for when to move a product idea forward, when to abandon a product idea, and how and when to validate the product ideas.

In more recent research, Fagerholm et al. (2017) introduce the Rapid Iterative Value Creation Gained Through High-Frequency Testing (RIGHT) model for continuous experimentation. They highlight the importance of experimentation to reduce uncertainties through iterations of the BML cycle on every hypothesis about the product. Their model suggests starting from a hypothesis and leading to making an MVP, which is then tested through data collection from customers. It then suggests comparing the results with the strategy of the firm, and finally deciding to either preserve or pivot. Along the way, the model
highlights the importance of having a technological infrastructure that allows for data collection and the deployment of these steps. The concept of iteration in startups applies not only to the product itself, but in a broader perspective to finding an appropriate business model. A widely accepted definition of the business model is (Osterwalder, Pigneur and Tucci, 2005, p. 10):

a description of the value a company offers to one or several segments of customers and of the architecture of the firm and its network of partners for creating, marketing and delivering this value and relationship capital, to generate profitable and sustainable revenue streams.

Accordingly, Osterwalder and Pigneur (2010) define the business model canvas, with nine building blocks that a firm should define for its business: key partners, key activities, key resources, value propositions, customer relationships, channels, customer segments, cost structure, and revenue streams.

Defining a sustainable, well-structured business model with attention to scalability is a challenge for software startups (Abrahamsson et al., 2016). In a study on digital entrepreneurs in 16 case companies, Standing and Mattsson (2016) find that software entrepreneurs are competent in recognizing the opportunities to create a business, but inefficient in defining their business model. They conclude that startups working in digital environments must have knowledge about the online environment in addition to technical knowledge. In the same vein, Vanhala and Saarikallio (2015), in their study on game startups, extend the building blocks of the business model and argue that the nature of software startups requires different building blocks on a business model than those defined by Osterwalder and Pigneur (2010). They discuss that in software startups, special attention should be paid to both the initial funding and the human capital as a main resource either in the form of developers or customers as references.

While the mentioned studies have a static view toward business model components, other scholars have tried to understand how the business model is shaped. In a study on 16 startups offering digital services, Standing and Mattsson (2016) discuss that startups develop a workable product and a user community by maintaining an iterative perspective and continuous learning. Similarly, Ojala (2015) conducted an in-depth longitudinal case study and highlights the need for an iterative assessment process to experiment and learn about the market need in order to find the best business model. Hence, the startup can follow technological and market changes by abandoning unnecessary components of its business model. Further, Spiegel et al. (2016) show that business models in the Internet startups are dynamic and in permanent flux around the startup founders’ core visions. Bosch (2012) also highlights the experimentation with the business model and mandatory evolutionary approach in software offering.

In the field of mobile app developer startups, researchers have found various styles of the iterative approach. Vanhala and Saarikallio (2015) find that game startups work in an ad-hoc innovative and artistic style, and make decisions
around a ‘coffee table.’ Davis et al. (2014) highlight that game developers have two methods of experimenting with their games: simultaneous vs. sequential. Simultaneous experimentation is when the startup publishes a portfolio of apps to the market simultaneously and analyzes their reception. In contrast, in sequential experimentation, the startup improves its offered app gradually with sufficient time between each version according to user feedback. They suggest simultaneous experimentation as a suitable strategy for game applications, and sequential experimentation as an appropriate strategy for non-game applications.

3.2.3 Success of Mobile Games

Researchers have considered several measures of success for mobile apps separately or in combination, including the volume of downloads (e.g. Carare, 2012; Garg and Telang, 2013; Ghose and Han, 2014; Liu et al., 2014), the volume of revenue (e.g. Davis et al., 2014), and the average user rating (e.g. Liu et al., 2014). These measures have been treated both as dependent and independent variables in different studies. Song et al. (2013) shows a positive relationship between the user rating and number of ratings with the download volumes of an app. Garg and Telang (2013) find a negative effect of price and a positive effect of ranking list position on app download numbers. Lee and Raghu (2014) find that offering a free app, placing the app in less popular categories, updating regularly, and receiving a high volume of user reviews and ratings can positively impact an app’s sustainability on top-grossing ranking lists in the App Store. Ghose and Han (2014) discuss that in-app-purchases can increase demand for an app, while in-app advertisements can decrease demand. In their earlier study, Ghose and Han (2012) found file size, number of apps by the same developers and pricing to have an impact on app downloads. Kajanan et al. (2012) studied the success of mobile apps, defined as staying on top-rank lists, in the light of positioning on the app store, the developers’ actions, and user engagement and quality perceptions. They find that the number of categories that an app belongs to, a larger device base, and more frequent updates increase the period an app stays on the top rank, and that negative user reviews have a negative effect on its chance of rapid growth. In another study, Diya and Sangaralingam (2016) find an effect of time of release on download volumes and show that weekends are optimal for releasing for high downloads.

For mobile games, the success measure includes the above-mentioned factors, as well as continued use for freemium games. For example, Ponnada and Kannan (2012) consider the success measures of download volumes, user ratings and reviews, and pay per download as success measures in their study of four successful mobile games. In freemium games, continued use has a particularly positive effect on the purchase of in-app items (Hamari, 2015). Continued use, in turn, is a consequence of enjoyment in hedonic systems and gaming contexts (Hamari, 2015; Merikivi et al., 2016; van der Heijden, 2004). Moreover, enjoyment is a result of various factors, such as the game design attributes (aesthetics, content, ease of use, novelty, etc.), social attributes (so-
cials, norms, connecting to peers and reputation, etc.), and the context of use (place of use, mobility, etc.) (e.g. Hamari and Järvinen, 2011; Hsu and Lu, 2004; Merikivi et al., 2017; Okazaki, 2008; Okazaki et al., 2008; Zhou, 2013).

Thus, putting all the above-mentioned success factors for both games and apps together, it is arguable that the measures such as the visibility of the game and its quality (Garg and Telang, 2013; Ifrach and Johari, 2014; Liu et al., 2014), as well as its social attributes for the existence of other users (Tang, 2017; Yi et al., 2017) can relate to the download and continued use of the game (referred to as user engagement in this study). These measures and their effect on downloads and continued use are explained in more detail below.

Visibility

A number of studies have shown the importance of visibility for the download of an app. In these studies, visibility is enhanced by the presence of the app on the store ranking lists (Bergvall-Kåreborn and Howcroft, 2011) and is measured by the rank of the app (Liu et al., 2014). Garg and Telang (2013) show the significance of the position of an app on the ranking list on its download volume in the Apple App Store. Ifrach and Johari (2014) find that moving from position 20 to position one in the Apple App Store could nearly double demand for the app. Carare (2012) discusses that users are willing to pay more for a top-ranked app compared with the same app unranked. Given the effect of the ranking position on app downloads, developers are incentivized to get on the top of the ranking lists (Bresnahan et al., 2014).

The algorithm for getting on the ranking lists is officially unknown (e.g. Yi et al., 2017). Academic studies on the ranking lists have taken their samples from apps that have already existed in the top lists, and only tried to find the factors that could affect the ranking position. Examples include the impact of the price, reviews, app size, and number of developed apps by a developer on the ranking list position of their apps (Jung, Baek and Lee, 2012; Lee and Raghu, 2014). Yin et al. (2014) find that a developer’s later games are more likely to get on the top-grossing ranking list in Apple App Store, and also that updating a game could negatively affect the likelihood of it getting on a top-grossing list. Qualitative data collected by Bergvall-Kåreborn and Howcroft (2011) reveals that the number of downloads is important for getting on the ranking lists in the eye of app developers. Eventually, Duarte and Picoto (2016) conclude from various Internet sources a few of the most important factors for getting on the ranking lists, such as the volume of the recent and aggregated downloads and revenue, the frequency of app usage, the number and value of user reviews, keyword relevance, and updates.

Although the visibility measure is studied in terms of ranking list positions, other factors such as making the app visible through marketing have also been slightly noticed by researchers. Ghazawneh (2016) highlights the importance of marketing due to increased competition. Bergvall-Kåreborn and Howcroft (2011) show marketing as part of app developers’ responsibilities. Qiu et al. (2017) find that Apple App Store developers use peer-developers to market their apps at release time.
Quality

The construct of quality has traditionally been a point of attention in IS research. The widely used model of IS success considers quality in three contexts: system quality (adaptability, availability, reliability, response time, usability), content quality (completeness, ease of understanding, personalization, relevance, security) and service quality (assurance, empathy, responsiveness) (DeLone and McLean, 2003).

In the context of mobile apps, scholars have paid considerable attention to the quality of apps and have related quality to issues such as the technical design and performance, security, and the possibility for users to test before purchase (e.g. Ghose and Han, 2012; Jansen and Bloemendal, 2013; Mei Li, Goh and Cavusoglu, 2013). It has been found that the availability and quality of apps on a mobile platform is as important as the price and service plans in the user’s decision-making to purchase mobile devices (Kim, Kim and Lee, 2016). Moreover, the developer’s experience with app development and publishing has a positive influence on app quality (Duarte and Picoto, 2016).

Despite the importance of the app quality, the construct of quality is not comprehensively defined in the extant literature, and the definition is taken for granted. That may be due to the notion of quality in traditional IS systems research being understood to address system and content quality. However, scholars have widely taken into account the reflection of the app quality in users’ reviews and star ratings of the apps in assisting potential users in their judgments (e.g. Ghazawneh and Mansour, 2015; Rollin et al., 2017). Li et al. (2013) view the valence of user ratings as an indicator of the app quality, and Davis et al. (2014) consider user ratings as their measure of app quality.

The significance of users’ reviews suggests that the quality of an app is defined beyond the technical quality and, in fact, is defined through users’ perceptions of quality. Furthermore, the attributes of the market to allow updating the app and purchasing in-app items in freemium models have created a service-oriented market. Mobile games, for example, are not offered solely as a product to attract customers on the store shelves and to be sold in a one-time payment, but instead they are offered in the form of a service where the interaction with customers is ongoing throughout the game usage (Hamari and Järvinen, 2011; Hamari, 2015). Consequently, it is arguable that rather than product quality measures, service quality measures should be considered in understanding the construct of quality in games. Hamari, Hanner and Koivisto (2017) took this into account and used the SERVQUAL (Parasuraman, Zeithaml and Berry, 1988) measure to study why people use freemium services in games. Following Hamari et al. (2017), I also consider the metric of SERVQUAL to identify the factors that can be related to game quality. Table 6 illustrates the SERVQUAL measures.

Table 6. SERVQUAL measures — adapted from Parasuraman et al. (1988, p. 23) and Harari et al. (2017).

<table>
<thead>
<tr>
<th>Tangibles</th>
<th>Physical facilities, equipment, and appearance of personnel</th>
<th>Appealing nature of the game content (e.g. visual, audio)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability</td>
<td>Ability to perform the promised service</td>
<td>Technical performance of the game</td>
</tr>
</tbody>
</table>
The service quality measure of SERVQUAL is the originating root of the construct of ‘service quality’ in the widely accepted IS success model (DeLone and McLean, 2003). The reason for using the earlier SERVQUAL rather than the more recent ‘service quality’ in the IS success model, is due to SERVQUAL’s more comprehensive definition of service quality. The construct of ‘service quality’ in the IS success model is limited to only the relationship with the users; however, game quality also includes the quality of the game content. Research shows that factors such as the design aesthetic, novelty, and perceived ease of use are found to be important in the user’s enjoyment of the game (Merikivi et al., 2017). Consequently, the SERVQUAL measure is thorough enough to address the technical build, as well as the content and service characteristics, in the mobile game context. The SERVQUAL measures are about the customer’s perception of quality, which is defined as their comparison of the quality with their expectations, and is different than objective quality (Parasuraman et al., 1988). Considering that in the app, context quality is eventually reflected through user reviews and star ratings, using SERVQUAL measure makes sense.

Social Attributes
Social attributes are found to be important in the success of games. In a study on the success of the Pokémon GO game, Tang (2017) discusses how the nostalgic, novel, and social attributes of the game explain its success. Furthermore, Yi et al. (2017) find an opportunity for remaining in the top position on ranking lists by connecting the game to mobile messengers – hence social factors.

This study positions the social factors as a component in the tangible offers in the game’s SERVQUAL measure, through which developers can offer options to enhance social attributes by either multi-playing attributes or connecting the game to social media. This makes an avenue for developers to leverage monetization of the game as well (Hamari et al., 2017). In addition to the psychological aspect of the social attributes, the existence of more users for the game and their engagement with the game creates a virtuous circle of getting more visibility. Through the market mechanism, the number of downloads feeds the possibility to get on the store ranking lists (Bergvall-Kåreborn and Howcroft, 2011). In addition, users’ positive reviews and star ratings and creation of word-of-mouth (WOM), can attract more users to the game and result in more downloads (Liang, Li, Yang and Wang, 2016). Therefore, mobile games need to have satisfied users who keep playing the game and leave positive reviews and star ratings.
User Engagement

User engagement as a construct has been used in mobile app studies to imply the use of the app. For example, Kajanan et al. (2012) see user engagement and app quality as important factors in the success of an app. In the context of games, engagement has been used in various ways, such as the point of making the decision to play a game, or as the process of immersing in the game during game play (Mengxiang Li et al., 2014), which is close to the concept of the flow experience (Csikszentmihalyi, 1990) in the hedonic system. In the context of a mobile app, the point of decision-making to play a game can start from the point of downloading the game to playing it. However, limiting the construct of engagement to either of these two definitions does not serve adequately to reach success with the game. Engagement referring to the point of decision-making to download the game is not fully inclusive, because in the general app market, and particularly with free apps, it is common for users to download and uninstall apps fairly easily (Kajanan et al., 2012). On the other hand, in the context of mobile games, mobility plays a major role in interrupting playing. In fact, as long as the user keeps using the game, the engagement is sustainable, even if the context of mobility may require the user to interrupt playing the game. In such a context, the sustainability of the engagement is a significant factor, especially when a freemium business model is employed which makes the continued play of the game (even at different intervals) valuable for developers (Hamari, 2015; Koskenvoima and Mäntymäki, 2015).

Taking a broader perspective, one can argue that the construct of user engagement is not limited to the decision-making point of use or the actual use period like the flow experience, but in fact it can be seen as a circular process. To exhibit this process and clarify the user engagement construct, O’Brien and Toms (2008) deconstruct engagement into four phases: the point of engagement, the period of engagement, disengagement, and reengagement. Following the widely accepted definition by O’Brien and Toms (2008), and for the matter of simplicity, I use the term user engagement in a way that is not limited to immersion with the game without any interruptions. Instead it considers the process of the engagement as long as the user plays the game, even if it is interrupted. In other words, the engagement process in this context consists of the point of downloading the game (the point of engagement), to playing the game even with some interruptions (the period of engagement and reengagement), to fully stopping playing the game (complete disengagement). Consequently, the engagement ends when the player is not playing the game anymore and may even uninstall it.

This definition of user engagement makes sense for mobile games given the mobility attribute and various contexts of use (Liang and Yeh, 2011; Liu and Li, 2011) as well as the nature of mobile games to help users ‘kill time’ or to have quick enjoyment (Okazaki et al., 2008, p. 846). Therefore, as long as the user comes back to play the game, the developer can gain value from the retention of the user. It should be noted that some game developers purposefully force players to take time off and disengage from game play before they can return to playing the game. An example is the game Candy Crash Saga where
users are forced to stop playing for a period of time unless they got help from other players through virtual goods in the game.

Furthermore, the point of the engagement and the period of the engagement are also in line with the visibility and quality of the game by the time of encountering a game and/or during its play. For example, the game icon and screen shots in the store are important factors to motivate users to download a game (Lee and Raghu, 2011; Mohd, Noor, Daud and Hasbullah, 2016; Wang and Li, 2017). Therefore, the first step to engaging users is to have a quality presentation on the app store page or any place where users encounter the game for the first time. The period of engagement can then be prolonged by the quality of the game.

A longer period of engagement paves the way toward revenue-making by implementing the monetization mechanisms in the game, for example through showing advertisements or features for purchasing. In fact, the quality measures of SERVQUAL include creating empathy with users and offering tangibles in design such as easier continuation of the game play, better social interactions, and more content that motivates users to pay and helps in monetization of the game (Hamari et al., 2017).
4. Utilizing the Accelerator’s Facilities

In this section, I present the empirical findings of the first core category, ‘Utilizing the accelerator’s facilities’. It is comprised of six selective codes: 1) Startup’s capabilities, 2) Market-related activities, 3) Experimenting activities, 4) Resources, 5) Game design activities, and 6) Accelerator’s facilities. Table 7 illustrates the constructs of the core category through its selective and open codes.

Table 7. Construction of ‘Utilizing the Accelerator’s Facilities’ through open and selective codes.

<table>
<thead>
<tr>
<th>Open Codes</th>
<th>Selective Codes</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Company profile; 2) Experience with game development; 3) Lacking business knowledge; 4) Lacking financial resources; 5) Reflecting philosophy.</td>
<td>Startup’s capabilities</td>
<td>The state of the startups in regard to their challenges and capabilities</td>
</tr>
<tr>
<td>1) Soft-launching; 2) Experimenting with AppCampus game; 3) Experimenting with previous game</td>
<td>Experimenting activities</td>
<td>Learning-by-doing activities to cope with lack of knowledge</td>
</tr>
<tr>
<td>1) Exploiting team member’s capabilities; 2) Accessing low-cost resources</td>
<td>Resources</td>
<td>Use of available resources to perform game developing and publishing activities</td>
</tr>
<tr>
<td>1) Differentiating the game, 2) Integrating social aspect in the game, 3) Developing the game, 4) Avoiding complex game design, 5) Designing for monetization, 5) Caring about getting user star ratings and reviews, 6) Paying attention to the game quality</td>
<td>Game design activities</td>
<td>Issues considered in game design and development</td>
</tr>
<tr>
<td>1) Getting invited by the platform owner, 2) Benefiting from funding, 3) Learning through AppCampus training, 4) Benefiting from networking in AppCademy, 5) Building a relationship with the platform owner, 6) Building a reputation through AppCampus</td>
<td>Accelerator’s facilities</td>
<td>The ways through which developers use the accelerator facilities</td>
</tr>
</tbody>
</table>

In the following, firstly, I explain the selective codes and the open codes that are constructing them. Subsequently, I demonstrate the relationships between the selective codes, and finally ground the findings by using the extended literature on boundary resources which I presented in Section 3.2.1.

4.1 Startup’s Capabilities

The selective code ‘startup’s capabilities’ is comprised of five open codes: 1) company profile, 2) experience with game development, 3) lacking business knowledge, 4) lacking financial resources, 5) reflecting philosophy.
**Company Profile**

The open code ‘company profile’ describes the profile of the companies, their age, their roles, the company team, and how they are dedicated to game development.

The core team for most of the startups was created by people who knew each other as colleagues, friends, study peers, or family members, and found that they could work together to develop games. In the case of not having the team members already, there was a more difficult path of finding like-minded people who wanted to invest their time and energy into creating a business, and the process of team development took longer and faced more challenges.

