Developer strategies in smartphone application markets

Dmitry Sharapov

Imperial College Business School

Abstract

This exploratory study uses interviews with 29 developer teams conducted during the development process of a focal mobile application to identify four strategies forming the basis of their app's intended competitive advantage. I then use objective data from secondary sources to examine which combinations of these strategies, along with the resources and capabilities of the developer team, led to timely app release, high app ratings and high app downloads. Preliminary results suggest that the emphasis placed by many developer teams on complementary assets as a foundation for success is misplaced. While owned complementary assets are associated with timely release, accessed complementary assets are associated with major delays in release or even outright failure. Neither owned nor accessed complementary assets appear to be associated with high rating or high downloads.

Introduction

A large and increasing share of global economic activity takes place in organizational ecosystems based around platforms through which distinct groups conduct transactions (Eisenmann, 2007). Examples of such platforms include newspapers, through which advertisers reach readers, and mobile device application markets, such as the Apple App Store, from which device users download third-party applications.

Both platform-leading firms and others operating in an organizational ecosystem based around a platform face distinctive management challenges.

The performance of firms taking a leading role in an ecosystem depends not only on how they deal with internal innovation challenges, but also on the challenges facing their suppliers and firms that provide complementary products (complementors), with the challenges facing these two groups having different effects on the performance of the focal firm. While much attention has been paid to approaches for co-operating with suppliers to overcome such challenges, less is known about how firms can effectively assist innovation efforts on the complementor side (Adner & Kapoor, 2010). Ecosystems leaders must also consider direct and indirect network effects when developing strategies to attract distinct groups of users to the ecosystem, and must be wary of threats of envelopment by other platforms (Eisenmann, Parker, & Van Alstyne, 2011). Additionally, platform leader governance choices and strategies towards competition and co-operation with different complementors have wide-ranging effects on innovation, and on the creation and capture of value within the ecosystem (Gawer & Cusumano, 2002; Iansiti & Levein, 2004; Casadesus-Masanell & Yoffie, 2007).

Complementors, on the other hand, must make choices regarding which platform(s) to make products for, taking into account a number of considerations, including platform popularity, competition, and technology (Venkatraman & Lee, 2004). They must also plan for the possibility that the platform leader might choose to enter into their product space should their offering be successful (Gawer & Cusumano, 2002; Iansiti & Levein, 2004).

The existing literature has significantly advanced our understanding of competition and innovation in platform-based markets. However, the focus of extant research has been mostly on platform leader strategies, with less attention devoted to the challenges facing complementors. While some recent work has improved our understanding of complementor strategies using data on the performance of applications in the Apple's smartphone ecosystem (Yin et al., 2014), what we know about complementor strategies comes largely from single

case studies, which provide detailed and highly contextualized evidence which is, however, limited in its empirical generalizability. As complementor input is crucial for ecosystem success, an improved understanding of complementor strategy in such contexts would add to academic knowledge about ecosystem competition, while also being highly relevant for practice.

This report analyzes 29 cases of apps created by smartphone application developers who were selected to receive grants and training from AppCampus, an accelerator funded by Microsoft and Nokia and managed by Aalto University in Finland. In the course of a number of visits to residential training camps run by the accelerator for selected teams from its portfolio, I conducted face-to-face interviews with the developers while development of the focal app was still ongoing, collected information from the accelerator's database containing records of the app's submission and development process to supplement and triangulate my understanding of these cases, and gained access to rich, objective measures of the performance of the apps post-release. In order to understand the strategies pursued by these complementors and to evaluate their effectiveness on app performance, I use fuzzy-set qualitative comparative analysis (fsQCA) methods (Ragin, 2008; Fiss, 2011).

The results show that a number of configurations of developer resources and capabilities and strategies result in timely app release, high app ratings, and high app downloads.

Data and method

Research setting

AppCampus was a three-year accelerator created in May 2012 through a collaboration between Microsoft, Nokia, and Aalto University in Finland. While Microsoft and Nokia each provided €9 million in funding, Aalto University agreed to manage the accelerator and to

cover its operating costs. The purpose of AppCampus was to attract novel, high quality apps to the Windows Phone platform by offering grants for app development in exchange for a temporary exclusivity agreement (the duration of the exclusivity period was reduced from 90 to 30 days towards the end of AppCampus' first year of operation). By the end of the programme in May 2015, AppCampus had received submissions of over 4,300 app ideas from developer teams based in over 100 countries and had invested approximately €10 million in over 300 entrepreneurial teams through grants of €20,000, €50,000 or €70,000.

