Aug 10th, 12:00 AM

Product Design And Success in A Platform Ecosystem

Sungyong Um
National University of Singapore, sungyong@nus.edu.sg

Gary Templeton
West Virginia, gary.templeton@mail.wvu.edu

Martin Kang
Mississippi State University, dk1029@msstate.edu

Product Design and Success in A Platform Ecosystem

Emergent Research Forum (ERF)

SungYong Um
National University of Singapore
sungyong@nus.edu.sg

Gary Templeton
West Virginia University
gary.templeton@mail.wvu.edu

DaeYoup Martin Kang
Mississippi State University
mkang@business.msstate.edu

Abstract

A platform ecosystem allows developers to leverage the software codebase innovations of others. Codebase (re)combination extends the range of opportunities for product innovation aimed at satisfying the functional needs of users. Despite facilitating extraordinary innovation gains among developers, prior research on how outside developers can more efficiently develop digital ecosystem resident products has not been addressed. We emphasize two layered properties of design features that may increase digital product innovation: 1) the dependencies of functions and 2) the dependencies of topical representation expressed in product descriptions. We analyze the source codes of R platform ecosystem packages using deep learning algorithms (i.e., Struc2Vec and Doc2Vec) to capture how layered software properties change as the dynamics of the R platform ecosystem increases in each month. We found a positive relationship between software codebase usage and topical representation and digital product design success.

Keywords

Guides, instructions, length, conference publications. platform ecosystem, product design, software components, deep learning algorithms

Introduction

A platform ecosystem has changed the competition of digital product design. Competitive pressures are more intense in a platform ecosystem, requiring far more immediate sourcing and shorter cycle times than traditional product lines (Alexy et al. 2018). Developers other than platform owners play crucial roles in triggering digital product innovation in a platform ecosystem (Boudreau 2012; Parker et al. 2017). Third-party developers attempt to fulfil the functional demands of users by leveraging collective resources (i.e., software codebases) and competencies to potentially spin off new product designs and niches. Hence, such features result in ever-increasing product design within a platform ecosystem. For example, as of October 2020, less than 200 applications in Google Play Store were created by Google while more than three million (e.g., Apple Music API and Uber Maps API) were created by third-party developers (www.statista.com).

Products are designed from the uses of various software codebases in a platform ecosystem (Parker et al. 2017). Software codebases are encapsulated to operationalize developers' ideas without developing all necessary functions in products (Tiwana 2018). Software codebases can be used with one another to build up the unique functions of products. Hence, some useful software codebases are commonly used in products. In addition, the shared uses of software codebases form interdependencies with other codebases to gain their functional benefits in product designs (Baldwin et al. 2014). Such interdependencies enables to form functional collaborations to enhance the usefulness of product that attract users. For example, in Google Play Store, some applications adopt Google Maps API to leverage its geographical information, and Google Maps API can be used to collaborate with other APIs providing various functions (e.g., weather and

photo) in applications. The key features of software codebases are explicated in product descriptions (e.g., description and tags) to promote functional usefulness. The descriptions attract diverse users by describing functional advantages relevant to users' needs.

The number of software components increases, as the size of a platform ecosystem increases (Fortuna et al. 2011). Product designs are diverse in a platform ecosystem (Boudreau 2012). Not all products gain the same competence when using similar software codebases. As such, the functional diversification and differentiation for attracting users are hard to achieve from selecting proper software codebases in product designs. However, some codebases having functional usefulness enable technological advances leading to the dependencies of product design more than others. Some design dependencies directly result from unique functions offered in new software codebases, leading developers to have more considerations in their product designs. Others continue until they are fundamentally replaced by the comprehensive advances of software codebases. For example, the release of Google Photos API providing convenience in photo sharing and editing in Google Play Store attracts third-party developers to replace the uses of existing photo APIs in their application design (Foerderer et al. 2018). Hence, capturing and reflecting dependencies in product designs is a critical innovation strategy toward increasing product competition (Pavlou and El Sawy 2006). In this conceptualization, changes in design dependencies derive from alternative technical options that guide to take the advantage of interdependencies in product designs while influencing the use frequencies of available software codebases (El Sawy et al. 2010; Sambamurthy et al. 2003).

