Too Many Complementors? Evidence on Software Developers

Kevin Boudreau*
HEC-Paris School of Management

January 2007

It is often presumed there are benefits from growing numbers of buyers and sellers around a platform, or “network effects.” While the benefits of more buyers are self-evident, the effects of ever-increasing numbers of sellers of complementary goods (“complementors”) perhaps deserve closer scrutiny. This paper uses point-of-sale data on a textbook example, large numbers of software developers around mobile computing platforms, to show that entry beyond several hundred software firms in fact led to reduced rates of new software development. Platform strategies emphasizing careful regulation of complementor entry would have led to more and possibly better software.

**JEL Classification** L1, L86, 03

**Keywords:** Platforms, licensing, network effects, organizing for innovation, market structure and innovation.

Within the broader question of organizing for innovation (Teece 1986; Aghion and Tirole 1992) an innovator might choose to organize as a platform—retaining control over just a subset of a technology, while outsiders produce interoperating complementary components. A most striking empirical regularity setting these cases apart from more usual multi-firm arrangements (*e.g.* alliances) is the sheer number of partnerships involved. Perhaps most famously, Microsoft is surrounded by hundreds of thousands of complementors including hardware designers, component suppliers, value added resellers, system integrators, independent vendors, trainers and more. NTT Docomo’s i-mode platform allows its mobile phone users to connect with roughly 100,000 content and service websites. Internet retailer Amazon.com allows tens of thousands of merchants to use its website as a storefront while at the same time allowing about 200,000 outside software developers to build “add-on” programs to

---

* boudreau@hec.fr. This is a draft; comments are most welcome. (Updates of this paper at web.mit.edu/boudreau/www.) I would like to thank managers at PalmSource, Psion, Symbian, Montavista, Trolltech, Lynuxworks and numerous software developers for contributing perspectives on the history of the industry. Handango’s team, especially Jason Wells, deserves special thanks for contributing data and insightful on-the-ground knowledge. All errors are mine.
its on-line store. Such fantastic numbers have led market structures of this sort to be analogized as “business ecosystems,” competing “networks,” “alliance constellations,” “modular clusters,” and innovation “communities.” The widely-held conventional wisdom has become that platform suppliers should “develop as many partnerships as possible with companies that make so-called complementary goods” (The New Yorker 2006). This paper seeks to clarify the boundary conditions of such strategies and whether bigger “ecosystems” are necessarily better by studying the effects of entry in a textbook case: software firms around mobile computing platforms. The analysis shows that it would have been more productive in this case for platforms suppliers to restrict the licensing and entry of software developer firms.

Problems of organization in platform contexts might, in a sense, come closer to managing market structure. Business strategies of platform suppliers have emphasized building “critical mass” and getting both buyers and sellers “on board” the platform (Schilling 2002; Eisenman, et al. 2006). Several suggested approaches to this problem include: strategic subsidies (Varian and Shapiro 1998); optimal pricing structures (Rochet and Tirole 2003, 2006; Parker and van Alstyne 2005); modular design, standardized interfaces, and other means of reducing coordination costs (Garud and Kumaraswamy 1995); facilities to ensure transparency and to reduce consumer search costs (Bakos 2007; Dellarocas 2003); trust between platform suppliers and complementors (Gawer and Cusumano 2002; Yoffie and Kwak 2006); and politico-institutional processes intended to gain support for a platform as an industry standard (Tushman and Rosenkopf 1992; Garud, Jain, and Kumaraswamy 2002).

Software developers around computer platforms are perhaps the most frequently cited example where platform suppliers engage in entry-promotion strategies. Developers may, for example, bring varied experience or specializations (Chesbrough, et al. 2006), idiosyncratic knowledge of consumer needs (von Hippel 2005), or varied motivations for doing the work (Kuan 2001; Hertel, et al 2003; von Hippel and von Krogh 2003). Where software development involves uncertain or stochastic “search” (i.e., trying to find the “killer app”) under certain conditions it will be the case that many searchers will have better luck than few searchers (Baldwin and Clark 2002). Free and unhindered entry might also be congenial to emergent coordination, trust and a norms of collaboration (Raymond 1998). Alternatively, high levels of entry might otherwise intensify software competition, which could potentially increase demand for the platform (Katz and Shapiro 1994).