The selection process for submitted ideas was rigorous, with only submissions that were judged to have the potential to meet novelty and quality standards chosen for funding. The acceptance rate of less than 10% illustrates this selectivity. In the second year of its operation, AppCampus started to offer residential training camps to selected teams from its portfolio. These training camps consisted of a combination of lectures, coaching sessions and networking events, covering a variety of topics related to mobile application development and marketing. The teams interviewed for this research took part in one of three two-week training camps that took place in April 2014, September 2014 and January 2015. The cases considered in this analysis are therefore not representative of the population of smartphone applications. With hundreds of thousands of applications populating mobile app stores and very few of these achieving significant downloads or revenues, I chose to focus on app ideas that should have a better than average chance at being successful, those that had passed the AppCampus selection process and those whose developers had also received two weeks training from AppCampus.[1]

Data

[1] On average, AppCampus-funded apps were of significantly higher quality than the Windows Phone store average, getting seven times more downloads, twice the revenue, and higher user ratings.

During these three training camps, I was present at the AppCampus office and approached all participating developer teams for interviews. This resulted in 46 interviews, excluding three developers who were participating for only part of their training camp. The interviews were semi-structured, covering a range of topics including the development team's composition and experience, and their plans for the AppCampus-funded app that they were developing at the time. The interviews were conducted with one or two members of the development team, their length ranged from 17 to 78 minutes, and all interviews were recorded and transcribed. Of the 46 cases, 17 had already released their app by the time of the interview. These cases were excluded from the analysis due to concerns about the initial performance of their apps impacting on their ability to recall the development process and strategy of their app in an unbiased manner, leaving 29 cases.

The information collected in the course of the interviews was later supplemented and triangulated using records regarding each team, their app, and the AppCampus selection and quality assurance process from the AppCampus customer relationship management (CRM) system. This included information about submitted by the developers in the course of applying for a grant from AppCampus, covering the composition and track record of the development team and a description of the proposed app and how it differs from and/or improves on apps already existing in the market, and notes made by AppCampus staff while evaluating the submission and following later communication with teams selected for grants.

Finally, I gained access to longitudinal, objective data on the performance of AppCampus apps that were released on the Windows Phone store to evaluate the performance of the released apps and to further triangulate some aspects of the cases.

Method

Combining evidence from interviews, the AppCampus CRM system, and Windows Phone store performance data, I built 29 case histories capturing aspects of the app that the developer team highlighted as giving it an advantage in comparison to other apps on the Windows Phone store, developer team characteristics, the type of app the team were developing, their plans for the app's release, and the eventual performance of the app.

To identify the conditions associated with good and poor app performance, I used fuzzy-set qualitative comparative analysis (fsQCA). fsQCA uses set theory and Boolean algebra to map configurations of causal conditions to an outcome which takes the form of a measure of case membership in an outcome set. Inferences from fsQCA are made on the basis of subset relations between configurations of causal conditions and an outcome rather than on the basis of correlations (Ragin, 2008). The goal of fsQCA is therefore the identification of (combinations of) causal conditions which are necessary and/or sufficient for an outcome. fsQCA allows the researcher to test hypotheses regarding relationships between different configurations of causal conditions and an outcome and also provides a means of testing whether causal conditions are causally core or causally peripheral for the outcome (Fiss, 2011). This method is also suitable for testing hypotheses of equifinality and causal asymmetry, which occurs when the absence of a condition associated with a high outcome does not result in a low outcome (Fiss, 2007). While fsQCA can be fruitfully used following a deductive approach in the analysis of a large number of cases (e.g., Fiss, 2011), it is well-suited for elaborating theory using data on a smaller number of cases (Redding & Viterna, 1999).

fsQCA allows for partial set membership, meaning that instead of a case being fully in or fully out of the set of cases exhibiting a particular explanatory condition (having a set membership score of 1 or 0), its degree of set membership can be calibrated to represent meaningful groupings (Ragin, 2008). The explanatory conditions examined in this report are

calibrated to have set membership scores of 0, 0.33, 0.66, and 1, where 0 represents nonmembership, 0.33 implies that a case is more out of the set than in, 0.66 implies that a case is more in the set than out of it, and 1 represents full membership in the set.