Unfortunately, little is known about the changes of product designs, factors and relevant strategies in a platform ecosystem. One clear omission in the literature is the identification and evaluation of effective product innovation strategies that consider market-based feature. Scholars have argued that product designs can be enhanced through a platform ecosystem, where the uses of software codebases lead to form interdependencies among products. Such product innovation studies have been focused on how product designs can be strategized using internally sourced evidence. Hence, existing studies are still limited to understand important market-based design considerations critical to developer innovativeness and eventually their product success. To gain more comprehensive insights into successful innovation strategies, our study argues that incorporating the feature of product designs may be invaluable to product innovation success.

We fill this gap by defining and empirically testing product design variables drawn from an emerging ecosystem environment. These factors relate to fitting design choice with the platform ecosystem environment. Given this background, the purpose of our study is to answer the research question: How does market-based design affect the success of software products in a platform ecosystem? By addressing design features, we aim to understand the effects of having various design alternatives available on strategic digital product development in a platform ecosystem. We aim to illuminate for development project managers how functional preferences are relevant in platform-based product innovation. Further, we aim to explain how new software codebases can be exploited by developers to enhance digital product innovativeness.

Theoretical Foundation

To answer the research question, we devise theory about product design strategies. We investigate the dependencies of product design properties to fit users' preferences at two different levels. Those properties include: 1) the interdependencies of software codebases and 2) the interdependencies of topical representations (i.e., descriptions) spawned from the core functional features of software codebases. The topical representations capture the dependencies of product designs as functional benefits of software codebase uses highlighted in product descriptions by developers.

Dependencies represent unique product features selected based on the functional uniqueness and interoperability of codebases. Developers will use codebases if it offers functional usefulness to products (Haefliger et al. 2008). Hence, such codebases embody the outbound dependency of a focal product. For example, Uber App may function independently or be used as a codebase having an outbound dependency on OpenTable or Hyatt ride booking apps. Hence, the anticipated outbound dependencies of focal products affect their design.

The outbound dependencies represent changes in the functional competency of exogenous codebases, which in turn helps anticipate functional competence. The uniqueness of codebases may spill over into competitive advantages in the forms of usefulness or innovativeness (Parker et al. 2017). While not all

unique exogenous codebases are useful, users are drawn toward more innovative product designs. Second, dependencies change depend on the extent to which product designs incorporate useful codebases. However, being uniquely useful does not always translate to greater codebase dependencies because competition (e.g., Google Maps vs. Uber Maps) will also affect design choices. The digital ecosystem environment continuously motivates developers to create products in new functional categories or risk obsolescence (Foerderer et al. 2018). In platform ecosystems, existing competitors and new entrants threaten dominate when they are more innovative and do not rely on collaboration. Because the dependencies of codebases is so critical to designing ecosystem products for achieving these outcomes, we posit:

Hypothesis 1: The dependencies of codebases are positively associated with the success of products.

Codebase usage is specified in product descriptions to highlight functional user advantages of products (Gong et al. 2018). The topical representations influence the selection of software codebases in product design. This is possible since the functional features of codebases are expressed in product descriptions. Thus, themes observed as product similarities can be used to explore the effect of a given topical representations.

Thematic dynamics affect the ability of developers to use product competence as a competitive advantage. Developers have to determine product function fit and evaluate possible technological advances. Of particular importance in such design choices is the consideration of functional categories (e.g., travel or gaming), which we refer to as the topical representations. Topical representations reflect the popularity and retention of functional categories (e.g., network visualization) in products and consequently changes in emerging functional demand (e.g., the recent popularity of TensorFlow and PyTorch in machine learning algorithms). Topical representations also affect the extent of design choices during product creation in a platform ecosystem (Milgrom et al. 1991). By defining the range of functional options, topical representations help developers focus on functional categories of product designs.