“...I started to do lots of interviews with potential designers, potential programmers, potential graphic designers and everything; and I suggested to them, ‘I have this idea, I want to do this, I want to publish this game, do you want to join me?’ And I showed them all the stuff and I was able to join eight people with no money to create a two years’ project which was [previous game], but after those two years, as you can imagine, lots of things happened; because people were expecting to go to market before, people wanted to make cash easier and faster than expected. [...] So, we re-joined new people with a much better product, because we had better products than any at the moment; so, much easier to join new people.” (Startup Q)

“We have done some games earlier; so, [as a] school project and then – let’s say – as a side hobby or something – not so serious – but the commercial, [this game] is the first commercial.” (Startup B)

The startups were in the range of less than a year to five years old and considered the current game as their first serious commercial game on the platform. Some developers had previous experiences with browser games or even mobile games, but they did not have a commercial view toward the previous games and had not monetized them. There were cases where the company had previous non-game apps, but they were not the owner of the app, and had only technically developed it. In only one of the startups (M), the developer had previous commercial experiences with other platforms, but the WP platform was a new platform for them and they were developing their first game there. Overall, for all the startups, designing and publishing the game on the WP platform and taking care of the business issues of game development was a new challenge.

“...Well, back in the Flash era, you could make games really, really fast [...] – and the lifetime of the games wasn’t so short. So, I made many games. I made like 20 games back in Flash and made a few more games here in the Windows Phone [...] , but I didn’t monetize it very well. Now, it’s this game I make right [...], I improve the monetization.” (Startup H)

The team members’ roles in the companies were quite flexible and due to the small size of the companies, most of the team members were multi-tasking. The main roles in the companies were related to programming and coding tasks, as well as marketing and business-related issues related to running the
company. However, in most cases people were not officially trained for the business and marketing tasks. Most of the companies had to outsource part of their development process, especially for tasks related to graphics and music.

“We are doing the design together; so, everybody is a designer – like the game design part. I’m doing, like, trying to run the company, which takes a lot of time, because I don’t have a business background. [...] I know a bit of everything; I might help to do some animations and stuff like that, but then [team member name] [...] supervises all the 3D stuff that me and the third guy [team member name] does.” (Startup B)

“We have a team of four people: me, I am the lead programmer and I am [the] team lead, so, I control the team. Then, we have our designer, and then we have other programmer; he is mostly specialized in our back-end programming, because we have statistics and analytics and everything. Then we have our also graphic artists, which is great.” (Startup N)

Furthermore, in most of the companies, working part-time was a means to earn revenue from other sources or to continue studying.

“We also study and make this game. [...] Since I’ve got the funding. Back in university, I was taking a bigger amount of classes. Now I’m taking a smaller amount of classes, and I’m making the game. I’d say like, 60 percent, yes, 65 percent of [...] my working time, I would say.” (Startup H)

“So, we’ve been in existence for about a year as a studio, but this is like our passion project that we do on weekends and nights, and then we work [on] other jobs, on other games, on contract – working for other studios.” (Startup C)

Experience with Game Development

The open code ‘experience with game development’ describes the startup’s learning from their previous game development experiences. Their learning was mainly related to business issues of game development; however, in some cases they highlighted learning technical issues as well.

The startups mentioned their learning about getting a higher user rate through their previous experiences, such as requesting ratings and localizing the game for several geographical markets. One of the startups (H) found that having a game that was translated into the language of the target market had a great impact on acceptance of the game by users and publishers, and the chance to get more rates and reviews.

“I noticed one thing; if the game is localized, [...] there is three times more chance that the player will rate the game, because it feels more natural – I think. Because they’re more likely to write a comment in their own language. So, they’re more likely to rate it. Also, in the game, there is the message requesting a review. So, if the player can read it, then it’s more likely they will be able to give a review.” (Startup H)
“Actually, for example, we made a mistake when we released the game; because we didn’t have the Rate My App.” (Startup L)

The startups shared their experiences of the differences in platforms, and how they worked on their game concept and quality according to their target platform. They learned that the lower competition on the WP platform allowed for easier revenue-making though offering less innovative games, but on mature platforms such as iOS and Android, providing a more outstanding and innovative type of game was more likely to make revenue.

“Well, it’s quite easy to do it cross-platform. But […], you have to stand out on Android; otherwise, you won’t make it. Same for iOS. [on] Windows Phone, with quite common titles or quite common things, you can do a decent job. At least, earn more than beer money. You can be drunk every day.” (Startup D)

“There’s some revenue from advertisements, but because the Windows Phone platform is not so, like, popular, at least, the advertisements don’t like to advertise on Windows Phone. So, you don’t get paid that much. It’s like 10 percent of what you get from Android. So, we are not generating much revenue; maybe about $20 per day. It’s pretty low.” (Startup N)

Startups also found it essential to pay attention to user acquisition and retention. They highlighted the importance of seeing the game development as a service for the users. They emphasized the importance of keeping the users constantly involved in the game in various ways, such as updating the game regularly, embedding social factors in the game, and paying attention to the expectations of the users rather than their own preferences.

“It is a wonderful job to do games, but we have to think of it as it is our service for users. You have to update it many times – maybe two times per month – with new features and you have to involve them on social sites. It’s a very important start to build the game and to develop the game with social features. When you start the first line code, you have to put something about Facebook or Twitter to help them later to spread your game and the word and to share it; it’s very important. Also, you have to think about the marketing side when you start to develop the game.” (Startup G)

“Make it easy for kids who actually don’t have patience anymore to play games […]. So, make it much easier; don’t lose like … let them just finish the damn thing and they will be happy. They will lose twice and they will say ‘What? I keep losing!’ So, I learned from [the previous game] – it was a beautiful game – you should look at it.” (Startup O)

Lacking resources was a drawback for unnecessary investments of any kind for the startups. Some learning from previous experiences meant they were concerned that they had to allocate resources properly to essential activities.

“The first mistake [that] we made [was that] we started to develop our own game engine. So, we developed the game engine, not developing the game. We
put a lot of effort and time to make the game engine, but not the game. I think that’s the first mistake. That’s why we started using Unity. Second, [...] we tried to make the game of our dreams, and we tried to make a lot of features there. So, it was like seven different game play mechanics. So, it’s a really big mix of different game types there. It was cool, it was great, but it takes a lot of time to develop and test it. Try to make the demo version of the game a little bit easy for the gamers, and yes! For example, don’t use voice dubbing in your game, because it’s really hard to translate to different languages.” (Startup R)

“The development process – it will be much faster because we know what works, and what doesn’t. We made a lot of small mistakes in the development process. In everything, we have improved.” (Startup N)

Startups had also learned about technical issues such as improving the development process and project scope, and learning about development environments.

“For me it was also getting to know Unity as a game engine, because [programmer name] would come in to code for a month; but that was when we had everything – like the design ready to go. But all the prototyping was just us trying to figure out GPS and which GUI system do we use in Unity? So, I got to know the engine much better, which then opened the door for me to make [the current application]. [...] My mind clicked: ‘Oh, I can use Unity this way! cool!’ Then now, I can code up this entire game in four months.” (Startup C)

**Lacking Business Knowledge**

The open code ‘lacking business knowledge’ describes the technical orientation the startup founders gained through education or work experience and their lack of knowledge about the business side of app development.

The interviewees mostly had a background related to software development and/or computer science gained by work experience or academic studies. For startups, the path toward becoming a game developer was through firstly studying at university and then getting interested in the field. In contrast, for others, the path started from getting interested in game development and then equipping themselves with the relevant educational background and technical knowledge. Overall, all the interviewees had been passionate about game-playing and game development and saw their activity not as a job, but as a personal interest that they sought the opportunity to fulfill.

“I always wanted to study computers to make video games. So, that was my main goal.” (Startup O)

“My dream has always been to create video games [...] Actually, I was saving money because I wanted to travel the world. [...] I’m also a game programmer. So, of course in my field I know all the things about gaming; and so, I discovered about the game founders [in one of the cities]; and I was working with my previous colleague on another indie project and I told them, ‘Yeah! Guys, what do you think?’; and they say, ‘Yeah! Why not?’ ‘Eh, OK!’ [short giggle] and everything started.” (Startup L)
A common trend among the startups was that they did not initially plan to become professional game developers but eventually moved from hobby development toward commercial development.

“This kind of started like a project in my spare time. Like I said at the beginning, I got training, got a phone, and started playing with it; and so, when I started it, it was not the idea [that] ‘Well, OK, I’m going to start a game business.’ It was just releasing a game and seeing how well it goes [...] That day, I didn’t have a lot of work [and] interesting things to do. So, I found the training; started to do some development in my spare time; saw that it was pretty successful; got a second game that was a little bit more successful; and in the end, I said, ‘I’m going to try it full-time.’” (Startup D)

The leading factors for this move included several issues, as explained in the following. Some interviewees had discovered their capability to develop a game in project assignments at school or at work, and consequently decided to start professional game development. For others the discovery was about finding the like-minded people with whom they could work.

“I was studying in the Aalto School of Arts and I started game design and production three and a half years ago. There, we slowly found out the team, and when we did the project with the guys that we now build up the company – like three people from that course. We found out that, ‘OK, we can do games with three of us,’ and it seems like a better option than to go into Finnish companies to do game design or graphics.” (Startup B)

“We didn’t have anything game-related when I studied [...] Anyway, I attended this summer course where we tried to make a mobile game in three months. That was after we made a funny [...] game in a game jam. After that, it seemed that we could really do it, if we could — if I found the right people.” (Startup P)

For several startups, an unexpected success with a hobby or educational app development in receiving a large number of downloads was the motivating factor to start game development professionally. For others, the evolving app market through which they could bring a new idea or innovation to market was a main factor in their move toward professional game development.

“We realized that even if we make funny apps – which we develop in one evening, for example – they get a lot of downloads there, because the market was free at that time. So, we decided to pay close attention to quality and try to make better apps. [...] We just asked the question for ourselves, ‘What if we make a really good app? Will it be worth a lot of downloads then? More than 10,000?”’ (Startup R)

“The handset device is usually very cool for me, because it’s personal and you can be close to the person in terms of experience and digital experience, or interactive experience; and one of my main focus – when I was developing apps for web – was the user experience [...] I thought that mobile would be great. So,
I started to focus on this when the mobile gold rush [started]. That’s the little bit reason why I am in the mobile apps sector, because of that!” (Startup Q)

Last but not least, for some startups, the entrepreneurship opportunity was the leading factor for entering the app development business.

“I was studying business administration at the university and I watched the Keynote presentation about the iPhone and the App Store, and how could anyone build an app for the App Store [...] [In] this presentation, they say that they already had 100 million users with credit card accounts; already done with one-click purchase. So, like, ‘Wow! This is a really huge market. If we manage to only have one percent of that, we would be rich!’” (Startup M)

**Lacking Financial Resources**

The open code ‘lacking financial resources’ highlights that the startups had scarce financial resources which prevented them from hiring experienced people to take care of the business side or complement the technical expertise. Consequently, they faced difficulty in terms of having enough human resources.

To cope with the financial challenges, the startups did multi-tasking and used their own time and energy. Many of them were working on external contract jobs as well.

“I was the only one who was ever employed full-time with the AppCampus money; just because I can do art and code and design all at once. So, we didn’t need too many other people.” (Startup C)

“I’d like to have money and to hire somebody who can dig through all that [game analytics] data and come back with actionable points. Super tough! Particularly when you waste a day just looking at the data – it’s crazy.” (Startup F)

“I’m a one-person company. [...] At least, I need a much stronger link between Facebook and the game. I’ve really just used Facebook largely as just a means to just communicate with [users] and not as a marketing [tool].” (Startup K)

**Reflecting Philosophy**

The open code ‘reflecting philosophy’ describes the philosophical points of view of the startups in their development and publishing activities.

Most of the interviewees expressed their desire to create a unique experience for the users rather than merely offering entertainment. For example, they wanted their users to experience feeling smart, to be distant from the current status and to get lost in the game atmosphere, and/or to educate the game players about some issue of concern. For interviewees who had artistic backgrounds, the desire to create something meaningful was particularly strong.

“My idea was that really, [...] I want to make them [i.e. people] feel smarter all the time. That would make them feel good [...] We all need to control something
When you go and play, you play for – depending of course which level you are – but, if you’re on a good level, then you play it. You release this control. So, you don’t need to control your wife, or anyone else. Then it builds up again and then it has to go on. So, this kind of – I wanted this feeling, for people to feel when they play my game. So, when they play my game they lose, they release their control and they relax; [they] can see the world, you know. Maybe they become wiser in a way. Then they come back.” (Startup J)

Besides reflecting their philosophies in the game idea, the interviewees were keen to follow their moral principles and values in running their businesses and using the opportunities, even at the cost of not capitalizing on some monetization or marketing opportunities and/or taking a more difficult path.

“[For monetization] I don’t want to frustrate players, like, ‘You have to wait ten hours or pay us money.’ [...] You know you’re going to get money with it, but it’s against my own principles; and I think in the end, it’s better to build a sustainable community of people that really appreciate the games you’re making, and people really like that it’s free – that you can play it for free. But, yes, there’s a couple of things that you can spend money on.” (Startup I)

“For example, I don’t like what Candy Crush is doing. I don’t like it. This is – I like it in the way that they know how to crack you in psychology [...] They do all kinds of manipulation on you to pay. I don’t like that in my games, because I think my game is different. I want to manipulate you to be smarter, or whatever, you know, but not to give me your money in a way.” (Startup J)

For some startups, game development was part of their lifestyle, so they would rather focus on their passion of development, without heavy concentration on the business side of game development and thoughts of business expansion.

“It would be nice to earn some money, but that’s not why we do what we do in this industry. We do it because we love doing it; and I think – and a lot of developers will say the same – that for us, this is a way of life. It’s not a job. This is what we do and if you try and just make a game with just monetization thought in it all the time, it doesn’t work. It’s not what making games is about. Making games is about making things that people enjoy playing.” (Startup E)

“We don’t have plans to grow as a big company. We want to keep our design pretty simple and small, so we can spend more time doing the games than managing a lot of people. Not like getting on really deep game mechanics that needs a lot of programming.” (Startup B)

The startups had strong opinions about their games. In some cases, they had planned a very ambitious game in the first place which they could not follow through due to their limited resources. However, they had very positive opinions about the potential of their games in attracting users.

“It’s actually a very easy game to sell, because the art itself is so beautiful and so unique as opposed to a generic-looking game where you then have to [be] like, ‘No, OK, but here’s how the mechanics work. You do this and then this.’ It’s a
much harder sell. For this [i.e. our game] it’s easy. I think we’re lucky that we have a much easier marketing and PR job to do.” (Startup C)

“The time limit was like, I’m running out of time; and I was trying to work as fast as I could to catch up. So, I had to throw a little bit of the design in order to make it in time. So, it’s actually a much bigger game in the original design, but now I’m following up.” (Startup O)

Summary of ‘Startup’s Capabilities’
The selective code ‘startup’s capabilities’ indicates the state of the startups in regard to their challenges and capabilities. The startups were less than five years old and considered the current game as their first commercial game on the platform. They were created by people who had moved from hobby development into commercial development. The core team in each startup knew each other from previous shared activities. They were passionate about game-playing and game development, and keenly followed their philosophical views to create an enjoyable experience for the users, even at the cost of not using certain methods of monetization.

The two main problems in the startups were a lack of business knowledge, and a lack of financial resources. The lack of business knowledge was reflected in their neglect of the tactics for user acquisition, user retention, and monetization in their previous hobby games. The lack of financial resources was reflected in their challenges in hiring people with the required expertise. To compensate, they tried to access low-cost resources through multi-tasking, using family and friends, taking interns, and outsourcing to low-cost services. Furthermore, they were working part-time in the company in order to have other sources of income or study.

4.2 Market-Related Activities

The selective code ‘market-related activities’ is comprised of seven open codes: 1) using low-cost marketing, 2) focusing on downloads and building a reputation, 3) getting featured to get downloads, 4) expanding the market step-by-step, 5) contacting users, 6) using a publisher to launch games, and 7) making a portfolio.

Using Low-Cost Marketing
The open code ‘using low-cost marketing’ describes how the startups used free or low-cost tools and channels to promote their games in the market.

The startups used various ways to gain publicity for the game. They all used social media such as Facebook and Twitter to announce the release of their game, and prepared press releases for reviewer blogs and websites.

“We wrote a really good press release, and then sent it to the biggest Windows Phone blogs, review sites […] All of those! Then we actually sent it to all local, small Windows Phone sites that are in the local language. At that point, we just
used Google Translate to translate [...] We got pretty much at least ten articles on the locals.” (Startup N)

“For example, there is a tool [...] that you can create a campaign and you subscribe your friends or different users to this, and when you want, you select a specified day, and when that day arrives, all these people will share the message automatically [...] on all their Facebook and Twitter accounts [...] We also encourage people to promote the app from the game itself. So, we give them a reward if they promote the app on Facebook or Twitter – something like that.” (Startup M)

In addition to using the Internet to spread the word about new games, some startups mentioned their efforts to use any possible outlet, such as exhibitions and gatherings to show the game and negotiate with possible promoters. They also tried to exploit their networks and contacts, particularly those working with the platform.

“We showed it not in a booth, but just [with our team member] walking around at GDC [Game Developer Conference] in San Francisco, and it sounds like it’s going to get us featured on Mashable and Pocket Gamer [...] Because we don’t have money for advertising or marketing, we have to just go on. The game is pretty and when people see it, they’re like, ‘Yes please!’” (Startup C)

“We’re going to sort of contact the standard bloggers and stuff, and we’re going to hit up the contacts we have here from Nokia. Also, our old previous Nokia and Microsoft contacts.” (Startup A)

However, the startups were also aware that getting featured or receiving a review that helped them to get downloads could happen as a matter of luck and they were not fully in control of this.

“Windows Phone Central reviewed our game on Friday night. So, we’re still on the front page for the whole weekend. So, that was great, and actually we got reviews from all over the world.” (Startup L)

If they had launched several games, the startups tried to use their existing community of users to promote their new games. Consequently, they cross-promoted their new games by advertising them in their previous games.

“Our marketing for [our game] will be with different cross promotions. At the same time we’re releasing [this game], we’ll release two more games, like simple easy games that will be completely free; and [we] will cross-promote [this game]. So, a lot of people will know about [our game] [...] We’ll also update our website, Facebook, Twitter and [...] our [photo editing app]. It’s like a kind of strategy that people will see these pictures of guys with [edited] nose or ears in social media, but they might wonder, ‘Oh, what is this?’, and then they can see they can download the app for Windows Phone.” (Startup M)
Focusing on Downloads and Building a Reputation

The open code ‘focusing on downloads and building a reputation’ describes how the startups prioritized the number of downloads over generating revenue in the early stages. They tried to get a critical mass of users, and to build a reputation to be able to compete with popular game studios, particularly before offering their game for a price (as a premium game). They tried to build awareness and attract people to download their games by offering the game for free and enabling monetizing via in-app purchases and/or in-app advertisements.