Set membership of cases in two of the outcome sets was calibrated following the direct method of calibration (Ragin, 2008) applied to objective Windows Phone store data. This method of calibration requires the researcher to specify three qualitative anchors used to rescale the quantitative data into set membership scores. The first anchor is the value of the variable in question corresponding to full set membership, with cases exhibiting this outcome given a set membership score of 0.95. The second anchor corresponds to the point at which a case is considered to cross over from being more out of the set than in it, to more in the set than out of it. Cases exhibiting this outcome are assigned a membership score of 0.5. The last anchor is the threshold at which a case is considered to be fully out of the set, with cases exhibiting this outcome assigned a membership score of 0.05. My calibration of the outcome and explanatory conditions was based on my in-depth knowledge of the cases and their context (Rihoux and Ragin, 2008), and is discussed below.

Application performance

Timely release. A basic measure of the performance of a project is whether or not the development process resulted in the intended output being finished. Furthermore, the timeliness of delivery, i.e., how close to its estimated completion date the project's output was finished, is frequently taken into consideration in project evaluation. In the context of this study, a timely release means that the app in question was released on the Windows Phone Store by the target release month given by the developer team during my interview with them. By the time of the interview, the team had already had multiple interactions with the quality assurance team of AppCampus and was aware of the quality standards required

for the final version of the app to be accepted by AppCampus as ready for release. I thus allocated full membership (1) to apps that were released before or during the developer team's target release month. Partial membership (0.66) was given to apps that were released between one and two months after the developer team's target release month. Apps that were released three or more months after the developer team's target release month were allocated a low degree of membership (0.33). Finally, in four of the cases considered in this study, the developer team did not release an app. These cases are not members of the set of cases with timely releases, and are given a set membership score of 0.

User ratings. An important measure of smartphone app quality is the average rating that the app receives from those who have downloaded it. Ratings range from 1 (low) to 5 (high). Following the criteria used by AppCampus staff to evaluate the success of the apps in their portfolio, the raw average ratings were calibrated using the direct method using the following thresholds: an average rating of 4.25 from ten or more reviews was taken as the threshold for full set membership (0.95); an average rating of 4.1 from ten or more reviews was takes as the crossover threshold (0.5); and an average rating of 3.8 or below, or any average rating given by fewer than 10 reviewers, was taken to be the threshold for non-membership (0.05). As four of the cases did not result in released apps, the analysis of the explanatory conditions associated with high user ratings is performed on 25 cases.

Downloads. The number of downloads received by an app is a key measure of its success, and is usually highly correlated with the revenues that the app generates for its developer team. Following the criteria used by AppCampus staff to evaluate the success of the apps in their portfolio, the raw average weekly download numbers for each app were calibrated using the direct method using the following thresholds: average weekly downloads of 5,750 were taken as the threshold for full set membership (0.95); average weekly downloads of 1,000 were taken as the crossover point (0.50); and average weekly downloads

of 500 were taken as the threshold for nonmembership (0.05). As 3 of the cases considered in this study released apps during the last week of the AppCampus programme, after which the Windows Phone store performance data of AppCampus apps was no longer tracked, I perform the analysis of explanatory conditions leading to downloads for 22 cases.

While I have access to objective, longitudinal data on the revenue generated by the apps considered in this study through premium downloads and in-app purchases, 9 of the 22 cases for which I analyze performance in terms of downloads chose to offer their apps for free and without any in-app purchases. As I do not have data regarding any advertising revenue generated by the apps, this leaves only 13 cases for which an analysis of explanatory conditions associated with high revenues can be performed. This small number of cases severely restricts the number of explanatory conditions that can be considered without running the risk of over-interpreting limited data (Marx, 2010), so an analysis of this outcome is left for a subsequent study.

Explanatory conditions

Analysis of my interviews with the developer teams and of the AppCampus CRM records revealed four recurring bases of competitive advantage that the developer teams believed their apps to have. While none of the cases were full members of all four sets, most combined at least partial membership in two or more sets.

Competing on novelty. A number of developer teams were developing apps that they considered to be highly novel in terms of a similar app not being available on any smartphone application market. For example, a developer stated of their app: “I think it should succeed because it’s different. I think it’s quite unique.” Another developer team said: “To my knowledge till now, only we are doing this for [our application category].” Talking of how the idea for their app came about, a developer team said: “We actually realized that there’s this new market that is still unexplored”. A further case of an app competing on novelty saw

the developer team applying a patent-pending approach to pricing in their business-to-consumer platform application. All such cases were considered to be fully in the set of cases competing on novelty (1). A number of cases had developer teams working on apps that added novel features to or targeted somewhat different target groups from existing competitors. Such cases were considered to be partially in the set of cases competing on novelty and were assigned a set membership score of 0.66. Cases in which the proposed app had a minor degree of novelty that was not much emphasized by the developer team or by AppCampus staff were assigned a set membership score of 0.33. Finally, cases in which the app under development was never described as being novel were considered to be fully out of the set of cases competing on novelty (0).