But to account for the kinds of fantastic numbers seen earlier, practitioners and researchers most often point to “network effects” (virtuous circles or positive feedback) between comple-

---

mentor entry and platform adoption by end-users. Growing numbers of complementors can enhance demand for the platform; a larger “installed base” of platform users can then entice further entry; and on and on. In a simplest example, a network effect is created if buyers simply prefer platforms with more sellers and *vice versa*, as in a trading platform or matchmaking service. In the case of software, entry by numerous developer firms is understood to contribute to network effects through some combination of wider software selection, mounting investments, improved quality, and declining prices (Church and Gandal 1992; Katz and Shapiro 1994; Tirole and Rochet 2003; Nair et al. 2004; Hagiu 2006; Evans, et al. 2006; and others). To date, empirical analysis has confirmed the existence of positive feedback between product variety and end-user demand in several contexts (Gandal, et al. 2000; Nair et al. 2004; Rysman 2004; Clements and Ohashi 2005).

But whether ever-increasing entry and fragmented supply best promotes this variety—or other benefits—remains untested. Long-observed possible costs of entry include exacerbated asymmetric information and search costs, coordination costs, and deteriorated investment incentives. The interactions across a platform might even be ambivalent. For example, growing numbers of advertisers may bring growing resources to an Internet portal, but might also pose a nuisance to consumers.

This paper studies the effects of varying levels of entry by software firms around mobile computing platforms, a widely-cited example of network effects (McGahan et al. 1997; Nair et al. 2004; Tirole and Rochet 2003; Evans et al. 2006). Growing entry was particularly understood to increase the value of platforms by expanding the selection of software available to end-users. So-called “third-party developer networks” of software firms numbered in the thousands. Descriptive patterns from the industry show an obvious first order correlation between large groups of developers and large software selection; the narrower empirical question is whether the positive association is indeed causal and whether restricting entry to some intermediate level might have generated yet more innovation of new titles. From an empirical standpoint, the context is desirable for considering what might have happened had platform suppliers more carefully regulated entry—precisely because they did not do so. As elaborated herein, it becomes possible to then draw inferences about the causal effects from variation in

---

2 These sorts of network effects are referred to as “indirect” in that they are created by interactions between buyers and sellers across the platform. (“Direct” network effects are those resulting from positive externalities acting directly among adopters.). A vital stream of research on indirect network effects and closely-related “two-sided markets” has grown considerably in recent years. See for example: Caillaud and Jullien (2003), Tirole and Rochet (2003, 2006), Evans and Schmalensee (2005), Economides and Katsamakas (2006), Armstrong (2006), Hagiu (2006).
numbers of software firms.

The analysis finds the widely-observed positive association between “more developers and more software” is in fact largely the result of spurious correlation with other factors—most notably the characteristics and “quality” of the platforms, themselves. Entry by more than just several hundred developers led to slower overall development rates. Individual developers’ price-setting and product development responses to the entry of close substitutes further suggests that the negative effect is created by crowding-out of investment incentives at high levels of entry. Thus, the patterns suggest that even in platform contexts with network effects, traditional innovation concerns for maintaining investment incentives and dynamic efficiency remain paramount (Arrow 1962; Dasgupta and Stiglitz 1980). The conclusions suggest a role for platform suppliers to play managing the number and identity of complementor entrants to avoid the crowding-out of investments.

Section 1 introduces the empirical context of mobile computing software development. Section 2 presents the data. Section 3 presents the empirical approach. Section 4 presents the analysis and results. Section 5 discusses the findings and implications for platform strategy in the light of prior research. Section 6 concludes.

