How to be a Successful App Developer: Lessons from the Simulation of an App Ecosystem

By Soo Ling Lim, Peter J. Bentley

Summarized by Vince Fasburg and Josh Thomas

1. INTRODUCTION

Mobile application development has exploded in recent years into an estimated $58 billion industry in 2014, an industry in which many application developers compete directly with one another for user downloads of their apps. This competition has created a handful of developers and companies earning massive profits, while many others are unable to ever get their apps off the ground. This paper proposes an agent-based evolutionary model for the "App Ecosystem" of the developers, the applications they develop, and the users that download the applications. The model seeks to evaluate which application development strategies will be most successful with consumers.

2. THE MODEL

2.1 Developers

The AppEco model defines five different types of developers that employ different strategies to develop Applications with different features. The strategy used by each developer effects the features that will included in their applications, as described in the next section. Each developer follows its defined strategy and releases a new app on a preset schedule of a fixed number of days which is different for each developer. It is also possible for a developer to become inactive based on a random probability of becoming inactive set for each developer. The five types of developers are:

1) The "Innovator" builds apps with random features each time.
2) The "Milker" makes variations of their own most recent app each time with random variation, thus "milking" the same idea repeatedly (their first app is generated randomly just as the innovator).
3) The "Optimizer" is similar to the milker, except that each successive app is a variation of their own best app (by number of downloads) instead of variations on their most recent app.
4) The "Copycat" copies a random app from the Top Apps chart (the most downloaded apps ever), with random variation.
5) The "Flexible" developer follows one of the previous four strategies initially, and after each app has a 1% probability of changing to a different development strategy.

2.2 Applications

Applications are modeled using a 10x10 matrix of features that any app can either offer or not offer. As such, the matrix filled with binary values representing which of 100 different features

are present in that application. A matrix layout is used to measure the similarity between features, where features that are closer together are more similar. The variations in features that occur based on each of the development strategies discussed in the previous section occur in the form of a 50% chance of replacing one filled cell of the feature matrix with another cell that was previously unfilled.

2.3 Users

In this model, users of applications browse the app store and download apps that they desire on a regular schedule, a set number of days that is different for each user. Each user has a fixed set of feature preferences, which is encoded as a 10x10 feature matrix just like that of each app. Each cell in this matrix represents a feature that each user desires or does not desire, based on whether or not the cell is filled in. However, no users will ever have any cells in the upper left 5x5 submatrix of their preference grids filled in. This represents features that no users desire, such as difficult to use or malicious applications. Every time a user browses the app store, they download every app which contains only features that that user desires.

2.4 Algorithm

Parameters of the algorithm such as initial populations and growth rates were tuned to match actual values of the iOS app ecosystem from the time period of Q4 2008 to Q3 2011. With these parameters set, the initialization of the users and developers is random except for the rules outlined in the previous sections. The model then proceeds in discrete timesteps representing days. Through each timestep, the population size of developer and user agents grows using a sigmoid growth function in which growth rates decrease as there are more individuals in the populations. Every timestep, some of the developers will release apps and some of the users will browse the app store and download some of the apps, and the populations of users and developers increase.

3. RESULTS

After the simulation was run, several “fitness” metrics were used to answer different questions about the results. By comparing *average downloads per app* that were developed using different strategies, the simulation determines that the Copycat strategy was the most effective for generating downloads, beating out the second place strategy by a ratio of almost 6 to 1. The Copycat strategy also topped the related *Top Total Downloads* category, which is the total number of downloads from any app produced by a strategy not controlling for the number of such apps produced.

Similarly, a metric called the *Feature Coefficient of Variation* was created to determine which app development strategy produces the greatest diversity of features between apps. Not surprisingly, the Innovator strategy leads this category by pumping out apps with randomly generated features.

Other insights that can be made using this data include which type of development strategy will be chosen by developers when all developers are flexible in their strategy? Interestingly, even

though the Copycat strategy seems to be the strategy that will generate the most downloads, this strategy quickly becomes very unpopular as the simulation progresses. This occurs despite parameter changes that were tried in an attempt to make the Copycat option seem more beneficial (postponing the time at which developers could change strategies so there would be very popular apps in the ecosystem to be copied). Different runs of the simulation produced varying results on the relative positions of the other 3 strategies, but Copycat was consistently the least popular.

4. CONCLUSION

AppEco is an agent-based simulation of the ecosystem of mobile app development and downloading. The simulation compares different development strategies and shows how these strategies interact to produce popular or unpopular applications. It seems that a Copycat strategy is the most lucrative in terms of downloads, but this strategy must only be used by a minority of developers so that there is a sufficient supply of good apps to be copied. Future research is planned to investigate the effects of publicity on app downloads, and explore how users might be able to better locate desirable apps and provide feedback of their preferences back to developers.