Competing on owned complementary assets. Some of the developer teams interviewed believed that owned complementary assets would give their app an advantage in the Windows Phone store. For some of the teams the complementary asset in question was an existing web-based version of the utility app that they were developing for mobile, while for others it was an existing prequel of the game sequel that they were developing, or a version of the game that was already released on another platform. For others, the complementary asset in question was a proprietary software engine or algorithm, or an established community of users of their previous apps. Cases in which developer teams heavily emphasized substantial complementary assets as a factor giving their app a competitive advantage were considered to be fully in the set (1), partial set membership (0.66) was assigned to cases with teams possessing some complementary assets but not putting much emphasis on their importance, a low degree of membership (0.33) was allocated to cases in which the development team possessed minor complementary assets, while nonmembership (0) was given to cases with no owned complementary assets.

Competing on accessed complementary assets. Complementary assets that the developer team did not own but believed it had negotiated access to formed the basis of case for their app's prospective competitive advantage in a number of cases. The complementary assets in question were often intellectual property owned by third parties, while a number of teams also highlighted their expectation of receiving marketing support from third party channels, usually based on the team's positive prior interactions with them. Cases in which accessed complementary assets were considered to be crucial for the app's success by the developer team were assigned full set membership (1), those in which some accessed complementary assets were mentioned as likely to have a positive impact were considered to be partially in the set (0.66), a low degree of set membership (0.33) was assigned to cases in which access to some complementary assets was mentioned in passing, while cases in which no mention of accessed complementary assets was made were considered to be fully out of the set (0).

Competing on execution. Being able to provide a better user experience than rival apps was mentioned as a basis for their app's competitive advantage by a number of developer teams. For game cases, this was often about having high production values, for example: "The graphics are very hand-crafted and the music is very hand-crafted as well." Another developer team described the standout features of their game as: "Nice artwork throughout it and nice music and sounds, which is what we've all focused on as our strong points within the company". For utility apps, it was about providing an easier and/or more reliable way of doing things compared to existing competitors. For example, one developer team stated: "We studied the market that is there. Competitors? Yes, there were some, but they weren't good enough." Another developer team described their app's competitors as "second rate products" before continuing: "We are doing it right. We have the best; it's easy to use, it works as you expect." These and other cases in which developer teams placed major

emphasis on execution as being the basis for their app's competitive advantage were considered to be fully in the set (1), those mentioning some aspects of execution as positive for the app's competitive performance were assigned a partial set membership score (0.66), cases in which the developer team made only a passing reference to execution were considered to be more out of the set than in (0.33), while cases in which no mention of aspects of the app related to execution of the concept was made were considered to be out of the set (0).

A second set of explanatory conditions concerns the resources and capabilities of the development teams working on the apps. These explanatory conditions include resources allocated to app development, team mobile development experience and team entrepreneurial experience.

Resources allocated to app development. In the course of my interviews with the developer teams, I was able to get a good understanding of both the duration of the app's development to date as well as the number of full-time and part-time people involved in its development over that period. Taking the app's Windows Phone release date (or date of exclusion from AppCampus in the case of unreleased apps) as the development end-point and counting part-time people as working half the amount that a full-time team member would work in a given month of development, cases in which teams spent more than 60 man-months on developing their app or outsourced their app's development to an external team were considered to be fully in the set (1), while those who spent between 40 and 60 man-months on development were assigned partial set membership (0.66). Teams spending between 30 and 39 man-months on development were considered to be mostly out of the set of cases with large resources allocated to app development (0.33), while teams who received only the minimum grant from AppCampus (€20,000) and who spent fewer than 30 man-months to develop, were considered to be completely out of the set (0).

Team mobile development experience. During my interviews with the developer teams, I gained a good understanding of the degree to which the team members had a background in and experience of developing smartphone apps. Teams with developers having many years of mobile development experience and a large number of previously released apps were considered to be fully in the set (1), those with several prior app releases and some years of mobile development experience were assigned partial set membership (0.66), cases with developer teams who had only limited mobile development experience, releasing one or two previous apps at most, were considered to be mostly out of the set (0.33), while teams for whom their AppCampus app would be their first mobile app were considered to be fully out of the set (0).