1 “Third-Party Developer Networks” in Mobile Computing

The mobile computing (PDAs and smartphones) industry has historically been dominated by a handful of leading platform suppliers, each surrounded by large groups of partners (Gomes-Casseres 1996). Leading platform suppliers have included Palm, Microsoft, Research in Motion (RIM), Symbian, and its predecessor, Psion. Among followers, several Linux platforms—such as those by Montavista and Wind River—are notable for rapid growth, if not yet absolute market share. Partners included software developers, chipmakers, peripheral hardware makers, trainers, technical consultants, value-added resellers and others. Among complementors, software developers were understood to be the most important in creating network effects (Nair et al. 2004). The consensus view in the industry was “PDAs will become increasingly valuable when other users adopt the product. . . The variety and range of available software will increase with the number of users... Similarly, software that enhances the utility of a PDA will grow as the user base expands” (McGahan, et al. 1997). The

3Several dozen other platforms have been released the history of the industry, beginning in the 1980s, but captured only a small proportion of overall sales.
complementarity between software variety and platform demand has since been demonstrated econometrically (Nair, et al. 2004).

Software development by outsiders proved important even in very early industry development. Early market leader Psion (later to become Symbian) allowed users as early as the 1980s to customize devices by supporting custom software programs, modular memory cartridges and connection slots for peripheral devices. The strategy led Psion to gain large volume contracts with large corporate customers, such as British Telecom and Marks & Spencer. But Psion’s attempts in the early 1990s to mimic the “third-party developers networks” and network effects of the personal computing fell flat. Problems included immature development tools and underdeveloped programming interfaces, as well as still-evolving (non-standardized) hardware specifications. But still worse factors were exorbitantly priced solid state proprietary memory discs; a still immature market size that numbered in just the hundred of thousands of units; and the simple fact that outside developers had nowhere near the experience, skill and knowledge of the Psion platform that Psion’s own in-house software development team possessed.

Barriers to third-party developer networks and the possibility of fomenting network effects fell by the mid-1990s. Widespread adoption of the Internet and advancing memory technologies made packaging, transmitting and storing programs qualitatively easier. A new generation of designs, such as the famous Palm Pilot of 1996, grew the market to millions of units per year—a level that could conceivably sustain independent developer firms. Suppliers of integrated development environment (IDE) tools, such as Metrowerks and Broderbund, appeased the need for platform suppliers to develop their own tools or for developers to learn new tools. Technical specifications and interfaces of devices stabilized for the first time and programs began to remain compatible across multiple platform versions. Great pains were also taken by then new platform suppliers, such as Palm and Microsoft, to design systems in a modular fashion from the “ground up,” fully anticipating arm’s length third-party development. Psion, soon then to become Symbian, was itself in the midst of a complete modular redesign of its platform in the mid-decade. Later entrants, such as RIM and Linux variants would adhere to these same design principles and eagerly ported their platforms to middleware (e.g. mobile Java or Trolltech Qtopia) giving them access to ready-made development

---

4Litchfield (1998), Langdon (2001)
5Even the simplest applications such as winelists or transportation schedules were priced on the order of $100.
6Potter (1998)
capabilities and software programs.

The Modern “Liberal” Approach to Software Complementors  Notwithstanding slight differences across platforms, regular changes with platform updates, and sporadic complaints from developers, it can be generally said that a relatively stable period of “liberal” licensing practices began by the mid-late 1990s that persists to this day. With basic programming skills and small investments in mostly standard computer equipment, a developer could enter and begin working without much complication. A license contract, documentation, and a development kit could be acquired by simply visiting platform suppliers’ websites or that of an IDE partner. Technical and often other sorts of support were available from staff within “developer relations” departments, FAQs sections of platform suppliers’ websites, growing user and developer on-line communities and in-person meetings, and regular developer newsletters. Marketing and retailing to customers were available through similarly straightforward on-line processes from on-line “aggregators,” such as Handango.