“I like the downloads, largely in part, because it builds my brand. And so, it’s kind of like, if you can get the numbers, you can make the money later; or at least, you have the potential to make the money later.” (Startup K)

“It’s [that] the competition in the pay marketplace is just so intense – or at least it seems that way from the outside – because you look at the top list and it’s Angry Birds, FIFA, Mirror’s Edge – major triple A studio brands – and those are massive marketing machines to try and compete with. So, you really need a hit to get visibility in that part of the marketplace [...] I think we have a lot of room to experiment as we go, or at least I would like the room to just try, ‘OK, what if we make it free this week?’” (Startup C)

Getting Featured to Get Downloads

The open code ‘getting featured to get downloads’ describes how the startups considered the market mechanism in order to increase their download volume generated by the increased visibility upon being featured. Getting featured was a critical issue for the startups.

“A few days ago, when we were featured in Brazil, [...] we got at least 600 downloads more than we usually got.” (Startup B)

“When it’s launched, we send out press releases. We announce it on our Facebook, Twitter page, and that’s basically it; except that we also cross-promote within our own apps [...] This has very well worked for us, because after – well, I’d say about two weeks, Microsoft and Nokia pick it up without us contacting them; and they start to feature us in different markets. So, up till now it has quite well worked.” (Startup D)

However, in some cases, the featuring did not target the startup’s desired market or time; hence it did not provide the benefits sought by the startups.

“I got notifications of getting featured in some markets. Not [the] markets I was particularly interested in, like, South Africa, Thailand, [and] a few others that I considered minor. I mostly am interested in English-speaking markets – of course – because I’m not localized.” (Startup K)
Expanding the Market Step-by-Step
The open code ‘expanding the market step-by-step’ describes that startups tended to invest in the markets gradually.

Since the startups were struggling financially, they tried to allocate their finances wisely. Some startups tended to consider famous game markets and localize for them. Others tended to put the game on the market in English in the first place, and, based on the analytics results, saw where their users were coming from and localized accordingly. In line with this approach, they considered their users’ reviews and requests for localization as well.

“So, the languages I’m going to go for is Spanish and French. I get many good reviews from the French; they really like it. And Italian, because there are many phones. And Portuguese because Brazil is very large. And Russia of course, because they want it.” (Startup J)

For most interviewed startups, localization included only the contents of the game and not the name. However, one of the startups (L) highlighted the importance of localizing the game name for some markets, such as China to increase the chance of downloads.

“We always called it [game name] everywhere in the world; in Italy, France, Japan, who cares. And we did it on iOS and Android too, and actually here I discovered that for example for China it makes sense to translate it; because they don’t like to read even the title in English, but it’s fair enough. It’s kind of easy to do.” (Startup L)

Contacting Users
The open code ‘contacting users’ describes how the startups were keeping in touch with their users and community. Their channels for contacting users were usually through social media, such as Facebook and Twitter, and through direct contact and response to emails if users wrote to them.

“Basically, it’s through mail. They usually write. Also, on Facebook, we have [our game] page [and] we have [our company] page; they write there quite often something. Those are really good, people like us ... Twitter as well. I am not that much of a Twitter user, so mostly I focus on Facebook and mails.” (Startup T)

“They send us a lot of emails, and I tend to reply to them personally. We have our Facebook page where they can ask us anything, or talk to each other, and where we show our game updates and stuff like that. We used to have a development blog, but it’s not been active for a while, because the traffic was really low, so the Facebook page pretty much covers it [...] We have a Twitter as well, which is not really actively used. It’s mainly the Facebook so far.” (Startup I)

Using a Publisher to Launch Games
The open code ‘using a publisher to launch games’ reflects that some of the startups were happy to use publishers, particularly in unfamiliar markets, to
avoid dealing with the difficulties. In addition, using a publisher allowed them to capitalize on the publisher’s already existing mass of users.

“We’re speaking currently to a publisher, who is quite cool, [and] who actually knows [our previous game], and really likes it, and so is keen to publish our next stuff based on that. So, that’s quite cool. I think I’m very at peace with the Chinese market, to just tell them – you know, [I mean to the] Chinese publishers – basically, ‘All right, however tacky you want to make this, just let’s do it.’ [Laughter] You know, because that’s just a fact about the market; and things are just not seen the same way. And I figure having someone there who really understands that stuff and wants to drive it – like, I’m OK with it.” (Startup A)

“[My game] has been published by [publisher name]. It is one of the biggest US Mobile publishers with a more than 100 million user-base, so kind of big! Yeah, it was nice.” (Startup L)

Making a Portfolio
The open code ‘making a portfolio’ describes the tendency of some startups to make and strategize for a portfolio of games. For example, they tended to offer a ‘sequence’ to their previous game to build or enhance their reputation. They aimed to make a series of games which were connected to each other to motivate users to follow them. The startups gave examples to connect their games through a storyline ‘cliff hanger,’ or by using the same main character in different games.

“I’m getting the six books that I have to get those out in the queue. I’ll have a big, strong portfolio that can all link to each other. [I will] get on iOS, and then get media attention and just […] try to intercept some of the money.” (Startup K)

“[We develop] video games for smart phones and tablets of our striking character [character name], a little [animal]. [Our new game] is awesome. It’s a re-edition of our [previous game].” (Startup M)

“This is because we wanted to create [a] portfolio; and we thought let’s do some kind of a simple app that could be folkloric for the locals, and this would be a new way to get noticed in the community.” (Startup Q)

Summary of ‘Market-Related Activities’
The selective code ‘market-related activities’ describes the startups’ activities related to the market upon publishing their games and afterwards. It indicates that the startups considered free or low-cost marketing channels to promote their games, such as social media, blogs, their contacts, cross-promotion, and/or visiting exhibitions. Startups aimed to gain a reputation by having a large volume of downloads at the cost of offering the game as freemium and/or ad-based. Some planned to capitalize on the large volume of users by providing sequential games in future. Startups paid critical attention to getting a fea-
tured spot to receive downloads; it was particularly important for them to get featured in their desired market and at their desired times. Their limited financial resources made the startups follow a step-by-step expansion of the market by initially launching globally to discover high-potential markets and localize for them. The startups had an open communication channel with their community of users through social media and email. Some startups were happy to use a publisher in unfamiliar markets, and to get a fast track to a large number of users by exploiting the publisher’s pool of users.

4.3 Experimenting Activities

The selective code ‘experimenting activities’ includes three open codes: 1) experimenting with previous game, 2) experimenting with AppCampus game, and 3) soft-launching.

Experimenting with Previous Game
The open code ‘experimenting with previous game’ describes how the startups tended to experiment when they were uncertain about the technical side and/or the business side of game development in their previous efforts. Their experiments were related to learning about a platform, a particular mobile device, or the development process.

“I just recently did a port of one of them for Android; but, that was – again, just not to have [a] huge success on it, but more like a learning exercise for the next games […] One of the simplest games we have, just did a port [of it for Android] to see how easy it was to make a port.” (Startup D)

“Half a year ago, we started working with Android and the iOS market, and also, we have few applications there, but only for testing. We didn’t promote them – or anything like that – so, we don’t have a lot of downloads there.” (Startup R)

“[With the previous game, we learned] that we are able to finish the game and launch it […] We wanted to answer some questions – that we are able to get the game into the stores from the very first idea until the end – some kind of exercise or experiment.” (Startup S)

Experimenting with AppCampus Game
The open code ‘experimenting with AppCampus game’ describes how the startups tried to use the AppCampus funded game as a learning experience to experiment with different aspects of game development, such as publishing, monetizing, and/or new innovations.

“[For] the AppCampus, what I wanted to test [was] to raise the quality of the app by having no ads in it and see if it would help with the conversion.” (Startup F)

“The goal of this game for us is to see if we can make like those other storytelling games in the PC game market, to see if we can adapt it and figure out what’s not
working and figure out how we can fix it, to see if we can find a storytelling model that we can then launch other games with. We’re trying to find a new way of storytelling, so this is the experimental game.” (Startup C)

“[We] want to learn the [publishing] process on our own and maybe we can deal in some publishing in iOS and Android, but as a last chance of course we will self-publish.” (Startup Q)

**Soft-Launching**

The open code ‘soft-launching’ describes the tendency of the startups to use the WP platform as a testing platform before offering their games to the more competitive platforms, especially given the 90-day exclusivity period. They used the opportunity offered by the funding and launching on the WP platform to polish their games in both technical and business terms before publishing them on other platforms.

“You make this exclusivity for Windows Phone, but you get a lot more back. For example, for 90 days, you get the data, you get the reviews from users, what they like, what they don’t like. It’s like, the soft launch for your product for your app or game, if your app or game is cross-platform, for example.” (Startup R)

“Windows Phone is an interesting market for us, because there’s no competition. Like, we’re going to be the biggest game in the genre there. Pretty much the only game in the genre [Laughs]. So, that’s why it’s interesting, but the download numbers are not super-high. But because of the funding thing, it’s sort of a chance to get two extra months of development time and to polish the game more, and to see if all our changes work and then get ready for our iOS launch, really.” (Startup I)

“On Windows Phone, I give everything away […] To me it was more valuable to be able to tout about, ‘Look at how many downloads I have.’ […] That, to me, is more valuable than a few bucks. And so, I give everything away.” (Startup K)

**Summary of Experimenting Activities**

The selective code ‘experimenting activities’ describes the startup’s learning-by-doing activities to cope with their lack of knowledge. It indicates that the startups used experimentation as a method of learning about technology, platforms, development, and other related issues in their previous activities. The funding received allowed the startups to try new experiments with their game, such as applying a new monetization model, trying an innovative idea, and learning about the app publishing business. The startups used the WP platform as a soft-launching platform where they could get feedback and increase the quality of their games, and gain a high download volume and a good reputation before targeting more competitive platforms.
4.4 Resources

The selective code ‘resources’ contains two open codes: 1) exploiting team member’s capabilities, and 2) accessing low-cost resources.

**Exploiting Team Member’s Capabilities**

The open code ‘exploiting team member’s capabilities’ describes how the startups tried to exploit their team members’ competence in order to save on expenses. In fact, they were using their capabilities inside the firm as much as possible and tried to do as much as they could in-house.

“We are covering [the translation in] Italian, Spanish and English in our company, because we have local native speaking people of these three languages. So, half of the translation is going to be done by them.” (Startup Q)

“We’re lucky that I can do the art and the code, but it would be a hundred grand if we needed a coder full-time for a year.” (Startup C)

**Accessing Low-Cost Resources**

The open code ‘accessing low-cost resources’ reflects that startups tried to acquire low-cost resources in various ways such as outsourcing, exploiting their community of users, colleagues and networks, and family and friends, and taking advantage of local opportunities. Thus, they tried to overcome their lack of knowledge in some certain areas, mainly related to art (such as music, sound and design, localization languages), and overcome the lack of finance.

“Well we do like everything – the most important stuff – but music and sound, we use third party people that do them.” (Startup M)

“Well, I outsourced a lot of the original work. I can code in Java and use Eclipse and everything. So, I would fix [the] bugs […] My brother-in-law was very good at Android development by then, and so, I was like, ‘Here’s $400’ [laughs]; and it was so I got the basic platform and now I just tweak it myself from here on out.” (Startup K)

External low-cost services for hiring freelancers for a specific task such as design, art, and translation services were mostly found through the Internet or networks, or by using interns in their firms.

“Some of them [i.e. the voice actors that we are using] are our friends, because we know other theater people. We put out calls to the universities and their theater programs.” (Startup C)

“So, there are forums where people can post jobs […] For the music, I found it on freelancer.com.” (Startup H)

Further, in order to save on the costs of developing the game, some startups used interns or less-experienced workers, which made it a good deal for them.
to get a low or a zero-cost service, in exchange for providing work experience for the workers.

“We try to reach out to students and non-union actors who would be more flexible and, to put it bluntly, we wouldn't have to pay them as much, which there's a trade-off because with union actors you get professionals who will give you really awesome first takes, whereas non-union actors will cost more of your time.” (Startup C)

“They are free for me. I don’t pay them anything, which makes it great for me. [Laughter] But in the end, I’ll spend a lot of time on coaching them.” (Startup D)

The startups also used their community of users to get help in translation, beta-testing, and acquiring early adopters as well as promoters for the game.

“Our gamer from Spain decided to translate the whole game for us and was very proud of it. There was no money request, just ‘Oh, thank you, let me do that!’ So that was very nice.” (Startup G)

“I may have under 3,000 [Facebook fans] in there; but when I ship, I know [that] I’m going to get at least 100 downloads like that [snaps fingers]; and I know they’re all people that like my stuff; and so my ratings tend to be high.” (Startup K)

Startups also used their networks to get information about opportunities in terms of funding, finding an expert in an area, and feedback about the apps.

“I borrowed some Euro from my dad to put toward the company.” (Startup P)

“A lot of Vancouver game developers know about AppCampus. So, someone who was here for the first or second camp, had been through the program, and once game developers find out that there’s money out there, it spreads through the community pretty quickly.” (Startup C)

“Here, we get a lot of connections with guys. We just ask them, for example, for beta test[ing] of our app. ‘Is it a good translation?’, for example, for Brazil and Portuguese – or something like that. Because when you are a native speaker, you have that opportunity, for example, to play the game. And you didn’t spend a lot of time with that. You can say, “This is wrong. This is wrong, and this is wrong.” (Startup R)

Moreover, some startups mentioned their use of local opportunities that they found in their place of residence.

“There was this accelerator program in [town] at the same time. So, we got an office. We really didn’t get anything done until we shared an office, and we were working every day there together […] We have this entrepreneurship society in our university.” (Startup P)
“I tried to show it to my friends’ kids and people who don’t play games. Like yesterday, in Aalto games, we go to this [type of events] and if there’s an expo and a possibility to show. So, we would just go there, put the phones on the table and then, when somebody’s playing, go and watch them. Don’t say anything and it’s like – as you probably know – the best resource that don’t help and don’t say anything. Then you also hear the honest feedback; because, of course, in the beginning all our friends said, ‘Okay, this is nice, this is great.’” (Startup B)

Summary of Resources
The selective code ‘resources’ describes how the startups used the available resources to perform their game developing and publishing activities. The startups tried to exploit their team member’s capabilities as much as possible in order to save on costs. They acquired low-cost resources for the missing expertise in their firms, which was mostly related to art or translation. Therefore, they used Internet-based services, freelancers, and less-experienced job seekers who could work for free or at a low cost. The startups used any resources at hand to get the job done, for instance through their friends and families, community of users, and contacts. They also exploited the privileges available to them in their local environment. These types of services were used for access to further resources as well in terms of knowledge, finance, and networks.

4.5 Game Design Activities
The selective code ‘game design activities’ consists of seven open codes: 1) differentiating the game, 2) integrating social aspect in the game, 3) developing the game, 4) avoiding complex game design, 5) designing for monetization, 6) caring about getting user star ratings and reviews, 7) paying attention to game quality.

Differentiating the Game
The open code ‘differentiating the game’ explains the ways through which the startups tried to differentiate their games in terms of innovativeness to make them stand out in the market. For the startups, it was important to offer a particular factor in their games that could interest users and make them continue playing the game.

“You have to show something valuable: it can be story, it can be relaxing fun, it can be very deep to a totally relaxing comedy. It doesn’t matter style. It matters quality. And you can get quality in every style. In games you can do for example very fun game that is Angry Birds style. It’s kind of a perfect example. They did [an] awesome thing with all the birds, all the game play and everything, and [the] addiction to the game. It’s fantastic. Then you have a deeper game [game name] that is more like atmosphere feeling, and kind of makes you strong emotions inside. So, that’s the opposite, playing with emotion.” (Startup T)
One way of distinguishing the game was to provide a more comprehensive game-playing experience by combining features that existed in distinct games or following an opposite direction to the mainstream in the market.

“We will think – really, really think – it’s the best game in the world, and that means that in this sense, it takes something from every kind of game. It’s a combination.” (Startup J)

“I was looking at the market at that time and thinking all the games are so sweet and so happy, like happy butterflies. So, I wanted something more Halloweenish, still funny, dark humor – something like that.” (Startup O)

Some startups tried to correct a ‘problem’ that existed in other games, such as offering a more challenging game, or providing the possibility for multiplayer as a feature that they could not find in similar games.

“Most of the games that I played have a lot of menials; like stupid enemies that you just shoot them once and they’re dead. They’re easy to kill, they’re stupid. This game focuses only on challenging enemies. They also have a lot of attacks, projectors that are different from most games.” (Startup H)

“We’ve introduced a synchronous multiplayer. So, I would be able to challenge my friend whenever I want during the day. I just do it and then the game will be able to make the two players interact in order to race each other in [a] synchronous model. We introduced a particular feature, car customization – not selling the pre-customized car, but the user will be able to customize their own car, basically by taking a picture, and the system will transform the picture in the car texture [...] What makes the difference nowadays in the game is the possibility to have your own stuff in the game; and so we’re allowing the user to choose what they really want their model to look like.” (Startup G)

In addition, some startups tried to provide a unique experience for the game player, for instance by implementing a distinguishing art (e.g. graphics, music) in the game. They believed that these aspects could help with the users’ emotional attachment to the game.

“It’s more about being poetic. It’s more about combining the lost voyager stuff with the awesome power of, like the BBC Planet Earth stuff.” (Startup C)

“I think what people like about our games [and] that we really like to do is just very weird and surprising stuff [...] You’re like, ‘What? Seriously? Did that just happen?’” (Startup A)

**Integrating Social Aspects in the Game**
The open code ‘integrating social aspects in the game’ describes how the startups tried to attract users to the game and improve user retention through implementing social aspects in the game. Two layers of social aspects were observed. The first layer was about the connectivity of the game to social me-
dia, which was considered as a factor to increase word-of-mouth and raise awareness about the game. The second layer was to enable the possibility of social interactions among users within the game.

Some startups used some mechanisms with the help of social media to increase the curiosity of the game users’ circles of friends by sharing game play results and/or information about the game entertainment.

“It’s a very important start to build the game and to develop the game with social features. When you start the first line [of] code, you have to put something about Facebook or Twitter to help them later to spread your game and the word and to share it. It’s very important.” (Startup G)

“We offer this campaign to gain more users, [in a way] that the users can like us on Facebook, and then they get the ad-free version. So, we get likes, and we get more users, and then when we launch on Android, we try to make [it] more visible then.” (Startup N)

Another mechanism used in connecting with social media was to encourage users to connect to each other, invite each other to play, or help each other to accomplish the game.

“We’re working on a friend system. So, we’ll have in-game chat, so, you can talk to people; you can actually meet people and add them as friends. So, you can see when they were last online. You can see if they’re playing a game, and you can check the game info and see if you want to join that game as well, […] and [the] Facebook integration, so, you can check, like, ‘Who of my friends on Facebook are playing this game?’ as well. ‘Do I want to invite them or not?’” (Startup I)

“You can reward users for liking your page on Facebook or Twitter. This all works very well, because you are directly in touch with them. There are lots of things you can [do]. It’s very important to put people in some need of cooperation with other people. Like Candy Crush Saga does […] So, make people share about your game, that’s awesome if it’s happening.” (Startup T)

However, as highlighted by one of the startups (E), when connecting the game to social media, it is important to consider the match between the target audience and the age restrictions required for using social media.