Team entrepreneurial experience. Alongside mobile development experience, entrepreneurial experience may also contribute to good app performance. Cases in which the team had successfully operated as a standalone business for a large period of time prior to developing the focal app and those in which at least one of the full-time team members was a serial entrepreneur were considered to be fully in the set (1), cases with team members that had a number of years of entrepreneurial experience in the focal or another company were assigned partial set membership (0.66), those who had operated as a standalone enterprises for only a short period of time or who had operated as a one-person business for a number of years were considered to be mostly out of the set (0.33), while those who only founded their first company after being selected by AppCampus were considered to be fully out of the set (0).

Finally, I consider two explanatory conditions relating to the characteristics of the market being addressed by the app: whether or not the app in question is a game, and the degree to which it targets the global Windows Phone market, as opposed to targeting a narrower geographic region.

Game. The important distinctions between games and other apps are well accepted by both practitioners and academics (e.g., Yin et al., 2014). Cases in which the app under development is a game serving no other purpose than entertainment are considered to be fully in the set (1). In a number of cases the app under development was primarily focused on the gaming side, but also contained important educational or utility elements. Such cases were considered to have partial membership (0.66). Some cases used in-app gamification mechanisms in the hope of increasing user engagement with the utility app that was the focus of development. These cases we considered to be more out of than in the set of games (0.33). Pure utility apps were considered to be completely out of the set (0).

Targeting global market. The size of the market onto which a product is being released can be seen as an upper bound on the number of users it can hope to attract. In addition to deciding whether or not an app should be released on all national Windows Phone stores worldwide, developers can also choose to localize their app for any number of languages. Localization often goes beyond simple translation of in-app text to include special features or notifications customized for each localized country. Cases in which the app was released globally and was localized for more than three languages were considered to be fully in the set (1), apps that were released globally and localized for two to three languages were assigned partial set membership (0.66), cases in which the app was released globally but only in one language were considered to be more out of the set than in (0.33), while cases in which the app was released in only a narrow geographical area were taken to be completely out of the set (0).

Analysis

Following calibration of set membership scores, the next step in performing fsQCA is to construct a truth table, which lists all possible configurations of explanatory conditions. Any

configurations not associated with any of the cases analyzed were deleted. To identify configurations consistently associated with the outcome of interest, I then specified a consistency threshold. Consistency measures the extent to which cases exhibiting a combination of explanatory conditions also exhibit the outcome (Ragin, 2008). Following common practice, a consistency threshold of 0.8 was chosen.

The truth table algorithm (Ragin, 2008) was then employed to logically reduce the observed combinations of explanatory conditions and outcomes to a parsimonious and intermediate solution using fs/QCA 2.5 software (Ragin et al., 2006). The parsimonious solution uses only the simplifying assumption arising from the combinations of explanatory conditions observed in the cases, and contains only the core explanatory conditions that have the strongest evidence for being associated with the outcome. The intermediate solution, which contains and extends the parsimonious solution, employs additional counterfactual analysis to use the researcher's hypotheses regarding the effect of the presence or absence of explanatory conditions on the outcome to hypothesize the degree to which some hypothetical unobserved cases would be associated with the outcome, thus allowing the researcher to identify explanatory conditions that have some evidence for being linked to the outcome, but which are more peripheral than the core explanatory conditions (Ragin, 2008; Fiss, 2011).

Results

Timely release

The explanatory conditions considered for explaining timely release include those listed in the previous section, with the exception of *targeting global market*, which relates to the size of market being addressed and which should have little impact on the developer team's ability to produce software of the technical standard required for an app to pass AppCampus quality control.

Table 1 shows the configurations of explanatory conditions associated with timely release. There are 6 parsimonious and 12 intermediate solutions. The assumptions used for the counterfactual analysis to arrive at the intermediate solution were that the presence of *resources allocated to app development, team mobile development experience, team entrepreneurial experience, competing on owned complementary assets* and the absence of *competing on accessed complementary assets* would be associated with timely release.