The topical representations reflect temporal patterns in the functional needs and demands of users that should be addressed over time from project initiation through maintenance. The trends of established topical representations reflect new technological advances when the uses of codebases shift. An example of how the topical representations can be used is the monitoring of emerging functional categories in the R platform. The popularity of R software packages has migrated away from DNA genome sequencing techniques (e.g., genetic network algorithms) toward big data analytic tools (e.g., visualization and deep learning algorithms) over the last 10 years. The establishment of new genres is also influenced when developers strive to update functions to attract and satisfy users. The topical representations is an important design consideration because they lead developers to create more popular products. It reduces uncertainty by pursuing increased popularity (i.e., organizational competitiveness via a larger user base and greater sales). By contrast, developers with less understanding and experience concerning niche dependencies will have fewer chances to increase returns on their digital product design investments. Therefore, we hypothesize:

Hypothesis 2: The dependencies of codebases' topical representations is positively associated with the success of products.

Empirical Result

Our empirical setting is the R platform ecosystem, which is typically used by outside developers to create various freely available digital products (or packages) in the statistical and visualization niches. We collected source code and descriptions of all archived R packages spanning from inception in 1991 to December 2016. We used a proprietary text-mining technique to access cran.r-project.org to acquire usage data on 8,392 software packages created by 11,930 developers. The data included the function libraries created by developers associated with other platform codebases. In particular, to capture the dependencies of descriptions in products, we applied a neural network technique (i.e., Struc2Vec and Doc2Vec) for extracting the information of codebases and topical representations. Using the detailed product information, this is the first empirical study including the product designs at the two different levels as design decisions in this critical area of digital ecosystem research. Because of ever-increasing numbers of codebases, pressure mounts on developers to better redesign and update product functionalities.

We run an econometric model to test hypotheses. Our main objective is to confirm the significance of reflecting the features of codebases and topical representations on product designs. The model is built up at the product level by extracting all software package information from the R platform. We focus on how the codebases and topical representations affects the success of digital product innovation (Haefliger et al. 2008; Parket et al. 2017). Our response variable is to denote the log of Weekly downloads for product i at every two weeks t, ln(WD_{it}). The number of downloads by users measures the product popularity (i.e., more popular products have more sales in product competition). The main explanatory variables are as follows: (1) DEX_{it-1} (Dependencies of Exogenous Codebases) denotes the dependencies of exogenous codebases used in a focal software codebase at time t-1. The dependency indicates the connection similarity of a neighboring structure of an exogenous codebase in the structure of a network. (2) TR_{it-1} (Topical Representations) denotes the topics of the functional categories of software packages, where the topic implies the thematic similarity of a software package with other packages' function at time t-1. The thematic similarity is measured by the uses of words associated with its function as described by its developers. Our controls include the number of developers, each software codebase update, software codebase size, and R platform update. After understanding the genres used in the R platform, we analyze the descriptive statistics shown in Table 1.

Variable	Description	Mean	STD	Min	Max
Dependent Variable					
log(WD_{ij})	Weekly Download (log)	4.025	1.465	0	13.55
Independent Variables					
MEX_{it-1}	Dependencies of Exogenous Codebase (%)	36.12	15.49	0	99.165
MTR_{it-1}	Topical Representations (%)	49.539	28.602	0	99.795
Control Variables					
NDV_{it-1}	Total # of Developers	1.560	2.973	1	110
$PKUP_{it-1}$	Package Update	0.0002	0.013	0	1
PKS_{it-1}	Package Size (MB)	0.934	3.023	0.001	102.283
RUP_{it-1}	R Platform Update	0.094	0.292	0	1

Table 1. Descriptive Statistics

We use panel regression models to estimate the coefficients of our model as displayed in Table 2 below.