“We’ve made sure that the game is 3+. So, we haven’t an integrated Facebook or anything into it, because if you do that, your game is automatically then teen. We don’t want to do that; so, it’s 3+, casual game players.” (Startup E)

The second layer of the social aspect was to enable social relations among users. This layer was often applied by the startups, through designing competition-related features or chatting environments for players in the game. They believed that by enabling this type of social environment, the players were encouraged to play for longer and to increase their mastery of the game to compete better with their friends. In their opinion, the increased competition
could encourage users to purchase more customization features to reach their objectives.

“You know a lot of in-app purchasing stuff is driven by the need to show off, the need to be better than people, duh-duh-duh? What we want to do is, we want to have this kind of cooperative thing that everyone can chip into, that benefits everybody; but whoever does it the most, gets, like, their information stuck in the game. So, it would be, like, ‘This building brought to you by ...’” (Startup A)

**Developing the Game**
The open code ‘developing the game’ reflects how the startups chose the development environment. All the startups were using the Unity game engine, due to its ease of use and cross-platform compatibility.

“The big plus of Unity is the fact that things are generally relatively easily portable. And the other big plus is – especially for a smaller team – it makes it so much easier to allow the artist to do their thing. Whereas before, with any other way, unless you’re a big studio – and I think even then – the developers are much more like the gatekeeper.” (startup A)

“It’s a Unity game. So, even before we started making it – after the experience we got from making the summer project – we thought it being very important to find a platform or utilize a game engine that would allow multiple platforms.” (Startup P)

**Avoiding Complex Game Design**
The open code ‘avoiding complex game design’ describes the ways in which the startups tried to simplify the game to fulfill the expectations of users in the mobile context.

According to the startups, a differentiating factor between games on a mobile phone versus other types of devices such as consoles was the matter of portability. They believed that mobile games should have short sessions, due to the fact that many users tend to play these games on the go or when they have a little amount of spare time.

“The idea is you can keep accumulating, but any given bit of gameplay’s really short. So, if you’ve got a little bit of time to kill, it’s in this nice unit where you’re like, ‘OK, well, I can go get some stuff.’” (Startup A)

“So, we were looking at some of those [video games] and they were doing these three-hour games or games with one to two-hour episodes and were working really well. We basically said, ‘OK, that’s really cool, these are the kind of video games we want to make. Can we move these into a mobile context? [...] What if we condense them into 15 to 25 minutes, so they’re more like comic books?’ The problem with a two-hour episode of something is if I’m playing on the bus and I put it away and I go to work for eight hours, I pull it out again, [I think] ‘Who are these people? What was I doing? What’s going on?’ So that’s the idea, trying to adapt it into short play sessions.” (Startup C)
Another important factor for the startups was the complexity of the game. They believed that the game had to engage the user quickly and consider the user’s learning curve in mastering control of the game. In their opinion, the vast availability of free-to-play games and the high competition in the market required the quick engagement of users with the game before they decided to leave for another game. The startups tried to consider this by having a simple tutorial at the beginning of the game to guide the users through visual and graphical aspects rather than making them read long texts.

“So, if you try to put lots of buttons and some combination, people just see a mess, and they leave the game. Nobody wants to learn the game that way. They want to ENJOY the game. And if they have to spend time learning controls, etc. they will leave it and try another game. They don’t have time for this. There are hundreds of games coming every day.” (Startup T)

“But we’re definitely going to introduce everything one step at a time before launch on the Windows Phone version […] So first you get your control – like, ‘Oh, yes, this is how I control my [vehicle].’ Then, an enemy appears: you have to aim, you have to shoot at him. And introduce one element at a time all the game elements.” (Startup I)

An additional benefit of minimizing the amount of text in the game was saving the workload of the developers in localizing the game for different markets.

“We structured the game in order for it to be not too ‘texty’, not to have too much stuff to read at once. It’s pretty simple in this aspect. If you see the game, it’s just play and multiplayer. We changed the back button to a simple arrow and then the option is like the option everyone uses […] The idea of the game was for it not to be too complex for the user to use, and also for a non-native speaker – like if someone is using the English version, they should [be] able to understand the game without really feeling like, ‘Oh no, this is too difficult.’ So, the game design was thought to be like that – to be accessible.” (Startup G)

Designing for Monetization
The open code ‘designing for monetization’ describes how the startups tried to follow the freemium model by offering the game for free to gain a large volume of downloads yet incentivizing users to buy additional features in the game.

Due to the trend in the market for free-to-play games, the startups mostly offered their games for free while wanting to motivate users to buy in-app items. The monetization mechanism was implemented in various forms. These included paying to make progress easier by boosting the power of the game character, avoiding a return to the first level upon failing in the game mission, and adding purchasable cosmetic features. A simpler model was for users to pay to get rid of in-game advertisements.

“You have the standard [game character], the best [game character], and you can also buy some special [game characters] with special powers that make it
easier to complete the level – to beat the bosses. [...] The rankings of the game are based on how long you take to finish the level, how many seconds [...] You take one hit, normally you die, but the serial players can pay to continue." (Startup H)

“If you don’t have enough coins because you haven’t collected them, you can buy a level up, that’s it, very simple.” (Startup L)

“There are three systems. There is a free-to-play model on in-app purchases. You can purchase cars, upgrades, power-ups, and other epic vehicles. Depending on the country [and] on the market, we want to put specific cars to be more engaging to the users.” (Startup Q)

Caring About Getting User Ratings and Reviews

The open code ‘caring about getting user ratings and reviews’ explains the importance of receiving user rating and reviews for the startups and how they tried to get ratings.

For the startups, user star ratings and reviews were important because in their opinion they could provide both credibility for their games in the eyes of potential users, and help the game to get a better position in the store through the ranking systems. Furthermore, startups could also understand the users’ opinions and fix existing problems with the game through the users’ reviews.

“It’s very important. It’s a psychological effect. When you see the game [has] many stars from lots of people, you feel like, ‘There can’t be everybody wrong, right?’ Also, if you see really bad reviews, [you feel like], ‘Why should I play a game that has bad reviews?’ [...] You just encourage the player to go and rate. Because even any rating is actually better than non-rating, because it’s growing in importance on the store. When somebody has lots of rating it’s more valuable for search engines.” (Startup T)

“[The star rating is] very important, because it’s a great start to have more downloads in a few days or a few weeks. Also, for the previous game, we paid a lot of attention to that and were always there to answer customers. Maybe there was some bug or something already answered. We wanted to change the star also, the star rating, and that sometimes happened.” (Startup G)

The startups implemented a tool inside the game (Rate My App) that would pop up and request the user to rate the game. They mentioned the importance of the timing of the pop-up in order to get ratings from satisfied users, rather than random users. In addition, some startups highlighted the importance of localization in receiving more ratings by users due to the ease of writing reviews in their own language.

“There is this Rate My App, and it shows after the fifth session [...] because if the player doesn’t like the game, he won’t reach five sessions [chuckles]. So, we don’t want bad reviews. So, we take reviews from people that actually play the game" (Startup H)
“We have got a Rate My App thing in there at level seven. It only comes up once though, because I don’t want it to distract [the users] from the game play.” (Startup E)

*Paying Attention to the Game Quality*

The open code ‘paying attention to the game quality’ reflects the high attention paid by the startups to the quality of the game in terms of its design and performance on target devices.

The startups believed that while the attention to quality could be hindered by the need for a fast launch of the game and lack of financial resources, participation in AppCampus had a positive effect on their games’ quality through: 1) training received in the one-on-one sessions, 2) the quality assurance review, and 3) funding to make the game more advanced.

“Of course, the [AppCampus fund] money’s nice, and that they drive this and that. They drive you to have a good product in a time limit, which is very good. That’s, if I have to put two things, this is the main.” (Startup J)

*Summary of Game Design Activities*

The selective code ‘game design activities’ describes the issues related to the game artifact that the startups considered in their game design and development. The startups tried to differentiate their games from those already available by implementing some kind of innovation in them, mainly driven by the founders’ own experiences and desires. They implemented social aspects in the games, either in the game mechanism by allowing users to compete and communicate with each other, or in connecting the game to social media to allow users to spread word-of-mouth about the game. Their objectives were to increase user retention through social interaction, to provide incentives to buy features to be more competitive in the game, and to bring new users to the game.

The startups used a cross-platform development environment to have an easy port to future platforms. They also paid attention to factors related to the mobile game context due to the high competition and ease of un-installment of the game by users, such as fast engagement of the users in the game, avoiding complexity and confusion for users, and short sessions of the game to allow playing while mobile. They also tended to use visualization rather than text in their game, which they found to be more attractive for the users as well as making the localization easier. The startups considered implementing monetizing mechanisms inside their game concepts. They offered the game for free to follow the market competition rules while incentivizing players to pay for extra features through the mechanism of the game and using social aspects.

The startups found user ratings and reviews to be important factors to raise their reputation and bring new users to the game. Consequently, they implemented tools asking for star ratings from users when they thought a user was already engaged enough with the game to provide a high rating. So, in getting the rating, timing was important for the developers. In addition, they consid-
ered the positive effect of localization on receiving more and better ratings from users.

The startups paid much attention to the technical quality and performance of their games on target devices, and they thought that participating in AppCampus increased their game quality through the training received, quality assurance, and funding which allowed them to invest more in their game.

### 4.6 Accelerator’s Facilities

The selective code ‘accelerator’s facilities’ includes six open codes: 1) getting invited by the platform owner, 2) benefiting from the funding, 3) learning through AppCampus training, 4) benefiting from networking in AppCademy, 5) building a relationship with the platform owner, and 6) building a reputation through AppCampus.

#### Getting Invited by the Platform Owner

The open code ‘getting invited by the platform owner’ explains the discovery of the startups by the platform owner. They considered the selection to be based on their high potential for success by showing historical download statistics, winning of competitions, or their pitches at MAAC events.

“I got the Microsoft person contact me because of my other game that I did. Yes, it was downloaded, people liked it, it had a nice number of downloads, but I didn’t monetize it very well. Now it’s, this game, I make [it] right now, I improve the monetization.” (Startup H)

“I actually met one of the people responsible for the program at Microsoft at the conference but at the time [my game] was in the top ten free games on Android and it was like, ‘yes, look at this, you should apply to AppCampus.’ I was like, ‘Yes, I’ll think about it,’ but I knew a bit already about AppCampus and I eventually applied.” (Startup F)

#### Benefiting from the Funding

The open code ‘benefiting from the funding’ describes how the funding offered by AppCampus was an important motivator for the startups to develop their games for the WP platform and to accept the 90-day exclusivity period.

Their main motivation to apply to AppCampus was the funding. Each startup was offered 20,000 euro. The funding and time schedule allowed the startups to make their games more advanced in terms of design and implementing new ideas such as adopting a new revenue model, and/or introducing a new gaming concept to the market. They considered that the funding enabled them to invest more time in their game development activities and be more disciplined in meeting deadlines.

“We can develop [the game] all our time from now, because we didn’t bother about go to other work a lot, for example, to earn money from different places. I think it speeds up the process.” (Startup R)
“So, in the matter of time limits it’s very good [...] QA [quality assurance] and so. They make you have a good product which otherwise, you know you might not have. So, in that sense it’s very good that I got it.” (Startup J)

“Probably if we hadn’t had AppCampus it would have gone in a lighter version and been more like a Flappy Bird style, but we would certainly have gone through with it, probably in a lighter version, skipping the in-app purchases and stuff in the first place, and making it more like an endless thing instead of level-based.” (Startup D)

The startups turned the 90 days of the exclusivity period into an opportunity to cover their costs of development, publish their game on the WP platform, and see how their games could perform without experiencing heavy losses in case of failure. They intended to polish their games before offering them to other more crowded and more competitive platforms.

“It’s the funding thing that makes it worthwhile. [...] It’s sort of a chance to get two extra months of development time and to polish the game more, and to see if all our changes work and then get ready for our iOS launch.” (Startup I)

Learning Through AppCampus Training

The open code ‘learning through AppCampus training’ describes the startups’ reflections on the training in AppCampus. In their opinion, the training improved their knowledge about the business side of game development. Some of the startups had developed a game with market appeal previously, but had neglected to promote and/or monetize their games. Through the training received during AppCademy, the startups learned how to manage their market strategies with limited resources, how to prioritize their goals, and how to improve their game design for better user retention and monetization.

“The one-on-one coaching, with the coaches; they gave us amazing ideas to improve the game with monetization. Like, we feel like we MUST improve, we MUST improve.” (Startup H)

“In the past I was just focused on creating a fun game with no monetization in my mind. So, in reality all of our older games don’t have good monetization systems at all, but now with these sessions – and of course we went to the Game Jam, in the middle of this AppCampus Alumni – and there it was awesome [with] people talking about monetization, and I really learned a lot.” (Startup M)

“Instead of going territory by territory, we got the advice to go global from the first day, because for small companies, it’s much easier to reach the good performing markets if you go global from the first time; because then you can get numbers easier of first time of which countries are performing better [and] then make action to optimize those countries.” (Startup Q)
Although the startups had technical knowledge, and did not mention any big changes to their game design as a result of participating in AppCampus, they still believed that the quality assurance requirements were a useful step to creating a better design for the game.

“QA was extremely useful in AppCampus, I mean like they pointed out things that I have not noticed or thought it’s not important, but actually it was important. When you see the end result, it’s much better than before.” (Startup O)

Moreover, some other benefit was delivered to the startups by being able to judge their performance compared with other games. This took place through the discussions, feedback, and AppCampus market reports distributed to them.

“I know that the first few days they – other people [i.e. peer-developers] – had more downloads than us in the same time or less time. There [were] the ones that had already published a game. So, I want to be better. I felt like I wasn’t so good. So, I have to improve.” (Startup H)

“If that demo that we make isn’t good enough for AppCampus, then it wouldn’t be good enough for mobile markets. So, we could just give up.” (Startup P)

**Benefiting from Networking in AppCademy**

The open code ‘benefiting from networking in AppCademy’ describes the reflections of the startups on the networking opportunities with peer-startups, the platform owner, and external contacts.

The first benefit was made by interacting with peer-startups. Having various international startups in a shared space for two weeks accompanied by social events provided a possibility for informal networking among the startups. Startups mentioned that the informal communications created a feeling of community, encouraging sharing experiences, and requesting help from one another in terms of feedback, beta-testing, user rating, and localization languages. The peer networking of the startups was not limited to the physical space, but through an alumni group on social media as well.

“This is the best thing of having 27 persons around the world here in this space. Then this is the good part because you by yourself ask everyone about your game and ask what they like and don’t like, and then you take notes and improve it [...] This is something that you have to spend a lot of money in events or traveling around the world if you want to get such a big feedback as an international point of view.” (Startup Q)

“We got to meet people from around the world and people like a studio from Finland, and it’s really cool, because you find out about what’s going on with games all over the world just in my Facebook news feed because now we’re all Facebook friends. I definitely value that aspect.” (Startup C)
In addition to the opportunity to network with peers, the accelerator provided the startups with the possibility of external networking with other market actors, such as experts, servicing firms, and publishers for various geographical areas.

“They helped us to meet [the] right people, so they give us [the] right feedback. So, on that feedback we could create our own ideas better. So yes, they had influence in a mentoring way [...] It was very nice to see their opinion of what they’re doing from a business side, marketing side, etc. We’ve been very experienced in developing but kind of rookies in marketing and running businesses. So, huge importance for us to meet these people and get on kind of [the] right track." (Startup T)

“Cross-promotion from [company name], I think. We will use it. AppCampus gives us a voucher for that.” (Startup R)

**Building a Relationship with the Platform Owner**
The open code ‘building a relationship with the platform owner’ describes how the startups tried to negotiate with the platform owner in order to receive a featured spot on the store according to their plans. A crucial issue for the startups, mentioned by all of them, was that during AppCademy they learned how the platform worked. Through knowing the platform decision-makers, the startups tried to have an impact on getting featured spots for their games in their desired markets. They also tried to negotiate with the platform to receive timely support for their marketing activities. They believed that getting featured was the best way to make their games visible to users.

“Being here, you can know from inside how they work, how they decide which app goes to the front of the store. Being here is a huge privilege, because I can get the contact person from each country [...] If I say, ‘I want to promote my game in this country,’ and I’m going to have my own campaign, also I can talk to the press of the country.” (Startup H)

“Sometimes it gives you also feedback about the problems you really haven’t thought about, like, ‘OK, I’m going to do it this way,’ and then they say, ‘But you might do it this way. It’s better to move around and contact, like this other possibility. You can contact these bloggers and you know, we have contacts’ [...] It’s a lot of news for us and we’re happy, it’s fantastic.” (Startup G)

“Even if I had money I couldn’t do it, because when he puts me on top of the shop, it’s worth much more than money. I couldn’t even say how much it was. I can’t get those downloads elsewhere.” (Startup J)

The possibility of getting to know the platform’s contact people was highly appreciated by the startups and was seen by them as a positive distinguishing factor for the WP platform against other platforms.
“And one of the main things I appreciate about it is the networking. Like, getting to know people who are responsible for that kind of stuff. Because on Google Play, you don’t know anyone there. I don’t have a contact in Google. I have no idea who’s working there. If I want to get my game promoted, there’s no chance, like 0.1 percent gets that, and it’s not as transparent as it is here. So that’s really nice, that we get to see faces to the people we’ll be working with, and, yes, if you have a question you know who you’ve got to ask about it. It’s really developer-friendly.” (Startup I)

“It’s not the same for them to promote a game of a guy they don’t know, than to promote my game when I am here and meeting the staff at AppCampus. It’s much different, so I think the real value more than the money will be only for the people who come here; otherwise, it’s just the money and a little bit of marketing.” (Startup M)

**Building a Reputation Through AppCampus**

The open code ‘building a reputation through AppCampus’ describes the startups’ reflections on the benefits of AppCampus in terms of building a reputation in the market.

The startups saw AppCampus’s support as a means of giving them credibility in the market. They also believed that having a professionally designed game and a large download volume on the WP platform would be beneficial to get more promotion and/or a higher chance of future fundraising.

“Like I’m from Brazil, so if I write a story about how has this AppCademy been to me, [...] they [the press] will probably make a notice [and] make news. So, we’d get more monetization to the game.” (Startup H)

“I had already talked to them [a regional funding office], but they didn’t want to give us this public funding thing, because we didn’t have any of our own money. We didn’t take enough risks, in their mind […] But after we had AppCampus money, [...] we could say we’re putting €20,000 of our own money into this, so we’re taking the risk, and then we got this.” (Startup P)

“We’ll also have the game done and we can show it working, and we can show download numbers from Windows Phone. So, it actually makes us more empowered. It’s actually forced us to go through proper development and now we’re better empowered to go and maybe find some real funding for the company.” (Startup C)

**Summary of Accelerator’s Facilities**

The selective code ‘accelerator’s facilities’ describes the way through which the startups used the facilities that were offered by the accelerator.