In this environment thousands of developers entered around leading platforms.8 Entrants included an array of entrepreneurial ventures, existing commercial software companies extending their desktop products to mobile computing, consultants whose bespoke projects evolved to packaged software, part-time professionals, hobbyists and individual programmers. This mix of entrants persisted even as the task of conceiving novel and improved software programs gradually grew more difficult over time, given counterbalancing periodic improvements in tools functionality of the devices themselves. Individual and part-time developers continued to compete alongside larger firms. As a result, the “average” software firm is widely described as a “3-person shop making several thousand dollars a month.”9

This liberal, low-cost, and arm’s length strategy towards third-party developers led software titles and also the numbers of developers to grow to tens of thousands (Figure 1). However, it remains an open question whether organizing software developer networks in some alternative configuration—perhaps inviting fewer firms, carefully selecting among them, or perhaps yet more aggressively lowering barriers to entry—might have yielded superior outcomes. While we do not observe these counter-factuals in historical data, the empirical

---

8Precise numbers of total developers are not available. Platform suppliers published numbers of downloads of developer kits, which frequently exceeded one hundred thousand. These should be a high estimate of actual software firms, given multiple downloads per developer, multiple developers per firm, and downloads by students and others not intending to produce third-party software.

9This was a consistent observation across interviews with multiple developers, interest groups, platform suppliers and leading retailer, Handango.
analysis to follow exploits the variation in levels of entry that did in fact exist so as to infer what outcomes might have emerged under alternative strategies.

Figure 1 The Generation of an Extraordinary Selection of Software Titles for Leading Platforms

2 Data

Proprietary point-of-sale data from the leading (on-line) retailer of mobile computing software, Handango Inc, were acquired for this study. The data include names of available titles, the developer firm name, the platform for which titles were designed, a broad indication of the category of software, and sales volume and revenue data. Data are quarterly and extend from the fourth quarter of 1999 to the end of 2004—and therefore relate to the period of relatively stable liberal treatment of entry, as described in the earlier section.

The original data set contains 56,759 products. To focus on software titles most clearly related to widely-held conceptions of network effects several data points were eliminated: non-software products (hardware, accessories and peripherals); software titles whose platform was not specified; server-side software; and multi-user contracts. This resulted in less than five percent reduction in the data set. Eliminating media titles (music, themes and backgrounds, digital content, graphics, e-books, guides etc.) that were not applications software (and most often not specific to any particular platform) then reduced the data to 19,005 titles. These included titles from eight platforms: Palm, Microsoft, Symbian, RIM, Linux, Motorola, Casio and Java. The panel is unbalanced as Java and RIM enter to the data set in 2002.

Descriptive statistics from the sample (Table 1) reveal that third-party developer networks large developer networks were large with many firms. These were populated by small firms

---

10 Internet retailers or "aggregators" of software dominated distribution in this industry. No major brick-and-mortar retailer carried meaningful selection or volume of mobile software titles. It is difficult to confirm market shares of Internet retailers, but there is broad consensus that Handango was far and away the market leader during the sample period, carrying perhaps half of all transactions.

11 Java deserves special mention as it is a "middleware" platform in the sense that it could run on each of the other platforms. This level of compatibility tended to limit Java to simple applications (largely games) that were not sensitive to hardware differences across systems and did not need to exploit low-level coding. Excluding Java in the analysis does not influence results to follow, nor do attempts to identify distinct effects of Java compatibility on different platform (not reported).
with modest revenues, on average. Software firms generally developed for a single platform; there were few “multi-homing” firms.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>StdDev</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Developer Networks&quot; around Platforms (N=135 platform-quarters; 8 platforms)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. Developers</td>
<td>365</td>
<td>579</td>
</tr>
<tr>
<td>New Title Releases</td>
<td>382</td>
<td>589</td>
</tr>
<tr>
<td>Unit Sales</td>
<td>45,917</td>
<td>75,153</td>
</tr>
<tr>
<td>Revenues, nominal</td>
<td>$778,757</td>
<td>$1,301,909</td>
</tr>
<tr>
<td>Developer Firms (N=27,751 developer-quarters; 3893 developers)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Titles</td>
<td>1.1</td>
<td>4.6</td>
</tr>
<tr>
<td>Revenues, nominal</td>
<td>$2,486</td>
<td>$14,852</td>
</tr>
<tr>
<td>Multihoming on Multiple Platforms</td>
<td>1%</td>
<td>n/a</td>
</tr>
<tr>
<td>Software Titles (N= 102,332 title-quarters; 19,382 titles)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit Sales</td>
<td>30</td>
<td>199</td>
</tr>
<tr>
<td>Average Price</td>
<td>$13.53</td>
<td>$15.08</td>
</tr>
<tr>
<td>Free Software</td>
<td>1%</td>
<td>n/a</td>
</tr>
<tr>
<td>Revenues, nominal</td>
<td>$507</td>
<td>$4,066</td>
</tr>
</tbody>
</table>