Solutions 1 and 2 suggest that a lack of entrepreneurial experience need not prevent timely release if it is compensated for by either owned complementary assets or by having a large amount of resources allocated to app development. Solutions 3 and 4 suggest game development aiming to compete on novelty and utility app development aiming to compete on owned complementary assets are both associated with timely release. A further path to timely release appears to be mobile development experience without reliance on either execution or accessed complementary assets as the bases of competition, as seen in solution 5. Solution 6 suggests that having a large amount of resources allocated to development in the absence of either owned or accessed complementary assets is a further path to timely release. Interestingly, it appears that absence of reliance on accessed complementary assets is an explanatory condition in all but one of the solutions, suggesting that developers may underestimate the difficulties of gaining access to complementary assets owned by other parties during their development process. The solution is highly consistent (0.93) and accounts for three quarters (0.75) of cases that resulted in timely releases.

Table 2 shows the configurations of explanatory conditions associated with the absence of a timely release. The assumptions used for the counterfactual analysis in this case are the mirror image of those described above: it was assumed that the absence of *resources allocated to app development, team mobile development experience, team entrepreneurial*

experience, *competing on owned complementary assets* and the presence of *competing on accessed complementary assets* would be associated with an absence of timely release.

The analysis produced three solutions. The first of these suggests that entrepreneurially experienced teams working on utility apps without a focus on execution are less likely to produce a timely release. A lack of resources combined with limited development experience, lack of owned complementary assets and a utility app was also associated with the absence of a timely release, as illustrated in solution 2. Solution 3 suggests that even cases with developer teams experienced in both mobile development and entrepreneurship may not lead to timely release if they are working on a utility app without a focus on novelty, and with limited resources. Overall, it appears that a combination of lack of resources and the development of a utility app is associated with failure to deliver a timely release. However, the consistency (0.77) and coverage (0.33) of this solution are much lower than those of the solutions associated with timely release, suggesting that other factors that are not taken into account in this study are likely to be in play.

User ratings

The explanatory conditions considered in the analysis of the achievement of high user ratings were the same as those used in the analysis above. The assumptions used for counterfactual analysis were that the presence of *resources allocated to development*, *team mobile development experience*, and *team entrepreneurial experience* would be associated with high ratings, and, conversely, that the absence of these conditions would be associated with the absence of high ratings.

The first six columns of Table 3 show the solutions leading to high user ratings, while the last three columns contains the solutions associated with low user ratings. Solution 1 suggests that high ratings can be achieved by a utility app despite a lack of team mobile development experience if the app is not competing on novelty, and if the lack of team

mobile development experience is compensated for by either resources of team entrepreneurial experience. Apps produced by well-resourced and/or experienced teams can achieve high ratings, despite the absence of a focus on complementary assets or execution as bases of competitive advantage, as shown by solution 2. Solution 3 suggests that a focus on both novelty and execution are associated with achieving high ratings for games. The overall consistency and coverage of this solution are high, at 0.91 and 0.63, respectively.

Regarding configurations of explanatory conditions associated with low ratings, solution1 suggests that resources allocated to app development in combination with team mobile development experience may not be enough to achieve high ratings if the team lacks entrepreneurial experience. Solutions 2 and 3 suggest that utility apps focused on execution may not achieve high ratings if resources or development experience are lacking. These solutions are highly consistent (0.97) but only account for a minority of pathways to poor ratings, as illustrated by their relatively low coverage (0.32), again suggesting that other, unaccounted for factors, may be in play.

Downloads

In investigating the factors leading to high weekly downloads, alongside the explanatory conditions relating to the intended bases of the apps competitive advantage, I considered the app's average rating, its membership in the set of games, and its membership in the set of apps targeting a global market. For the counterfactual analyses, the presence of high *user ratings* and *targeting global market* were assumed to relate to membership in the set of apps with high *downloads*, while the absence of these explanatory conditions was assumed to relate to membership in the set of apps with low *downloads*.

The configurations of explanatory conditions associated with high downloads are presented in the first three columns of table 4. The first solution suggests that games that are highly rated are likely to be in the set of highly downloaded apps. Solution 2, on the other

hand, suggests that the combination of competing on novelty, competing on execution, and high user ratings are associated with high downloads for all apps, regardless of their membership in the set of games. The overall solution is highly consistent (0.95) and accounts for just over half of highly-downloaded cases, with coverage of 0.56.

The last six columns of table 4 show the configurations of explanatory conditions associated with low downloads. Solution 1 suggests that regardless of other explanatory factors, poorly-rated apps are unlikely to end up in the set of apps that are highly-downloaded. Solution 2 suggest that apps targeting a narrow geographical area without a focus on execution are also unlikely to receive many downloads. The consistency and coverage of these solutions are 0.97 and 0.65, respectively.