The startups had a good background in creating games with mass appeal as hobby projects, and they found this to be the reason for their invitation to the accelerator program. They found the funding an incentive for developing for the WP platform. They mentioned the benefits of the funding in making higher quality games, becoming more dedicated to game development, and trying out
new and innovative ideas. In fact, the funding gave them more financial resources to get started and try out commercial game development. They used the funding and the exclusivity period as an opportunity to get feedback from the market and polish their games before going to more competitive platforms.

The startups found the training to be useful in understanding market-related issues such as having a better game design in terms of user retention and monetization, as well as better marketing and promotion for their games.

The startups used the networking opportunity in the program on various levels. They found the peer-relationship to be an important factor in receiving future support in their activities. They also found the opportunity for external networking with publishers, promoters, experts, and other firms very useful in promoting and improving their games. An important issue for the startups was the possibility to connect with the platform owner's contacts to negotiate getting featured for their desired markets and times. They also found their acceptance to the accelerator program as a credibility measure through which they could negotiate with other market actors to get promotion or receive future funding.

4.7 Discussion

In the final phase of coding in the first category, I drew the relationships between the six selective codes of startup’s capabilities, market-related activities, experimenting activities, resources, and game design activities and accelerator’s facilities (Figure 5). This is known as ‘theoretical coding’ (e.g. Glaser 1978; 1998). Next, I will present the theoretical integration of the first category in the light of the extant literature.

Relating the Six Selective Codes

Figure 5 illustrates how the selective codes in the first core category relate to each other.

![Figure 5. Utilizing the accelerator’s facilities.](image-url)
Utilizing the Accelerator’s Facilities

The accelerator’s facilities increase the resource base of the startups by providing seed funding, vouchers, and contacts. The increased resources affect the game design activities, experimenting activities, market-related activities, and the startup’s capabilities. With more resources, the startups can invest more in their game design activities, and also improve their game quality through sharing experiences with peer-startups, whom they meet in the program. Furthermore, the increased resources allow the startups to experiment with various scenarios, such as implementing new innovations in terms of the market (e.g. a new revenue model) or the game concept and technology (e.g. a new game mechanism or concept). Experimenting is a way for the startups to compensate for their lack of knowledge and uncertainties about the market or technology. Moreover, the increased level of resources can assist the startups with their market-related activities. Therefore, they can invest more time in their market-related activities to promote their games, but also, they can exploit the new contacts with application reviewers and publishers made during the program to get more assistance in promoting their games. Moreover, the startups can use their peers as their early game adopters to boost collecting early user reviews and ratings, which eventually translate into more downloads. The increased resources can affect the startup’s capabilities by allowing a more dedicated approach to their game project, rather than having to engage in outside contractual works to cover their costs. In order to receive the AppCampus funding, the startups need to meet deadlines, hence achieving a more organized game development project.

The accelerator’s facilities enhance the startup’s capabilities. Through training and mentoring, startups can learn how to monitor their market-related activities. They can also set strategies for the future expansion of the business according to their relationship with the platform, for instance, in their choice of future multi-platform platforms. They learn the importance of using various analytics tools in their games to monitor the performance and expand their markets. Enhanced startup capabilities can increase the quality of the game in terms of better design, and monetization and market-related mechanisms (e.g. user rate acquisition). A well-designed game is likely to be downloaded more, have a higher user retention level and be able to generate more revenue.

In addition, through the accelerator’s facilities, the startups can shape their market-related activities with the opportunity to negotiate with platform contacts to receive promotion and features for their games in their desired markets at their desired times. Having been selected by the platform owner for funding, they can gain credibility and access to channels for contacting other market players (e.g. reviewers, publishers), which not only shape their market-related activities, but also help them to acquire more resources from other ecosystem members (e.g. local fund providers). During the exclusivity period on the platform and the mentorship and support, the startups can focus on getting a high volume of downloads and improve their games to build a brand for their future activities.

The accelerator’s facilities can also directly affect the market-related activities of the startups through enhancing the visibility of their games by featuring
them on the store, and by promoting them in exhibitions in different markets. The increased visibility can lead to more downloads for the game.

Furthermore, the accelerator’s facilities can also directly affect the design of the game and its compatibility with the platform by requiring the games to get AppCampus quality assurance approval before publishing. Figure 6, which is a more detailed version of Figure 5, shows the information presented above as the properties for each relationship.

![Figure 6](image)

In the next section, following the GTM, I conduct a theoretical integration, which means that I compare the concepts generated from the empirical data with the extant literature (e.g. Glaser, 1992; 1998; Hekkala and Urquhart, 2013; Urquhart et al., 2010). For this purpose, I use the extended literature review which is at the same substantive level (e.g. Glaser, 1992; 1998) on boundary resources and show how the emergent theory confirms, contradicts, or extends the literature (Urquhart et al., 2010).

**Theoretical Integration of Utilizing the Accelerator’s Facilities**

The extant literature shows that accelerators may help startups by increasing their resources in terms of finance, knowledge, networks, and credibility (Cohen and Hochberg, 2014; Hallen et al., 2014; Kim and Wagman, 2014; Miller and Bound, 2011). However, the existing literature has not addressed how the increase in resources and credibility actually helps the startups and affects their activities (e.g. Bergek and Norrman, 2008). The model (Figure 5) confirms the existing literature and extends it by conceptualizing how a corporate accelerator controls or affects the game development and publishing activities. In addition, it shows the direct impact of the accelerator on the quality of the complementary offerings to its ecosystem by its quality assurance tools, as already noticed by Haines (2014).

The results imply that the utilization of the accelerator by the startups is not limited to receiving future capital, as is defined as a measure of success in most
existing studies on accelerators (cf. Barrehag et al., 2012; Hallen et al., 2014). In fact, the findings show that for early-stage startups, high user acquisition is a success measure, and their activities are directed toward increasing the download volume as the main objective in the early stages. This is in line with the existing evidence that mobile application developers may postpone immediate revenue-making for future returns by focusing on downloads (Bergvall-Kåreborn and Howcroft, 2013b).

I argue that in the case of a platform-owned accelerator in this study, the accelerator was performing beyond the role described in the literature for corporate accelerators as building an ecosystem and merely connecting end-customers with startups (Pauwels et al., 2016). Instead, it developed into an artifact of a social nature that was connecting and coordinating the relationships among not only the startups and the platform owner, but also among the startups and various types of ecosystem members who provide supporting services to developer startups. In fact, considering the boundary resource model (Ghazawneh and Henfridsson, 2013), I argue that the accelerator evolved into a social boundary resource for both resourcing and securing the platform by controlling and influencing the platform’s ecosystem (Ghazawneh and Henfridsson, 2010).

In the existing literature, a boundary resource is described with the aim to transfer knowledge between the platform owner and developers, or between developers (Bianco et al., 2014; Ghazawneh and Henfridsson, 2010; Rudmark and Ghazawneh, 2011). Nevertheless, the findings show that a boundary resource in the form of an accelerator has, in fact, developed into a social platform per se, not only to guide and empower the developers in the direction of supporting the platform objectives as discussed in the extant literature (Rudmark and Ghazawneh, 2011), but also to connect and coordinate the ecosystem members with each other, including publishers, peer-developers, and promotional firms. While most of the research on boundary resources have focused on technical boundary resources, this study presents in-depth empirical evidence to highlight the strong social role of the accelerator as a boundary resource. Furthermore, the findings confirm the shaping of the boundary resource through the influence of the internal resources of the platform owner (Mohagheghzadeh and Svahn, 2016), but extend the literature by showing that the boundary resource in the form of the accelerator contained and was influenced by resources external to the platform owner. These external resources were offered by other ecosystem members as well (e.g. making publishers and promotional firms’ resources available to the startups) in addition to the platform owner’s resources.

The accelerator’s facilities as a boundary resource both enhance the generation of complementary apps (resourcing) and impose control on the quality of the apps and the platform’s competitiveness (securing). The funding affects the game quality (Haines, 2014) and allows for new innovations in terms of the technology and market. By empowering the startups both financially and knowledgeably, and by facilitating their interactions with the platform and other ecosystem members, the accelerator helps the startups to focus more on
their core activities of developing and publishing their games and provides them with tools for an increased chance of market success. All this may eventually enhance the competitiveness of the platform (Boudreau, 2012). At the same time, through mechanisms such as the quality assurance review, mentoring, and the exclusivity period, the accelerator makes sure that the startups are following the platform owner’s desired quality and disciplines in the competition, hence securing the platform (Rudmark and Ghazawneh, 2011). The findings indicate that the startups try to take advantage of this boundary resource in order to be able to better communicate with the platform owner. They use the opportunity to negotiate getting their games featured on the store in their markets of interest. They tend to influence the platform owner’s market-making mechanisms (see Bresnahan et al., 2014; Qiu et al., 2017), make it more favorable to themselves, and create a better balance of power between the platform owner and themselves. This confirms the importance of communication between third-party developers and the platform owner as already suggested in Rafiq et al. (2013).

To sum up, the theoretical model in the first category discusses the answer to the first research question, ‘How do early-stage mobile game startups utilize the accelerator’s facilities in developing and publishing their games?’ It shows that the startups utilize the accelerator to improve their capabilities and knowledge, which can affect their game design and access to resources, and improve their market-related activities. They also use the opportunity to influence the platform and connect with other members of the ecosystem to gain visibility for their games. The accelerator is developed into a social boundary resource, through which startups can collaborate more effectively with the platform owner and use the resources that are provided by other members of the ecosystem with the aim to acquire more users for their app.
5. Excelling in the Market

In this section, I present the empirical findings of the second emergent core category, ‘Excelling in the Market’. It contains five selective codes: 1) Monitoring activities, 2) Post-launch activities, 3) Insights on the game performance, 4) Success with the game, and 5) Restraints. Table 8 illustrates the constructs of the core category consisting its selective and open codes.

Table 8. Construction of ‘Excelling in the market’ through open and selective codes.

<table>
<thead>
<tr>
<th>Open codes</th>
<th>Selective codes</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Focusing on analytics; 2) Monitoring users’ behaviors in the game; 3) Monitoring users’ reviews</td>
<td>Monitoring activities</td>
<td>Activities for watching the game performance in the market and users’ performance inside the game</td>
</tr>
<tr>
<td>1) Improving the game after launch; 2) Multi-platform activities</td>
<td>Post-launch activities</td>
<td>Improving activities after publishing the game</td>
</tr>
<tr>
<td>1) Getting insight into users’ behavior in the game; 2) Getting surprised by the download volume; 3) Regret about the past activities</td>
<td>Insights on the game performance</td>
<td>Understandings related to the performance of the game in the market and users’ behavior in the game after publishing the game</td>
</tr>
<tr>
<td>1) Defining success by the download volume; 2) Defining success by the revenue volume; 3) Defining success by personal achievements</td>
<td>Success with the game</td>
<td>What the startups consider as success definition for their game</td>
</tr>
<tr>
<td>1) Restraints by AppCampus; 2) Restraints by the platform; 3) Restraints by supporting tools</td>
<td>Restraints</td>
<td>The limitations that were imposed on the startups</td>
</tr>
</tbody>
</table>

In the following sections, in a similar manner to the first category, I explain the selective codes and their constructing open codes. Subsequently, I demonstrate the relationships between the selective codes and finally ground the findings by the use of the extended literature review in the area of the iterative approach in software developer startups which I presented in Section 3.2.2.

5.1 Monitoring Activities

The selective code ‘monitoring activities’ is comprised of three open codes: 1) focusing on analytics, 2) monitoring users’ behaviors in the game, and 3) monitoring users’ reviews.

Focusing on Analytics
The open code ‘focusing on analytics’ describes that the startups tended to use tools which allowed them to collect data on the performance of the game in the
market and users’ demographic data. For this purpose, they mostly used available analytics tools such as Flurry and/or Google Analytics. They collected various types of market-related data such as number of downloads, number of uninstalls, user retention, number of active users, and users’ demographic information (e.g. age, location, gender).

“We’ve got about 96 percent males. So, it’s good to know what kind of audience you’re targeting. So, yes, the geography, ‘Where do people come from?’ We’re going to add – we have to add – what’s the lifetime value for a user?’ because we’re not checking that yet.” (Startup I)

“We’re going to start analyzing some basic stuff like how much they [the users] play? Are they still playing? How much time [do they keep the game] when the people install the game? Maybe he did that [i.e. uninstall] the third day, maybe he doesn’t delete it. So, we’re going to see the life value of the user, like how much time he plays. How much does he spend in terms of monetization?” (Startup M)

Monitoring Users’ Behaviors in the Game

The open code ‘monitoring users’ behaviors in the game’ describes that the startups monitored the behaviors and activities of the users while they were active in the game. By observing users’ activities in the game, they could gain an insight into their points of interest, challenges, drop-outs, or usage of in-app purchase features. Startups could then use this information for their future updates and strategizing for their games.

“I know anything that you touch. Whatever you touch, I know. So, I know how you think. [...] We have all this data, and then we calculate it [...] I saw that I have a drop off [...] From the first one to the second one we had a drop of 200. I said, ‘OK, that’s nothing. It’s probably [because] people didn’t want to play it again or didn’t understand [it].’ But from the second one to the third one we had a drop of from 3300 to 2000. That’s when I understood that we have a problem.” (Startup J)

“We just started using funneling, to see at what page people drop off [...] We’ll implement heat maps for the levels. So, we can see ‘Where are people moving? What’s the most logical flow? Where do people get killed most?’ So, we can sort of navigate everyone to the same hot spot, so [that] there’s more action going on. Yes, a lot of small things, but I guess those are the main ones.” (Startup I)

Monitoring Users’ Reviews

The open code ‘monitoring users’ reviews’ reflects how the startups paid attention to the text reviews generated by the game users and/or their star ratings. Through user reviews, the startups understood what their users cared about, and what issues the users had, and got new ideas on how to extend and improve the game.
“So, the languages I’m going to go [for localizing the game] is Spanish and French. I get many good reviews from the French, they really like it. [...] And Russia of course, because they want it.” (Startup J)

“Sometimes our gamers just send us their stories, like, ‘In order to win this season, I did this; I met my friends and I used to discuss with them how I can tune the car and get the best results.’ So, our community really helps us a lot with that. [...] They write: ‘You know, I was doing this [and this thing happened],’ and it’s very important. [For example,] ‘I don’t like this. Why don’t you [change it to this]?’ So, communities are a very good resource.” (Startup G)

Summary of Monitoring Activities
The selective code ‘monitoring activities’ describes activities related to watching the game performance in the market and users’ performance in the game. The startups mostly used available and, in some cases, self-developed analytics tools. They monitored users’ behaviors in the game in order to set plans for future updates. The startups used user-generated reviews and star ratings to understand the actual users’ expectations and opinions.

5.2 Post-Launch Activities
The selective code ‘post-launch activities’ contains two open codes: 1) improving the game after launch, and 2) multi-platform activities.

Improving the Game after Launch
The open code ‘improving the game after launch’ reflects that the startups still improved their games after their release, based on the information collected by using analytics tools or users’ reviews. The improvement could take place for various reasons, such as increasing user retention, creating more content for the game, tailoring the game to the target markets, improving/implementing monetization schemes, and fixing bugs.

For some startups, their games faced difficulty in engaging users fast enough and/or for long enough, and users tended to opt out for various reasons such as difficulty in mastering the game. To resolve these problems, the startups tried to involve the users in the game more smoothly, for instance by changing the game tutorial to a simpler and more engaging one, as well as improving the challenge level of the game according to the actual users’ abilities.

“We’re still fixing it and changing it a bit according to the market [...] All you have to do is just read the reviews. That’s the easiest thing to do [...] What I heard from them is that I should give the people a very slow walk in the game. So, after 20 games they play, they say, ‘I’m smart.’ Now they can cope with the next one.” (Startup J)

“We have [a] deep analytics system. At first, we wanted to see how players play the games, where they die mostly, so we can change the levels a bit, so [that] more players enjoy it more; they can more easily complete it, but not too easily [...] Also, we have this backtracking software that, if the game crashes for player,
we get email notification and information about the crash, so we can fix it immediately.” (Startup N)

The startups tended to enhance the content of their games gradually after launch by adding more features and substance to the games in terms of customizations, obstacles, levels, etc.

“[We are] adjusting the game play, because now we found out that the beginning of the game is too hard, and then the ending is too easy. So, like changing the curve of challenge […] We are [also] adding some enemies there and a few new mechanics.” (Startup B)

“People are like loving the game, and they ask for us to make more levels and updates; and we have been doing that since we launched. We have just doing that because the users enjoy.” (Startup N)

An interesting point mentioned by one of the startups (T) was the importance of the meaningfulness of the updates made to the game. In their opinion, meaningless updates could result in low-star ratings from angry users.

“[The update] has to have a sense why you do update, you know. If you are doing update for just update, people see it like, ‘OK, there are some cheesy things. So, nothing is there.’ They may even give you [a] bad review if you do updates without changes. It has to be a point, I think the better is regular updates each time with some content and you know, longer times, huge updates.” (Startup T)

Most of the startups tended to publish their game globally without any restrictions except for some markets that required additional work, due to permits, complexities and/or cultural differences (e.g. China, Russia, South Korea). In publishing the game globally, some startups offered an English version of their games first, while others tried to localize them initially for a few languages. However, in both cases, the startups tended to extend their localization to more geographical markets in future. The data that the startups collected by monitoring the game could guide them to find their major markets and prioritize them for localization. In line with monitoring the game, they also considered users’ reviews and requests for localization.

“We have detected which countries had a huge potential [and] localized in those languages.” (Startup D)

“Many times, you write something [and] you put it into the game, and then – at a certain point – someone writes you: ‘Hey, that’s the meaning! But that really isn’t catchy in my language. What about using the new stuff?’ So, localization goes through languages, and then, always feedback helps us to improve that.” (Startup G)

The startups did not plan for fast monetization. Instead, they focused on creating a mass of users in the first place. So, deploying monetization was among the future improvement plans. In addition, in some cases, they found that the
initial in-app-purchase monetization mechanism was not effective and they had a low conversion rate of users who paid for in-app-purchase items. Accordingly, improving the monetization mechanism was among the future development areas of the game.

“Oh about monetizing, if you are doing a free to play game, you have to think about monetizing in the first months. I mean it’s inside the concept and the design, and then, of course, after you release the game you have to improve your monetization.” (Startup L)

By understanding users’ behavior on the WP platform, the startups could set out their multi-platform plans as well as their monetization plan for other platforms.

“We have this 90-day period from the Windows Store. So, then, when we launch in iOS and Android, it will be even better, because we will see or hear feedback from our users and it will create even more [...] It’s compared to a soft launch to test it, improve it, and then release it to the world in all the platforms, all the strategies, all that.” (Startup M)

Fixing bugs in the games was another area of improvement after their release. Although the startups beta-tested their games before publishing them, sometimes users faced bugs due to the diversity of mobile devices, unknown to the startups before releasing the game.

“There were some bugs. Usually not a bug in itself; they find a bug when they play. We listen, and we fix right away. We always listen, we always read, and we always analyze what they say. There is a bug, and they complain that they don’t have enough content, so we are working on those.” (Startup T)

Multi-platform Activities
The open code ‘multi-platform activities’ reflects the startups’ tendency to enter other platforms after the 90-day exclusivity period on the WP platform.