DV: log(WD_{ij})	Panel OLS Model 1	Panel OLS Model 2
MEX_{ij-1}	0.010^{***} (0.0001)	0.012^{***} (0.005)
MTR_{ij-1}	0.021^{***} (0.0003)	0.021^{***} (0.001)
Constant	2.405^{***} (0.190)	0.726^{***} (0.080)
Observation	1,294,031	1,294,031
Number of Groups	8,284	8,284
R^2	0.297	0.379
Package Fixed Effects	No	Yes

Time Fixed Effects	No	Yes
Robust Standard Error	No	Yes

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Table 2. Panel OLS without including control variables

The coefficient of DEXij-1 is positive and significant at the 0.01 level, showing the outbound dependency of exogenous codebases with more dependencies have greater sales than those which have less dependencies, supporting H1. The coefficient of TRij-1 is positive and significant at the 0.01 level, highlighting the importance of selecting appropriate topical representations regarding the change of popularity. This indicates that the topical representation is positively associated with the number of downloads, supporting H2.

Conclusion

Our study complements the theory of digital innovation by emphasizing the significance of leveraging the functional usefulness of codebases in a digital platform ecosystem. We emphasize unintended developer collaborations through sharing software codebases in a platform ecosystem. Our findings suggest that developers can be successful in product design by efficiently obtaining competitive advantages and (re)combinatorial design can be a useful strategy in changing technological environments. We characterized the effective uses of codebases by capturing the changing dependencies of codebases' uses. The adoption of exogenous codebases does not guarantee that digital products will be technologically advanced. Developers who follow existing design patterns may find it hard to attain functional benefits expected by users. We focus on which exogenous codebases need to be adopted for effective design, how market trends can be understood, and how changes can be appropriately implemented in digital products. We emphasize agility to reflect product design's topical representation in a digital ecosystem. Developers create diverse digital products characterized by functional uniqueness to attracts users. To be successful, developers need to effectively assess market uncertainty so their digital products appropriately reflect market demands. Digital products cannot provide unique user experiences if they reflect an outdated design pattern. Our findings suggest the importance of efficiently show topical representations in descriptions to increase product competence in a digital ecosystem. Because developers rarely understand exactly what functional demands are preferred and pursued, their designs need to efficiently capture prevailing technological trends.

REFERENCES

Boudreau, K. J., 2012. Let a thousand flowers bloom? An early look at large numbers of software apps developers and patterns of innovation. Organization Science 23 (5), 1409–1427.

El Sawy, O. A., Malhotra, A., Park, Y., and Pavlou, P. A. 2010. "Research Commentary—Seeking the Configurations of Digital Ecodynamics: It Takes Three to Tango," Information Systems Research (21:4), pp. 835-848.

Ethiraj, S. K., Levinthal, D., and Roy, R. R. 2008. "The Dual Role of Modularity: Innovation and Imitation," Management Science (54:5), pp. 939-955.

Foerderer, J., Kude, T., Mithas, S., and Heinzl, A. 2018. "Does Platform Owner's Entry Crowd out Innovation? Evidence from Google Photos," Information Systems Research (29:2), pp. 444-460.

Gong, J., Abhishek, V., and Li, B. 2018. "Examining the Impact of Keyword Ambiguity on Search Advertising Performance: A Topic Model Approach," MIS Quarterly (42:3), pp. 805-830.

Haefliger, S., Von Krogh, G., and Spaeth, S. 2008. "Code Reuse in Open Source Software," Management science (54:1), pp. 180-193.

Parker, G., Van Alstyne, M., and Jiang, X. 2017. "Platform Ecosystems: How Developers Lnvert the Firm," MIS Quarterly (41:1), pp. 255-266.

Sambamurthy, V., Bharadwaj, A., and Grover, V. 2003. "Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms," Mis Quarterly), pp. 237-263.

Tiwana, A. 2018. "Platform Synergy: Architectural Origins and Competitive Consequences," Information Systems Research (29:4), pp. 829-848.