Conclusion

The analyses reported above lead to a number of tentative conclusions about developer strategies in mobile application markets. First, the emphasis that many of the interviewed developers placed on complementary assets, whether owned or accessed, as a basis of competitive advantage for their app on the Windows Phone store appears to have been misplaced. While having owned complementary assets appeared to be a core explanatory condition in a number of configurations associated with timely release, accessed complementary assets appears to often be a hindrance. Neither owned nor accessed complementary assets were core explanatory conditions in configuration associated with either high user ratings or high downloads. In other words, while having owned complementary assets may indeed help in the development process, the developer teams studied here were perhaps guilty of over-estimating the value that app users would gain from them being part of the app.

Second, the results suggest that the frequently made distinction between games and utility apps is well-justified. Utility apps appear to be more likely to be in the set of apps not achieving timely release, poor ratings, and low downloads, while games that are novel and well-executed appear to perform well on all outcome measures, regardless of other explanatory factors.

Finally, it appears that the absence of one of the causal conditions relating to the resources and capabilities of the developer team, unless it compensated for by presence of one or more of the others or by owned complementary assets, is often associated with failure to achieve either timely release or high user ratings.

Smartphone application developers face many challenges in competing in a potentially very lucrative but highly competitive market. Future work should look to evaluate and further expand on these preliminary findings in order to improve our knowledge of the strategies that do and do not work in these fascinating and increasingly important competitive contexts.

References

Adner R & Kapoor R. 2010. Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. ***Strategic Management Journal***, 31: 306-333.

Boudreau KJ. 2012. Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation. ***Organization Science***, 23: 1409-1427.

Casadesus-Masanell R & Yoffie DB. 2007. Wintel: Cooperation and conflict. ***Management Science***, 53: 584-598.

Eisenmann T. 2007. *Managing networked businesses: Course overview for educators*. HBS note no. 807-104.

Eisenmann T, Parker G, & Van Alstyne M. 2011. Platform envelopment. ***Strategic Management Journal***, 32: 1270-1285.

Fiss PC. 2007. A set-theoretic approach to organizational configurations. ***Academy of Management Review***, 32(4): 1180-1198.

Fiss PC. 2011. Building better causal theories: A fuzzy set approach to typologies in organization research. ***Academy of Management Journal***, 54(2): 393-420.

Gawer A & Cusumano MA. 2002. *Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation*. Harvard Business School Press, Boston, MA.

Iansiti M & Levein R. 2004. *The keystone advantage: What the new dynamics of business ecosystems mean for strategy, innovation, and sustainability*. Harvard Business School Press, Boston, MA.

Marx A. 2010. Crisp-set qualitative comparative analysis (csQCA) and model specification: Benchmarks for future csQCA applications. ***International Journal of Multiple Research Approaches***, 4: 138-158.

Ragin CC, Drass KA, & Davey S. 2006. *Fuzzy-Set/Qualitative Comparative Analysis 2.0.* Department of Sociology, University of Arizona: Tucson, AZ.

Ragin CC. 2008. *Redesigning social inquiry: Fuzzy sets and beyond*. University of Chicago Press, Chicago, IL.

Redding K & Viterna JS. 1999. Political demands, political opportunities: Explaining the differential success of left-libertarian parties. ***Social Forces***, 78: 491-510.

Rihoux B & Ragin CC (eds.) 2008. *Configurational comparative methods. Qualitative Comparative Analysis (QCA) and related techniques.* Sage, Thousand Oaks and London.

Venkatraman N & Lee CH. 2004. Preferential linkage and network evolution: A conceptual model and empirical test in the U.S. video game sector. ***Academy of Management Journal***, 47: 876-892.

Yin P-L, Davis JP, & Muzyrya Y. 2014. Entrepreneurial innovation: Killer apps in the iPhone ecosystem. ***American Economic Review***, 104(5): 255-259.