The startups believed that in order to be successful on popular and more competitive platforms, their games needed to have superb design quality. Therefore, they tended to use the WP platform for soft-launching to understand the strengths and weaknesses of their games and users’ behaviors. Accordingly, they intended to update their games and publish them on other platforms when they were completely happy with the game quality.

“When we started, we were thinking about, ‘OK, after that we’ll port to iOS and Android,’ but it might actually depend on the success of it [...] On Android and iOS, you need to have either something really cool or be really good at user acquisition. And that’s not something that we can handle at this moment – to pump lots of money [for] user acquisition – unless the game has proven itself on the Windows Phone that we can earn it back.” (Startup D)
For most of the startups, offering the games on the iOS and Android platforms was the default path for porting to other platforms. They found these platforms interesting for different reasons. Some startups believed that being active on iOS was an indication of being a serious game developer and wanted to build a reputation in the game industry by going to iOS. In addition, they found the iOS platform to have high potential for revenue-making, where users were more likely to spend money to purchase features. On the other hand, the startups considered Android as a large-scale market with a high potential for downloads and attracting a mass of users which they could capitalize on.

“In the months ahead, I definitely plan to ship on iOS just because everyone’s on there. I don’t know how well it’ll do in such a saturated market, but the point is I can’t go around saying I’m trying to bring [my game] to the masses if I’m not on iOS.” (Startup K)

Among the interviewed startups, those with more intellectually oriented games (e.g. book reading and puzzle games) tended to consider platforms where more users were interested in reading, such as the Amazon store. They thought that the target audience on those types of platforms were used to paying for content, so there was a high chance of monetizing.

“Yes, I want to target Amazon very much, because I know these are really good clients. These people, they read. First, they like to read books, so they’re more intelligent in a way. And they pay, all of them are paying. So, anyone that has a Kindle has paid for something for the content. So, it’s much easier to attract them […] Of course, I want iOS because they pay. And Android because it’s massive.” (Startup J)

“Android. Google Play probably makes me the most at, like, $60 or $70 a day, and then Kindle Fire is much lower, but the conversion rate on Kindle Fire is the best. I just don’t get many people getting my free game – for whatever reason – on Kindle Fire, but the ones that do get it, they’re more likely than on Google Play to go and get the next one.” (Startup K)

A few startups considered the option of porting the game to consoles. Some wanted to leverage their networks with a specific platform owner (Microsoft in this case) by offering the game to the platform owner’s console platform.

“Because we use Unity, it’s really easy for us to make the cross-platform game for iOS and Android, and we really tried to go for Xbox One program to make the game a little bit more than just a mobile game, for put[ting] it in the console, Xbox One, and maybe PlayStation.” (Startup R)

Interestingly, one of the interviewed startups with a puzzle game (J), planned to also consider the offline world for the game. They planned to offer the game for free to newspapers distributed on public transport with a link to the mobile game for interested players. The startup hoped to raise awareness of the game through offline channels and to increase people’s interest in playing the digital version.
“I offer it to newspapers, and they put it on and then they put a link to the game […] I’ll wait a bit until I have it on Android and iPhone, then everybody can see it and know the name. Then it can spread out. Then I can give the newspapers for free, they don’t want to pay. It’s hard to get money from them. But this is the best promotion I can ever get.” (Startup J)

Generally, the rule of 90-day exclusivity on the WP platform did not become an obstacle to the startups’ multi-platform activities. Instead, the startups turned it into an opportunity to collect data on the performance of the game before launching it in the more competitive markets of iOS and Android.

“It’s also an awesome way to soft-launch and get, ideally, a lot of publicity and a lot of downloads, and then make the iOS people wait for it. So then, you know, obviously we hope to play that down the line into a great iOS/Android launch by being like, ‘Finally! The moment you’ve been waiting for!’” (Startup A)

Summary of Post-Launch Activities
The selective code ‘post-launch activities’ describes the activities that the startups performed to improve the games after publishing. The startups updated their games by considering actual users’ behaviors to fix bugs, increase user retention, create more content, localize the game for the target markets, and implement/improve the monetization mechanism.

The startups considered preparing their games for porting to more competitive platforms during the exclusivity period. Monitoring the performance of the game enabled them to see the strengths and weaknesses of their games and to improve them before porting to other platforms. The platforms were chosen based on the potential for success and monetizing for the game type.

5.3 Insights on the Game Performance
The selective code ‘insights on the game performance’ contains three open codes: 1) getting insight into users’ behavior in the game, 2) getting surprised by the download volume, 3) regret about the past activities.

Getting Insight into Users’ Behavior in the Game
The open code ‘getting insight into users’ behavior in the game’ describes the fact that the startups’ learning about the game performance could be contrary to their expectations. The contradictions were related to various aspects such as game play, buying in-app purchase features, and users’ reviews.

In some cases, the startups had an incorrect expectation of the users’ ability to overcome the game challenges. After publishing the game, they found out that the challenge for the users was not as easy or as difficult as they had considered.

“We’ve found out that it’s much harder to play it on the phone than it is on the computer, because you can’t see all the pictures anyway. And it’s a mind game.
It’s harder [...] So, I was afraid that on the phone people would say, ‘OK, let’s go to the fastest level.’ But it was completely different.” (Startup J)

“We found out at the beginning of the game is too hard, and then the ending is too easy. So, [we are] changing the curve of challenge.” (Startup B)

Interestingly, one of the startups (J) thought about the psychological effect the game messages had on the users. In their case, they had tried to remind the users how smart they were if they could accomplish the game challenge. However, they thought that they had overemphasized the message, which made people disappointed if they failed to pass the level.

“So, now my idea is that when you read up all the time you think you’re smart. If you don’t think you’re smart you won’t touch it, you’ll be afraid.” (Startup J)

In some cases, the games received unexpected negative or positive reviews by users. The founder of startup (D) was surprised by the users’ good reception of his design in the game, as he had never considered himself as a ‘designer.’

“Instead of calling my designer to make a design, I made it myself, and I’m not a designer. Nonetheless, maybe I should become one, because it got excellent reviews on the design. I don’t understand why.” (Startup D)

Some startups had experiences with a few unexpected types of reviews resulting in unfair star ratings.

“You can never get five stars, because there’s always someone who gives – like, man, the other day, we had someone who was like, ‘This game is too addictive. I can’t stop playing it. So, I’m giving it one star.’” (Startup A)

“Of course, some people would put like one star, ‘Not my thing.’ Then, we find someone who says, well, my favorite one was like, ‘I don’t have a mobile yet, looks good. One star!’ It looks like, ‘... but I don’t have a mobile yet, my father said Christmas maybe, but let’s see, I will review it back.’ I was like, ‘What??!!’ So, you’ll find these crazy comments.” (Startup O)

In some cases, the monetization mechanism of the game did not work as expected, which was discovered only after publishing the game.

“Actually, that’s one of our problems, because there is not much incentive to purchase, because you can get almost everything for free.” (Startup N)

Getting Surprised by the Download Volume

The open code ‘getting surprised by the download volume’ reflects the startups’ facing an unexpected download volume. On some occasions, the startups received a large volume of downloads in the past when they had just developed a game for fun or for testing and learning.
"I didn’t expect this to be honest. I knew that I am doing something [that] I like. I knew I am doing something that I would love to play. You know, you never know. If [...] you’re doing something just for hard-core players, [it is] going to be just few people that really enjoy what you do. You can’t say it before people say it [...] and people are saying it, so I am happy for that.” (Startup T)

“Also, we did advertising but we didn’t spend too much; just a little bit to spread the word, and then we got many downloads. The first days we got something like 2,000 to 3,000 downloads per day, so it was very easy to have people play our games.” (Startup G)

Obviously, the startups were aware that free games were much more likely to be downloaded than paid ones. However, some startups with strong beliefs in their game quality and attractiveness were shocked that their past paid games received an ‘unexpectedly’ low download volume.

“It was a cool game but I put it for money [...] and I was shocked that no one actually want[ed] to pay in the market.” (Startup O)

“It was actually a paid app at first, but the sales were so low – less than 1,000 sales. So, I just made it free with no monetization at all. I didn’t expect anything about it, then I just went, ‘Oh my God! So many downloads! I could have put some ads in there and make some bucks.’ [...] You can’t know the markets. I didn’t do anything, I just put the game on [for] free.” (Startup N)

**Regret About the Past Activities**

The open code ‘regret about the past activities’ describes the startups’ regret about losing opportunities in regard to getting downloads or monetizing in their past activities. The regrets were mainly a result of their lack of knowledge about the business side of game development. Consequently, they had made mistakes in selecting target markets, marketing the game, localizing the game, and monetizing.

“We did it in all countries except Brazil, Russia and China. Now we open it to those markets. It was a mistake that we didn’t do it in the beginning,” (Startup J)

“[Our previous game was] totally free. Totally free. Very non-professional.” (Startup A)

“We didn’t know that it was that cheap to localize. If we would have known it was that cheap to localize, we would have probably did it before launch [...] We thought that it would be much, much more expensive.” (Startup N)

**Summary of Insights on the Game Performance**

The selective code ‘insights on the game performance’ reflects the startups’ understanding of their games’ performance after publishing them. It indicates that the startups could have incorrect expectations and assumptions about the
users’ abilities, opinions, and behaviors. Therefore, they faced some issues after publishing their games in terms of the difficulty of the game challenge for users, receiving unfair reviews, having an ineffective monetizing mechanism, and receiving an unexpected download volume. The startups were regretful about mistakes they had made in the past in not exploiting opportunities because of their lack of business knowledge.

5.4 Success with the Game

The selective code ‘success with the game’ is comprised of three open codes: 1) defining success by the download volume, 2) defining success by the revenue volume, and 3) defining success by personal achievements.

Defining Success by the Download Volume

The open code ‘defining success by the download volume’ reflects the attention paid by the startups to receiving a high volume of downloads as their first success measure. A common belief was that this would enable them to create a reputation and also be useful for possible future income on the WP and other platforms.

“If I make money I will be happy, but it is not the measure of the success that I’m looking for. I want downloads. I want to try the marketing [by] going as much as I can to get the downloads [and] the conversion rate on the other side. It would be great if I can raise it and make a ton of money, but my goal is to try to get downloads for this title.” (Startup F)

“Our success will not be in terms of money, it will be in terms of downloads, because first of all we want to increase brand awareness. So, in reality we’re not waiting for money.” (Startup M)

Since AppCampus had defined some success measures for its funded apps (i.e. number of downloads, revenue, and user-rates), some startups chose these criteria as their success measures. They aimed to build a good relationship with the platform owner for exploiting future opportunities.

“At least 500,000 downloads in the first year, 4.3 star rating or up; basically, the criteria that they [i.e. AppCampus] are looking for. Because we want to continue our relationship with Nokia and Microsoft. So, our main goal is to kind of comply with their demands or see what they’re looking for.” (Startup I)

“My goal is to reach actually the console, so to impress Microsoft. Because Xbox One console is only to top of the top. So, if you are an indie accepted to Xbox One, then you are in good hands. You did it! Now, if you did this or whatever, a good game, you are good, financed for a couple of years.” (Startup O)

“So, we want to have more than 3 million downloads. We want to be the top AppCampus game. That way, they will promote us a lot, because they always talk about [their top game]. Well, I want them to talk about [my game].” (Startup M)
Defining Success by the Revenue Volume
The open code ‘defining success by the revenue volume’ describes the attention paid by the startups to making an income from their games in order to survive in commercial game development. However, profit-making was not their immediate measure of success. For their first commercial game, they only considered the amount of income enough for survival rather than profit-making.

“We’ve gone pretty far on the sweat equity aspect of it, but that’s kind of maxed out now, so we would need the revenue to then build more content. That’s the only way to do it or investment or something.” (Startup A)

“If we made enough money to make the other game [i.e. next game]; [...] if we make €50,000, that’s enough to live for a very poor salary. But still, we basically would live on [the] same standard of living that we have been living for the past year.” (Startup P)

Defining Success by Personal Achievements
The open code ‘defining success by personal achievements’ reflects the attention paid by some startups to succeeding in their experimental endeavors and learning aspects of game development. These startups considered accomplishing the project and publishing the game as their success measure. This approach was taken by those startups who wanted to implement an innovative and experimental factor in their games or who had a side revenue from contract work.

“The goal of this game for us is to see if we can make like those other storytelling games in the PC game market, to see if we can adapt it and figure out what’s not working and figure out how we can fix it, to see if we can find a storytelling model that we can then launch other games with. We’re trying to find a new way of storytelling, so this is the experimental game.” (Startup A)

“When, people say, ‘I really like this. I really like this. It’s fun. I love the artwork. I like this,’ that’s when I’ll feel I’ve achieved something and that we’ve achieved something as a company as well; because we all believe in the same thing. It’s not about money for us.” (Startup E)

Summary of Success with the Game
The selective code ‘success with the game’ described how the startups define the measure of success for their games. For them, success was having a high volume of downloads in their early stage of business to build a reputation and a mass of users, whom they could monetize later. However, for the monetization objective, they mainly considered covering the costs and surviving rather than profit-making. The definition of success for the startups was also affected by the accelerator’s defined measures, because some startups were aiming to create a good relationship with the platform owner. The startups that wanted to implement some type of innovation or experimentation in their game or had
external contractual works, considered their success as lying in accomplishing the current game project and publishing their games.

5.5 Restraints

The selective code ‘restraints’ consists of three open codes: 1) restraints by AppCampus, 2) restraints by the platform, and 3) restraints by supporting tools.

Restraints by AppCampus

The open code ‘restraints by AppCampus’ reflects the limitations that were imposed by AppCampus. These restraints are categorized into three groups: 1) restraints by the support timing, 2) restraints by the AppCampus’ interests, and 3) restraints by the AppCampus’ change of strategy.

Some startups thought that the timing of being invited to participate in the AppCademy was not appropriate considering their development progress. They believed that they should have participated in the AppCademy before publishing their games or in an earlier stage of their development process. This was mainly due to the fact that startups learned the proper launch strategies during the AppCademy and also learned that the monetization mechanism should have been designed in their game concept. Making changes was difficult for them and created a lot of extra work in the later stages.

“The game was never designed as a free-to-play model; they [i.e. AppCampus] knew that right from the beginning. Nobody ever mentioned making it free to play until I came here last week and now it’s all, ‘It should be free to play. You should have made it free to play.’ It was never designed like that, and you tell me [this] when the game’s finished.” (Startup E)

Another restraining factor was the timing of delivering the funds to the startups. The delivery time was designed as 30 percent of funding upon approval of the design document, and 70 percent after launching the game. Some startups believed that receiving the funding after they had done the job did not help them to hire the right people to perform the tasks more effectively and efficiently during the development period, and this gave them extra work to fix the problems.

“Until you get to this 30 percent [of the fund] you have to do everything on your own […] Then came the worst part, because now the 70 percent you get only when you finish. I said, ‘Even if I had 70 percent at that moment and said, OK, now I’ll get a good designer and I’ll get whatever I need.’ You know, I can split the money better. So now the money came, it’s good money, it’s nice, but it’s like you’ve finished everything, so you don’t need the money anymore.” (Startup J)

Another restraining factor for some startups was the lack of customization of AppCampus for their specific needs according to their game characteristics. They believed that AppCampus’s training was focused on getting downloads without considering the startups’ objectives. These inconsistencies between the objectives of AppCampus and the startups were mainly related to issues
such as the monetization plan, the experimental innovations in the game, the single game versus a portfolio of games, and launch plans.

AppCampus’s training suggested offering free-to-play games initially to attract a large volume of users; however, some startups were more focused on having a few users but providing them with a special experience. One of the affected startups (E) was planning to publish a paid game in a week but was criticized by AppCampus advisors about the monetization mechanism and was advised to change it to in-app purchase. Another startup (C) was seeking to implement an innovative artistic concept in their game and wanted to create a special user experience inside the game. Their focus was not on receiving a high volume of downloads in their early stage, but rather on creating their desired experience for the users.

“Then AppCampus has been great. Yes, but I think it’s very heavily focused on downloads, which I can understand, because it’s platform holders. They want the downloads – that’s what they want. That’s not necessarily what’s best for everybody [...] I think there’s a different market for different things.” (Startup E)

“Then a layer of the AppCampus are the talks and the support and everything. I think it’s a little different for us because we are definitely in the art game camp, and I don’t know if I would go to [a mentor] and say we are not interested in monetary success right away.” (Startup C)

In one startup’s case (A), their strategy of publishing the game was different than the strategy advised by AppCampus. Where AppCampus’s advice was to launch the game globally to all markets, this startup wanted to launch the game to a small market in order to have the possibility to improve it before publishing the game to their main markets. They were concerned that these different desires might create a conflict with AppCampus.

“We’re now thinking of launching in only a few of our less popular markets, actually, until we have some of these really cool features [...] We know from experience that usually comes a few iterations down the line. So, this idea of maybe only soft-launching in a few places is very interesting. We have to weigh that in with what AppCampus may or may not be happy with in terms of immediate download numbers.” (Startup A)

The newness of the WP platform and consequently the changes in its strategy and AppCampus’s strategies forced some restraints on the startups. For instance, initially, using the ‘Try and Buy’ monetization model was advocated by WP and AppCampus. Within this model, users could use the game for a certain period of time, and then had to pay to continue to use it. However, with the change in the platform owner’s strategy for promoting freemium games, developers were strongly advised not to use the Try and Buy model. Therefore, those startups who had followed the previous advice faced difficulties changing the monetization mechanism.
“When we applied, the application strongly encouraged this Try and Buy model that Microsoft was pushing with Windows, and when we got here, everyone was like, ‘Uh, I don’t think you want to do that, if you know what I mean.’ So, then we were like, ‘OK, so now we need to rework the whole game to be [an] in-app purchase. It’s actually quite painful, because it’s very different.” (Startup B)

Restraints by the Platform

The open code ‘restraints by the platform’ reflects the limitations and/or strategies of the WP platform that were imposed on the startups. The main restraints fell into two groups: 1) preference-related conflicts, and 2) store-related conflicts.

The WP platform required the startups to design their games according to the mobile device’s hardware specifications. If the startups did not follow the rules, their games were not approved by the quality assurance team. One of these rules was to use the hardware ‘back button’ which already existed on the phone. However, many developers used a virtual back button in their games, which they had to remove, and redesign the game to use the hardware button.

“Oh, you shouldn’t have a back button, because every phone has a hardware back button. So, they didn’t want it. We would have wanted it, because it’s better for users, but they actually forced us to remove it. So, we had no choice. But it looks OK. It was no big deal.” (Startup N)

“They returned the app I think four times or five times. Usually the problem was dedicated to user experience conflicting with user experience of the game and UI [i.e. user interface] of the phone […] Sometimes, especially on smaller screens, there was [a] problem with [the] readability [of the text].” (Startup S)

Another restraint related to the preference of the platform was not allowing the developers to directly contact users who left reviews on the store. Many of the startups mentioned this limitation as a problem in serving the users. However, to cope with this restraint, the startups designed their own channel inside the game to allow users to contact them directly.

“The rating system [has limitations]. On Windows Phone, you can’t answer that player. I think that they should make an option to answer, because I read all the reviews and I want to answer and to chat with some of the players, but I can’t. In the game, there is a support button they press and they can send us an email. Through that we can talk a bit.” (Startup H)

“We have a help button, they can email us. It’s all in the game.” (Startup E)

Some startups with more innovative and multidisciplinary games faced a restraint related to the store design and limitations. They could not find a proper ‘category’ in the store to publish their games.

“We were looking at it [to choose a right category for the game], and we were going through categories and nothing really made sense. So, I don’t know [...] we
might ask Microsoft - because we have good Microsoft connections in Vancouver – if we can get a new category made for it.” (Startup C)

“So, I put it in the shooter category, but in Brazil this category is translated to something that doesn’t match the game exactly. So, we decided to move it to action.” (Startup H)

**Restraints by the Supporting Tools**

The open code ‘restraints by the supporting tools’ reflects the limitations of the analytics tools or development tools for the startups.

One of these restraints related to the incompatibility of some development tools with the platform due to ongoing changes. As one startup highlighted, they had to delay their development due to a problem that the Unity engine had with the WP platform.

“The Unity integrations of the Windows Phone exporter are still very new. So, we’ve been hung up a few times. We find a bug in the Windows Phone 8 integration in Unity and then we have to wait for Unity to fix it.” (Startup C)

Another restraint was the challenges that some startups had with the analytics tools. In some cases, these tools did not provide the exact data the startups wanted to collect. Consequently, the startups had to combine multiple tools in order to receive their desired reports.

“For example, the revenue coming from ads, you cannot have an API with them and see from which country it is. You have to download a report and put it either in your system or an Excel sheet. It’s the only way to do it.” (Startup D)

In addition, the lack of integrity between the data provided by various analytics tools was another challenge mentioned by some startups.

“I want to get a clearer picture of the retention [...] I don’t trust the numbers so much. I get numbers out of Flurry and Google Analytics and they’re different and like, ‘Crap, why?’ I need to get some visibility into that.” (Startup F)

“There is one thing I never understand in this game; the number of downloads was half a million but the Flurry Analytics said that at least one million people played the game. I don’t know if it was a bug from Flurry or piracy.” (Startup H)

**Summary of Restraints**

The selective code ‘restraints’ indicates the limitations that were imposed on the startups by the accelerator, the platform, or the supporting tools.

The restraints caused by the accelerator were related to the timing of support, conflicts of interest between the accelerator and startups, and changes in the strategies of the accelerator. Overall, these restraints created extra work for the startups to amend their games according to the accelerator’s requirements. Some startups thought that the training should have been delivered earlier in
their development process before launching their games. They also thought that the timing of funding did not necessarily help them to hire the right expertise in time. In some cases, the success goals of the startups were not in line with the success measures defined by the accelerator. While the accelerator’s training was focused on getting a high download volume, some startups were seeking other objectives with their games. Some strategies of the accelerator changed due to the modification of the platform strategies, which created extra work for startups to adjust.

The platform-caused restraints were related to the platform owner’s preferences and the store’s limitations. The startups had to adjust their designs according to the hardware specifications of the mobile devices as required by the platform. The startups were also not able to contact users directly through the store and had to build other channels to contact them. Further, some startups faced problems with store limitations in terms of finding a suitable category for the game.

The supporting tools restraints related to the limitations of the tools the startups were using in developing or monitoring their games. The ever-changing aspect of the technological tools and the platform caused some incompatibility between the platform and the third-party development tools. The analytics tools did not always provide the startups with their desired data and sometimes they had to combine several tools to achieve their purpose.

5.6 Discussion

In the final phase of coding in the second category, I drew the relationships between the five selective codes of monitoring activities, post-launch activities, insights on the game performance, success with the game, and restraints (Figure 7). The aim was to conduct the ‘theoretical coding’ in a similar manner to the first core category (e.g. Glaser, 1978; 1998). Subsequently, I will discuss the theoretical coding in the light of the extended literature.

Relating the Five Selective Codes

Figure 7 illustrates how the selective codes in the second core category relate to each other.
The startups initiate their monitoring activities using both analytics tools and user reviews in order to understand how their games are received by the users, their behaviors and their opinions. So, the startups understand the strengths and weaknesses of their games and get actual insights into the performance of their games against their perceptions. The insights help the startups to realistically define and approach their success objectives. The insight also guides the startups to undertake post-launch activities about the quality of their games and to prepare them for porting to more competitive platforms. The games’ quality is raised in terms of increasing user retention, creating more content, improving or applying the monetization model, and localizing the games to serve the target markets. The post-launch activities are facilitated by the monitoring activities by collecting data from actual users and markets using technological tools. In turn, the post-launch activities facilitate the achievement of the defined success for the startups, which is mainly focused on receiving a high volume of downloads to pave the way for future revenue-making.

Nevertheless, there are restraints that affect the post-launch activities, restrict monitoring activities, and sometimes contradict some elements of the success objective. The restraints of some analytics tools in providing the desired data for the startups, and the limitation of the platform to allow direct contact with users, restricts the startups’ monitoring activities, and causes them to seek ways to cope with the limitations. Furthermore, the restraints can cause additional work in the post-launch activities due to changes in platform strategies, lack of timely support during development, and/or changes in game categories in the store. Meanwhile, the restraints of the encouragement by the AppCampus to focus on download volumes can contradict some success elements for startups, particularly if they are not immediately looking for downloads. Figure 8 shows the presented discussion in more detail by showing the properties of each relationship.

Figure 8. Excelling in the market – detailed view.
Taking a similar approach to the first category, I compare the concepts generated from the empirical data with the extant literature (e.g. Glaser, 1992; 1998; Hekkala and Urquhart, 2013; Urquhart et al., 2010). For this purpose, I use the extended literature on the iterative approach in software startups that I presented in Section 3.2.2 to discuss how the emergent model confirms, contradicts, or extends the existing theories (Urquhart et al., 2010).

**Theoretical Integration of Excelling in the Market**

The findings confirm the earlier studies that the scarce resources of early-stage startups force them to focus on only one game at a time rather than on multiple games (Li et al., 2013). They also indicate that startups undertake a sequential experimentation rather than simultaneous experimentation approach to improve their games after their release (cf. Davis et al., 2014).

The findings are in line with earlier studies that startups follow their founders’ experiences to conceptualize their game ideas, develop their games (Olsson and Bosch, 2015), and beta-test the games in small groups before launching. However, startups may face unexpected challenges and problems after publishing the games, which confirms the unpredictability of the game players (Koskenvoima and Mäntymäki, 2015).

**Using Continuous Experimentation**

To reduce the uncertainties regarding the reaction of the customers to a software product and by considering the limited resources of the software startups, the extant literature emphasizes taking an experimentation approach to gradually improve the product based on data collected from customers (e.g. Fagerholm et al., 2017; Giardino et al., 2016). However, for mobile game startups, the literature suggests not taking this sequential improvement approach due to the high competition in the market (Yin et al., 2014). However, the findings show that the early-stage startups took a sequential experimentation approach and relied on gradual improvements of their games and markets based on user data.

In software startups, the type of data collected from users has been found to be different for various stages of product development (e.g. Bosch-Sijtsema and Bosch, 2015; Olsson and Bosch, 2015). In earlier stages of development, startups use qualitative data which is deliberately shared by the customers. In the later stages, the focus is on quantitative data from a large number of customers without the users’ control. The findings show that in the context of mobile games on the platforms, the startups exploited both quantitative and qualitative data after publishing their games. They collected the quantitative data by using the game analytics tools. In addition, they exploited the qualitative data provided by the users’ reviews as an important source of ideas to improve the game and set future strategies. Besides collecting the data by game analytics, the users’ reviews were seen as an important source of improvement ideas and strategy setting. This confirms the role of users as an invaluable asset for the mobile game startups and emphasizes the importance of the human capital in building the business models for software companies (Vanhala and Saarikallio, 2015). The exploitation of users’ feedback is not limited to the app
ideation phase, as discussed in an earlier study on mobile app developers (cf. Qiu et al., 2017), but is extended to the improvements made to the game artefact and markets. Nevertheless, it should be noted that not all the user-generated feedback was necessarily fair and useful for the startups; however, the startups managed to leverage them. The possibility to receive and exploit the users’ reviews on the app stores creates a ground for open-ended experimentation for the software startups without any preset assumptions to be validated (Fagerholm et al. 2017). Instead, it allows the startups to discover areas of improvement as well as new potential markets.

The existing literature suggests the creation of a minimum viable product to get users’ feedback (e.g. Ries, 2011; Fagerholm et al., 2017; Björk et al., 2013). However, the level of completeness of mobile games upon release while leaving room for further adjustments can be a challenging decision for the developers. The findings show that the improvements made to the games after the launch were mainly small tweaks to increase user engagement, offer additional content, and prepare the games for scaling-up through localizing and monetizing. At the very early stage of playing the game, the users demand a completely playable game to enjoy in order to continue playing (Li et al. 2014; Merikivi et al. 2017). There have been cases where some famous video games received very negative feedback because of being released with inadequate technical performance, including Superman 64 (1999), SimCity (2014), and Batman: Arkham Knight (2015). In fact, the amendments that the startups made to the game were in terms of adding content, localizing, and monetizing which are all directed to the enrichment of the game with more content.

**Scaling Up**

In the existing lean startup literature concerning the necessity of experimentation using real customer data, the main focus is on finding the product-market fit and the scaling-up phase is not addressed. This could be due to the scarcity of resources in the startups, so a need to invest in having a proper product before scaling-up. The existing literature considers the gradual development of the markets so that the product is delivered to an existing set of early adopters and then extended to future markets when it has been proved (e.g. Nguyen-Duc, Shah and Ambrahamsson, 2016).

In the platform-based market, the interviewed startups released their games on the platforms at a global level. They targeted all the potential users at the outset and collected data on a global scale to understand the markets with potential. Eventually, they invested gradually in localizing their game in the markets where potential existed. This is in line with entrepreneurs’ logic of using the means at hand to exploit opportunities (Sarasvathy, 2001). Taking this approach, the startups turned the heterogeneity of the user expectations and their cultural differences into an opportunity to figure out the right market for their games (cf. Davis et al., 2014). Of course, it is worth highlighting that the startups were already quite certain about the quality of their games under the mentoring received from the accelerator and the quality assurance approval. In the absence of more formal quality assurance, startups may want to deliver their games first to a test market on the platform, for example a specific terri-
tory, to make sure of its performance quality (Koskenvoima and Määttymäki, 2015). Therefore, the scale-up and launching phases were to some level intertwined in the platform-based market setting for mobile game startups. They could immediately experiment in the global market on a large scale and then narrow down to some specific markets and provide more targeted marketing.

Coping with the Platform Owner Rules
Despite the existence of the exclusivity period, all the startups considered launching on multiple platforms due to the ease of porting. However, they were realizing the specific characteristics of each platform and wanted to enter more competitive platforms with more preparation to decrease the chance of failure. So, the startups were more careful about having a polished product when entering more competitive markets. While Koskenvoima and Määttymäki (2015) highlight the role of soft-launching by considering smaller geographical markets, in this study, the startups considered the less competitive immature platform as the place for soft-launching.

The interaction of the startups with the platform owner in terms of the imposed restraints was in the forms both of accommodating and resisting the platform rules (Eaton et al., 2015). Startups were accommodating to the platform rules in areas where they had no power against the platform securing efforts (Ghazawneh and Henfridsson, 2013). An example was changing the game design to be compatible with the hardware device specification according to the platform owner’s wish. However, wherever they could, they found ways to meet their objectives through self-resourcing (Ghazawneh and Henfridsson, 2013). For example, they implemented a contact button inside the game design to contact users directly despite the limitations of the platform.

To sum up, the theoretical model in the second core category answers the second research question ‘How do early-stage mobile game startups excel in the market?’ It shows that the startups took an iterative approach in developing and publishing their games. They made improvements in the game according to the collected data from actual usage to provide a better experience for users as well as expand their markets with validated learning. The empirical evidences also indicate that in a platform-based context, experimentation and scaling up can happen in an intertwined manner. Finally, the findings show how the startups took both accommodating as well as self-resourcing approaches to meet their objectives despite the restraints imposed on them by the platform owner and the supporting tools.
6. Dynamics of Success for Mobile Game Startups

I started this study with the objective to understand how mobile game developers operate in their early stages. So far, I have presented how they utilized the accelerator’s facilities in the first core category, and how they tried to excel in the market in the second core category. In this section, I discuss the integration of the findings from the two core categories and the consequent scale-up with the help of the extended literature on the success of mobile games in Section 3.2.3. Through this process, I answer the main research question, ‘What are the dynamics of success for early-stage mobile game startups?’

In the following, firstly, I present a summary of the two emerged core categories and explain how they relate together. Then, through the creative process of scaling-up, I illustrate the theoretical model of ‘Dynamics of Success for Early-Stage Mobile Game Startups’. Subsequently, I use the aforementioned extended literature review to ground the findings.

6.1 Summary of the Findings in the Core Categories

The findings in the first core category, ‘utilizing the accelerator’s facilities,’ show that the startups faced two main challenges: 1) scarce financial resources, and 2) lack of knowledge on business-related issues in game development and publishing. Exploiting the accelerator’s facilities, the startups had more resources at hand in terms of capital, knowledge, and networking, and gained credibility in the market. Having more resources made the startups more capable of producing higher quality games in terms of design and monetization, as well as trying new ideas in their game concepts. Through the mentorship and training sessions in the accelerator, startups were advised to design their market expansion activities in an evolutionary manner. The notion of the platform-owned accelerator enabled the startups to utilize the accelerator as a social boundary resource for transferring knowledge and negotiating with the platform decision-makers, peer-startups, and other members of the ecosystem such as promoters and publishers. The negotiations helped the startups to arrange ‘planned’ visibility in the market to get promoted at their desired times to their desired markets. Furthermore, the requirement to get quality assurance approval before publishing the game increased the quality of the games in terms of design and compatibility issues.
The findings in the second core category, ‘excelling in the market,’ show the evolutionary approach of the startups in the market. The startups developed their games based on their experiences. Consequently, their perceptions of their games’ performance and the users’ behavior were not necessarily supported by the actual market data. Therefore, they published their games and monitored them with the help of game analytics and users’ reviews to address two objectives: 1) increasing user retention, 2) preparing the game for market expansion. They tried to increase user retention by improving the complexity of the game and enriching the game by adding content gradually. They also aimed to expand their markets through localizing the game for markets with potential and polishing the game before entering more competitive markets. Their scarce resources only allowed them to concentrate on one game at the early stage. Meanwhile, the restraints imposed by the platform, the accelerator, and/or the analytics tools could hinder the activities of startups’ improvements, and sometimes required extra effort from the startups to meet their objectives.

6.2 Toward the Core Theme

By integrating the two core categories of ‘utilizing the accelerator’s facilities’ and ‘excelling in the market,’ the big picture appeared to show how mobile game startups operated in their early stages.

The integration of the core categories and its scaling-up indicate that the startups worked toward fulfilling their success objective, which was to accomplish the project and have a large volume of downloads and continued use of the game to allow for future monetization. This was decided considering two logics: Firstly, by adopting a freemium model of revenue-making, startups could benefit from the large number of users that could pave the way toward generating revenue in the future by in-app purchases. Secondly, success in the first commercial game could facilitate receiving more visibility for future games and in future markets through gaining popularity. The startups used the available resources including the accelerator’s facilities to create their games with good quality and appeal to the mass market. They also implemented techniques to encourage the users to engage with the game and leave positive reviews and spread word-of-mouth. Furthermore, they tried to have more control around making their games visible in the market. They noted that visibility was not completely controlled by them, and other environmental factors could affect the visibility as well, such as unplanned featured spots on the market, an accidentally timely review by a review center, or the algorithm of ranking lists.

Subsequently, in order to excel in the market, the startups were prepared to take an evolutionary approach and make changes to the game after its launch with the help of game analytics and users’ reviews. They followed two goals in their evolutionary approach. Firstly, they wanted to improve the game content according to their understanding of the actual performance of the game to increase user engagement and get more visibility in the market. In addition, they wanted to enrich their games in order to customize them for new markets and
future competitive platforms. Following this approach, they aimed to capitalize on their initial success to gain credibility for their future markets and games.

Figure 9 depicts the above-mentioned arguments through conceptualization and scaling-up which has resulted in the use of theoretical constructs. Thus, it shows the theoretical model representing the ‘dynamics of success for early-stage mobile game startups.’

![Figure 9. Dynamics of success for early-stage mobile game startups.](image)

Next, I will discuss the model and its relationships in light of the extended literature review.

### 6.3 Theoretical Integration of the Core Theme

Startups work to offer a product with appeal to the mass market and the potential for scalability (Paternoster et al., 2014; Giardino et al., 2016). For mobile game startups, this could relate to offering a game with potential for success in terms of number of downloads, revenue generation, and positive user reviews (e.g. Carare, 2012; Garg and Telang, 2013; Ghose and Han, 2012; Liu et al., 2014). However, research shows that app developers may concentrate firstly on downloads volume at the cost of delaying monetization (Bergvall-Käreborn and Howcroft, 2013b). The findings of this study confirm this attitude of developers by showing that the startups under study defined success for their first commercial game as creating a large number of engaged users who would generate positive reviews and word-of-mouth. In line with the findings of Li et al. (2013) and contrary to the suggestion by Davis et al. (2014), the startups focused only on one product in their early stages due to their resource scarcity. They intended to make this game a success to be empowered for future games and markets.
The findings show that the startups were created by people who were game players and gaming fans and had moved from hobby development toward commercial development. Departing from this point, the startups were mainly creating games according to their founders’ experiences and knowledge of gaming, as previously observed (Davis et al., 2014; Vanhala, Kasurinen and Smolander, 2013). This could increase the risk of a biased view, with a consequence of failure in offering a product with appeal to the mass market, due to their lack of knowledge and market-validated learning (Giardino, Wang, et al., 2014). In addition, their lack of capital forced them to mostly invest their resources into creating the game, rather than conducting market research.

Dynamics of Quality, Visibility, and User Engagement

Through the combined utilization of the accelerator’s facilities and an evolutionary approach to excelling in the market, the startups could get extra resources and manage their exploitation of the resources. Participation in the accelerator could help the startups to create a product with better quality in all aspects of SERVQUAL except for responsiveness. During the accelerator program, the startups could improve their games by taking advantage of the cultural capital (Haines, 2014) of networking and sharing knowledge with peer-startups (Qiu et al., 2017) and validating their games with market experts (Dempwolf et al., 2014). This opportunity, along with the existence of the funding, allowed more investments into aspects related to aesthetics, content, novelty, monetization mechanisms and the integration of social features, to name a few. All these issues relate to the concept of the tangibility of the game as a quality element in SERVQUAL (Parasuraman et al., 1988, p. 23). Furthermore, the requirement to go through quality assurance approval before publishing the game could enhance the reliability of the game in terms of compatibility with the platform standards and the users' mobile devices. Consequently, through the advice of the platform owner (Ghazawneh, 2016), startups created their game artifacts with an acceptable quality that was certified by the platform owner, as is expected when a corporate accelerator is involved (Haines, 2014). The direct promotion of the games by the accelerator in both their online and offline channels, and its securing that the games were in line with the platform requirements (Ghazawneh and Henfridsson, 2010), along with having the accelerator’s support, could increase the credibility and assurance of the game for members of the ecosystem such as promoters, publishers, and users (Miller and Bound, 2011). However, the responsiveness of the games was limited by the platform structure by not allowing the startups to directly contact users who left reviews. This issue has already been observed in other platforms as well (Qiu et al., 2017). However, the startups overcame this problem through self-resourcing (Ghazawneh and Henfridsson, 2013) and designed an in-game contact button for users who wanted to contact the developers. They also exploited social media by having company pages and dedicated game pages through which they could directly communicate with users.