Table 1: Configurations of explanatory conditions leading to timely release

	Solutions											
Causal conditions	1	2	3a	3b	4a	4b	4c	5a	5b	6a	6b	7
Resources		●		•			•			●	●	•
Dev. experience	•			•	⊗	⊗	⊗	●	●	•		
Ent. experience	⊗	⊗		•		•	•		⊗	•		
Novelty	⊗	⊗	●	●	•	⊗		⊗		⊗	⊗	•
Own comp. assets	●				●	●	●		•	⊗	⊗	
Accessed comp. assets		⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Execution	⊗	•	•		⊗	•	•	⊗	⊗		•	⊗
Game	•	⊗	●	●	⊗	⊗	⊗	⊗	⊗		⊗	⊗
Raw coverage	0.07	0.05	0.24	0.19	0.09	0.05	0.07	0.11	0.03	0.19	0.07	0.16
Unique coverage	0.03	0.02	0.12	0.05	0.02	0.02	0.03	0.03	0.00	0.05	0.00	0.03
Consistency	1.00	1.00	1.00	1.00	0.84	1.00	1.00	0.85	1.00	1.00	0.99	0.81
Number of cases	29											
Overall solution consistency	0.93											
Overall solution coverage	0.75											

Key: ● = core causal condition (present). • = peripheral causal condition (present). ⊗ = core causal condition (absent). ⊗ = peripheral causal condition (absent).

Table 2: Configurations of explanatory conditions not leading to timely release

	Solutions		
Causal conditions	1	2	3
Resources		⊗	⊗
Dev. experience		⊗	●
Ent. experience	●		●
Novelty	●	●	⊗
Own comp. assets		⊗	
Accessed comp. assets			
Execution	⊗	⊗	●
Game	⊗	⊗	⊗
Raw coverage	0.17	0.10	0.13
Unique coverage	0.10	0.10	0.07
Consistency	0.72	1.00	0.80
Number of cases	29		
Overall solution consistency	0.77		
Overall solution coverage	0.33		

Key: ● = core causal condition (present). ● = peripheral causal condition (present). ⊗ = core causal condition (absent). ⊗ = peripheral causal condition (absent).

Table 3: Configurations of explanatory conditions leading to high ratings and low ratings

	Solutions leading to high ratings								**Solutions leading to low ratings**		
Causal conditions	1a	1b	2a	2b	2c	3a	3b	3c	1	2	3
Resources	●			●	●		●		⬤		⊗
Dev. experience	⊗	⊗	●		●			●	⬤	⊗	⬤
Ent. experience		●			●			●	⊗		
Novelty	⊗	⊗	⊗	●		⬤	⬤	⬤	●	⬤	⊗
Own comp. assets	⊗	●	⊗	⊗	⊗	⊗			●	●	●
Accessed comp. assets	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Execution	●	●	⊗	⊗	⊗	⬤	⬤	⬤	⊗	⬤	⬤
Game	⊗	⊗	⊗	⊗		⬤	⬤	⬤	⊗	⊗	⊗
Raw coverage	0.07	0.05	0.10	0.07	0.20	0.22	0.17	0.24	0.08	0.08	0.16
Unique coverage	0.05	0.02	0.05	0.02	0.11	0.05	0.02	0.05	0.08	0.08	0.16
Consistency	1.00	1.00	1.00	1.00	0.85	0.90	0.88	0.91	1.00	1.00	0.94
Number of cases	22								22		
Overall solution consistency	0.91								0.97		
Overall solution coverage	0.63								0.32		

Key: ⬤ = core causal condition (present). ● = peripheral causal condition (present). ⊗ = core causal condition (absent). ⊗ = peripheral causal condition (absent).

Table 4: Configurations of explanatory conditions leading to high downloads and low downloads

	Solutions leading to high downloads			**Solutions leading to low downloads**					
Causal conditions	1	2a	2b	1a	1b	1c	1d	1e	2
Ratings	⬤	⬤	⬤	⊗	⊗	⊗	⊗	⊗	
Global release	●		●		⊗		⊗		⊗
Novelty		⬤	⬤	⊗	●	⊗		●	●
Own comp. assets	⊗		●	●	●	⊗	●	●	⊗
Accessed comp. assets	⊗	⊗	⊗	●	⊗	⊗	⊗	⊗	⊗
Execution		⬤	⬤		⊗	●	●	●	⊗
Game	⬤	●		●		●	⊗	⊗	⊗
Raw coverage	0.35	0.36	0.25	0.11	0.16	0.16	0.06	0.09	0.06
Unique coverage	0.15	0.08	0.05	0.11	0.16	0.16	0.06	0.09	0.06
Consistency	0.94	1.00	0.98	0.91	1.00	0.98	0.88	1.00	1.00
Number of cases	22			22					
Overall solution consistency	0.95			0.97					
Overall solution coverage	0.56			0.65					

Key: ⬤ = core causal condition (present). ● = peripheral causal condition (present). ⊗ = core causal condition (absent). ⊗ = peripheral causal condition (absent).