In addition to the quality, startups also tried to increase the visibility of their games and their firms in the market using the accelerator’s facilities. They were seeking visibility to increase the chance of getting a large volume of
downloads for their games, confirming the existing literature on the positive relation between visibility and download volumes (e.g. Garg and Telang, 2013). Although the startups were aware that uncontrolled factors could affect the visibility, they were seeking ways to have a more control over this and were investing in getting into top-ranked positions (Liu et al., 2014) by negotiating the time and place of getting featured and harmonizing it with their strategies in the related markets. This confirms the awareness of the startups about the limitations and ineffectiveness of the ranking lists (Bergvall-Kåreborn and Howcroft, 2013a; Bresnahan et al., 2014).

By working on quality and visibility, the startups were trying to increase user engagement with their games (O’Brien and Toms, 2008). The efforts put into the visibility and quality of the store-shown data (e.g. icon, description, languages, accelerator’s watermark on the game icon) were meant to affect the point of engagement (e.g. Lee and Raghu, 2011; Mohd et al., 2016; Wang and Li, 2017). Further, trying to get positive reviews from users was another means to increase the effectiveness of the point of engagement, by increasing the motivation and interest (O’Brien and Toms, 2008) of future users, considering users’ reliance on peer reviews (e.g. Ghose and Han, 2014; Lee and Raghu, 2014). Startups used the mechanism of asking for user ratings and reviews while aiming to get reviews from satisfied users rather than random and possibly dissatisfied users.

To prolong the period of engagement, the startups were prepared to fix their games and raise their quality after launch according to data collected from the actual performance of the games. Naturally, in line with the existing literature on the significance of offering polished games (Yin et al., 2014) with adequate features (Qiu et al., 2017), the startups tried to offer technically reliable games with high potential for attracting users. They performed beta-testing prior to release and polished the games with the feedback they received via the accelerator’s facilitation. However, they were also prepared to adapt the games to the users’ expectations in case of observing problems after release, to enrich the games’ content and customization for current and future markets, and to re Fine the monetization mechanisms. These findings confirm the exploitation of users as a key resource for app developers (Vanhal’a and Saarikallio, 2015), and the sequential approach of game development (Davis et al., 2014), and are contrary to the suggestions to avoid fixing games and moving on (Qiu et al., 2017; Yin et al., 2014). This could be due to the lower competition on the platform under study compared with the more competitive Apple platform in the above-mentioned studies. It should be noted that despite the expertise of the developers as game players, they still faced unforeseen user expectations and behaviors, confirming the findings of Ghazawneh (2016).

It is worth noting that the quality improvement of the games was rarely related to the technical quality, but more broadly addressed the enrichment of the game content and user engagement. Using game analytics to get actual market data, startups could improve the design of the games in terms of SERVQUAL’s tangible elements (e.g. extending content and monetization mechanism), as well as empathy elements (adding customized in-app-
purchase items and localization). The findings extend the findings of Koskenvoima and Mäntymäki (2015) on the use of analytics by game developers. While they observed the effect of analytics on the design phase of the software, the findings of this study show that in case of early-stage mobile game developers, analytics are a decision support tool beyond the design phase, and actually drive decisions on market expansion as well. The startups localized or aimed to localize their games and started to invest in marketing activities in new markets with the guidance of the analytics. The improvements were not only made through quantitative analytics, but also by the help of user-generated reviews and opinions (cf. Fagerholm et al., 2017). The findings indicate that the users’ feedback are not only used during the development period of games or during app ideation (cf. Qiu et al., 2017), but can be used for further extension of the market and enriching the game quality in the SERVQUAL aspects.
In this chapter, I will summarize and conclude the study in the form of the theoretical and practical contributions and close the chapter by presenting the limitations of the study and avenues for future research.

### 7.1 Theoretical Contributions

The theoretical contributions of this study are threefold. Firstly, I contribute to the literature on the role of the platform owner in affecting the activities of third-party developers through the social boundary resource. Taking the developers' perspective and answering the first research question, I discuss that the platform owner deploys resourcing and securing mechanisms through the accelerator boundary resource which affects the quality and visibility of the games in the market and increases the credibility of the startups.

Secondly, I contribute to the literature on the use of game analytics in mobile game startups. Answering the second research question, I explain how startups excel in the market through continuous experimentation by collecting both qualitative and quantitative data to improve their games and expand their markets.

Thirdly, the key contribution of this study is the theoretical model of the 'dynamics of success for early-stage mobile game startups.' The model illustrates that startups took an experimentation approach to further increase quality, visibility and user engagement for their games. It explains the dynamics of reaching success for early-stage mobile game startups. In the following, I present the contributions in more details.

**The Role of the Platform Owner**

In this study, I contribute to the literature on the role of the platform owner in affecting the activities of developers and extend the body of knowledge on the role of the accelerator as a social boundary resource. I provide an answer to the call by Qiu et al. (2017) to discuss the role of the platform owner in developers' activities. Further, by focusing empirically on the developer's perspectives, I answer to the call by Ghazawneh and Henfridsson (2010; 2013) and Yoo et al. (2010) to study the governance of platforms by boundary resources from the perspective of third-party developers.

Existing studies on the role of the platform owner through boundary resources have mainly focused on enforcing the compatibility of the mobile apps
with the platform rules (Ghazawneh and Henfridsson, 2013; Schreieck et al., 2016). The role of the boundary resources has been limited to organizing the third-party developers (Bianco et al., 2014) and increasing the shared understanding between them with the platform owner. Nevertheless, none of the defined roles address the effect of the boundary resource on the quality of the apps to include the *market appeal* as well as the developer’s *business development* initiatives rather than solely the technical quality. An exception is the study by Qiu et al. (2017) on the Apple platform where the third-party developers took the existence of the Mac developer community as an opportunity for partnering with successful peer-developers to increase their chance of market success and leverage their quality. Despite the fact that the Mac developer community is a member of the Apple ecosystem, Apple as the platform owner, does not maintain an official role in increasing the app quality (beyond the technical reliability) and the developers’ business quality.

In contrast, I present a direct role of the platform owner in increasing the chance of success for startups by creating a social boundary resource in the form of an accelerator. Through the social interactions inside the boundary resource, the startups tried to increase the visibility of their games and increase their market appeal. Thus, the platform owner provided guidelines and created social interactions with the third-party app developers (Ghazawneh, 2016) and through its control mechanism supported the generativity of the platform, confirming the findings of Remneland Wikhamn et al. (2011).

In addition, I extend the extant literature on the creation of boundary resources. The literature discusses that the creation of the boundary resources is influenced by the internal resources of the platform owner (Mohagheghzadeh and Svahn, 2016a). In this study, I provide empirical evidence of where the boundary resource is created by *external* resources to the platform owner as well as its internal resources. By importing the services of other members of the ecosystem (e.g. publishers and promoters) to the accelerator, the platform owner extends the role of the social boundary resource to facilitate interactions between the various types of ecosystem members rather than only between the platform owner and the developers (Ghazawneh and Henfridsson, 2013; Mohagheghzadeh and Svahn, 2016b).

**Analytics in Mobile Game Startups**

I contribute to the literature on mobile game startups and the use of game analytics in finding market-product fit and scaling-up. I extend the contributions of Koskenvoima and Mäntymäki (2015), who have called for further research on the use of game analytics in game developers, with a larger set of studied startups. While obviously the more established game producers have more resources to spend on analytics, my findings show the value and cruciality of using game analytics in early-stage startups in developing their businesses and excelling in the market. My empirical evidences from a set of 20 early-stage companies show how startups cope with their scarce resources and exploit game analytics and market-collected data to develop their business. Using game analytics, the startups can turn the heterogeneity of users (Davis et al., 2014) into an opportunity and find the right target markets for their games.
The dynamic theoretical model highlights the role of game analytics in driving startups’ activities in their market expansions.

**Continuous Experimentation in Software Startups**

I contribute to the literature on continuous experimentation for software startups and, more broadly, business model evolution in startups (e.g. Fagerholm et al., 2017; Ojala, 2015; Spiegel et al., 2016). My key contribution is the theoretical model of the ‘dynamics of success for early-stage mobile game startups’ with continuous experimentation at its heart. My model is an answer to the call by Fagerholm et al. (2017) for research on when and how to apply continuous experimentation in the design and development of software products in coping with uncertainties.

I explain how the early-stage mobile game startups used experimentation after releasing their games to improve them and find potential markets for further investments. The objective of the continuous experimentation is to understand the areas of investment in both improving the product and expanding the markets. Confirming the service-oriented notion of mobile game business (Hamari and Järvinen, 2011; Hamari, 2015), my model explains that experimentation is used for improving the service quality of the games in terms of the SERVQUAL measures (Parasuraman et al., 1988). It is used to create more empathy with users (e.g. localizing the game and offering targeted customization features), as well as to improve the tangible aspects of the games (e.g. adding more content). However, it must be emphasized that even with continuous experimentation, the initial game release needs to be complete enough to allow users to enjoy playing it (Li et al., 2014; Merikivi et al., 2017). My theoretical model highlights the requirement for continuous experimentation in moving startups forward in their early stages to achieve their initial success objective and prepare for future markets and platforms.

### 7.2 Practical Contributions

The multidisciplinary nature of this study and its location at the intersection of mobile game development, software startups, and business accelerators make the findings useful for several audience groups: mobile game startups, platform owners, and business accelerator managers.

**Mobile Game Startups**

In this study, I discuss an approach that can be used by early-stage game startups in developing and publishing their games. The developed theoretical model guides startups on how to consider the humanistic, technical, and business aspects of game development (Hakonen et al., 2008). It suggests that startups initially focus on attracting a large number of continuously playing users rather than monetizing, as noted in existing studies (Bergvall-Kåreborn and Howcroft, 2013b; Hamari, 2015). However, startups should already decide about their monetization mechanisms while developing their game concept to avoid extra work and cost in the future.
The findings show that game startups should pay attention to the business-related issues of game development and find a balance between the build quality and the market appeal quality, as already discussed by Qiu et al. (2017). It is attractive to have an innovative concept in the game, but startups should avoid creating innovations that may hinder ease of use. The notion of ease of use is about both the simplicity of the game control and the context of mobile game usage. The startups should consider the portability of mobile phones and, by designing short game sessions, allow for interruptions in the game usage in case of need. In their game build, they should not only consider the technical reliability, but also pay attention to designing in a way to allow the future improvement and enrichment of the game. Examples include using cross-platform game engines such as Unity for easy porting to multiple platforms, as well as using less text and more visualization to attract lower localization expenses.

Further, considering the humanistic aspects and the effect of social attributes on game-playing, the startups should include social aspects inside their games. The possibility to create interaction among users as well as designing connections to social media and encouraging users to spread the word in fun ways are among the low-cost tools for attracting more users to the game and keeping them engaged.

Startups should concern themselves with the unpredictability and heterogeneous demands of game players (Koskenvuoima and Mäntymäki, 2015; Yin et al., 2014). They should monitor the game performance and adapt fast. Confirming the findings of Spiegel et al. (2016) and Ojala (2015), the business model of the startups is not defined in a concrete way, but rather in a flexible form to allow adaptation to the market conditions. In fact, the business model of game startups should focus on the notion of ‘adaptability.’ Mobile Game startups should be alert to the fact that collecting performance data and users’ reviews and data-led maneuvering is a critical part of their business model. The available free or low-cost analytics tools for monitoring the game performance can work adequately to serve the purpose of monitoring and adapting.

The monitoring does not solely apply to improving the game and enriching its content but guides the startups in their market investments and strategies. Given the positive effect of localization on receiving users’ ratings, the game startups should have a strategy for localizing their games, given their limited resources. The experimentation approach suggests initially offering the game in English or with a few localization languages for major game markets, and then finding high-potential markets by monitoring the game analytics and localizing for them.

Significantly, startups should harmonize their localization with their game visibility. The localization should be a combination of translation and the use of local promoters and marketing activities. Startups should leverage all the available tools and channels for creating word-of-mouth and publicizing their game. The range of marketing efforts is quite broad, from running campaigns on social media to contacting local bloggers and reviewers. For markets with
special cultural and regulatory attributes (such as China), the startups may want to use publishers who have good knowledge of the markets.

While this study has shown that startups took a global approach in launching the game, developers should note that they must offer a completely playable and flawless game in terms of the technical build in the first place. Therefore, if they do not execute large-scale beta-testing during the development period, developers may try to soft-launch the game in a less critical market on the very first attempt and monitor the game quality. Thus, they can raise the quality effectively without jeopardizing their global launch and critically significant markets. In the case of this study, the initial quality was already approved by the mentoring and quality assurance process of the accelerator.

**Platform Owners**
The findings of this study show that early-stage startups suffer from various issues related to scarcity of resources and lack of knowledge about their business context. Since the value of the platforms is defined by a combination of the hardware devices and the available apps (Boudreau, 2012; Yoo et al., 2010), platform owners should take a responsible approach in supporting startups by creating a more desirable business context for them.

As noted by Bergvall-Kåreborn and Howcroft (2014), the power asymmetry between the developers and the platform owner is a source of challenges for the developers. When the startups are not fully aware of the platform’s future strategies, they will need to spend extra resources in order to adapt to the platform changes. Given the common problem of resource scarcity in startups (Giardino et al., 2016; Giardino et al., 2014; Paternoster et al., 2014), handling the additional costs can be a serious problem for the startups. In this context, platform owners can set boundary resources as effective means to help the startups harmonize their activities with the platform’s evolution. Surprisingly, the role of boundary resources so far has been limited to providing technical development environments or regulating the platforms (Ghazawneh and Henriksson, 2013). However, much more benefit can be gained through boundary resources, especially social ones, to assist developers in learning the know-how of creating a sustainable business.

Furthermore, the findings of this study can benefit any future platforms coming into existence. While developers face a lower level of competition in new platforms compared with mature platforms, they may also face lower possibilities for monetization due to the unpopularity of the new platforms. For example, in this study, the developers who used the in-app advertisement model of monetizing were unhappy with the small number of advertisers on the platform. To create competitive platforms, platform owners should consider incentivizing various groups of ecosystem members to join the platform. Meanwhile, they should consider that multi-homing is a common approach for developers due to the ease of porting. Therefore, it is through appropriate incentives and monetization opportunities that they can attract the developers to stay and continue working on a specific platform.
Corporate Accelerators

The findings can benefit corporate accelerators by shedding light on the interaction of the startups with the accelerator. The context of this study was a corporate accelerator for mobile apps with the aim to create an ecosystem for the platform (Pauwels et al., 2016). An accelerator with this bridging role should perform well through the timely delivery of knowledge and information between the platform owner and the developers.

The accelerator in this study was trying to assist the startups in raising their game quality as well as their game visibility. Although this is very helpful, the accelerator’s staff should consider the heterogeneous nature of the startups’ games and objectives, and design the mentorship and support program accordingly. They should be cautious in that not all startups aim for a large number of downloads, and some may want to target a niche market. Therefore, according to the selected games and startups’ objectives, the accelerators should hire the right mentors and experts who can guide the startups with their specific needs rather than a generalized approach.

Moreover, the timing of the assistance is important for the startups. While external funding provides critical aid to startups (Vanhala and Saarikallio, 2015), the allocation of the funding should be done at an appropriate time that allows the startups to hire the right expertise when they need it. This may provide complications for the accelerator in terms of bureaucracy and risks, yet if managed, can create much value for the startups by helping them to work effectively and efficiently from the beginning. In addition, the training should be provided to the startups at earlier stages of development and before they release their games. Then, the startups can already be equipped with business knowledge related to monetization mechanisms, user ratings, and use of analytics, and can apply them appropriately in their game design. This would provide the startups with the appropriate business insight for launching their games and taking immediate action for market adaptation.

7.3 Limitations and Future Research

This study has some limitations and provides avenues for future research. The key contribution of the study is the theoretical model of the ‘dynamics of success for early-stage mobile game startups.’ Obviously, the theoretical model is developed through empirical data collected from early-stage mobile game developers. Therefore, the findings may not be generalizable to other contexts with other types of app startups and/or to more experienced firms with a market reputation and more resources. However, the theoretical model has a modifiable nature when new data appears (Glaser, 1978, pp. 4-6). For future research, it would be interesting to see if the early-stage developers of non-game freemium apps follow the same dynamics for success, and, moreover, if and how the model would change for more mature firms.

In addition, the findings of this study indicate the activities of startups on a new platform. The new platform differs in terms of the level of competition by being less crowded, which can affect the strategies and exploitation of opportunities. Although the Windows Phone platform has practically stopped work-
ing, the findings of this study can still benefit mobile game developers, platform owners and accelerators in understanding early-stage developers’ activities. They can also guide future platforms that want to enter the smartphone ecosystem. Future research could investigate whether the theoretical model of the ‘dynamics of success for early-stage mobile game startups’ can be used by startups on more competitive platforms such as Apple and Google, or how it can be modified to fit the business context of these platforms.

Further, this study considers the startups in a platform-owned accelerator. It should be noted that AppCampus has been the only accelerator owned by a mobile platform to date of writing this dissertation. However, the findings of the study can still be useful for independent accelerators in terms of assisting startups with quality and visibility issues. While independent accelerators may not directly affect platform decision-makers, they can still assist startups to harmonize their strategies with the market actors working on visibility. An avenue for future research is to consider independent accelerators working in the context of mobile apps to see if and how they help startups with the same attributes of quality and visibility.
References


References


Ram, A. (2018). Rovio Shares Drop 50% as “Angry Birds” Maker Falls to Earth. URL: https://www.ft.com/content/86e7f392-17b4-11e8-9e9c-25c814761640/ (visited on 23/02/2018)


A decade has passed since the birth of the first app store in 2008 as the trigger for the evolution of the app economy. Since then, millions of apps have been created and their importance has spread through the daily lives of citizens and businesses, from providing entertainment to solving real problems. Many app developer startups have joined the game with the aim of creating a sustainable business. Coping with the shortage of resources and business experience and being compelled to operate within the limitations of the app stores’ structures, these startups strive to succeed in a fiercely competitive market. The question is, 'How do they do that'? This dissertation provides an answer by explaining how mobile game startups work to succeed in their first commercial game and treat it as a stepping stone to future growth and success.