(1) Maximum and minimum values are confidential

Table 1: Descriptive Statistics

Not surprisingly, given earlier characterizations of the industry, numerous patterns in the data are consistent with indirect network effects based on growing software selection. To begin, the thousands of titles supplied represent extraordinary variety (Figure 2). Visual inspection of the titles in the data set reveals innumerable subcategories, from calculators, calendars, clocks and timers, spreadsheets, word processors, chess games, web browsers, and on and on. Demand is also spread widely across these titles, consistent with a demand for variety (Figure 3).

The data would also seem to affirm the widely-held intuition that many software firms should lead to a wide selection of software titles. Not only is there a clear positive association between numbers of developer firms and the range of titles available (Figure 4); there is a positive association between number of developer firms and the rate at which new titles are developed (Figure 5). There also do not appear to be palpable economies of scale at the level of individual software firms. Concentration within each developer network is lowest in the largest and most mature developer networks (Figure 6) and decreases over time (not shown).12 Although there appear to be certain developer firms in the data with high market

12This is consistent with the casual observation that developers did not attain a scale which would allow them to accrue, say, brand awareness, advantages in distribution (they generally all used the same third party channels), or power in markets for technical talent.
share within their platforms, this is limited to cases of small networks in early time periods when markets were particularly thin.

While it is not likely that growing entry and fragmented supply would lead to, say, coordination problems (given the arm’s length regime set-up by platform suppliers), lost economies of scale (given no apparent tendency to market concentration) or increasing customer confusion and asymmetric information (given the presence of Internet retailers with searching and rating capabilities), there would seem to have been scope for diminished investment incentives. The pattern perhaps casting most doubt on the value of entry and perhaps its costs is that the majority of new titles do not come from new entry. Roughly nine of ten titles are developed by incumbent software firms releasing multiple titles. In all, over half (53%) of developer firms in the sample release multiple titles in the sample. Those that do release roughly seven software titles within the sample. The generation of new titles was then more sensitive to factors affecting the development decisions of incumbent developers than it was to new entry. Further, visual inspection of titles in each platform and even among only the top titles reveals not just different types of titles but also differentiated versions of the same program types, such as multiple spreadsheet titles or multiple translator software titles from competing developer firms. Further, the most common price trend over time (not reported)—this, despite upward pressures on pricing from development costs, increasing quality of software and potentially greater utility conveyed with growing adoption.

One possible argument for benefits of keeping marginal entrants even if they were to deteriorate investment incentives would be that it is difficult to know which firms will developer “hit” products ex ante. But the relatively flat distribution of sales across titles suggests there were no real hit titles in the first place. There also appears to be little uncertainty in the process—at least enough to account for letting in thousands of developers. Churn into and

---

13Firms releasing just one title also tended to fetch lower prices and sell smaller volumes, hinting at the possibility that their product quality was also lower than average.
out of the top 100 firms in each platform is less than 5% per year, with exceedingly few firms jumping anew to the top ranks, or from a low rank to a top rank.

While these trends do not conclusively demonstrate a crowding-out of investment incentives, neither does a the simple association (correlation) between more developers and more titles prove a positive causal relationship. The analysis to follow attempts to more carefully evaluate whether growing networks lead to more or less software development.

### 3 Empirical Approach

Empirical research studying interactions around platforms typically estimates two equations (e.g. Gandal et al. 2000; Rysman 2004; Nair et al. 2004; Clements and Ohashi 2005). The first equation relates platform demand, \( S \), to the platform (and embodying hardware), itself, and the expected utility from associated complementary products. Let this ancillary benefit be \( Z \). In most prior work \( Z \) is the range of compatible software.\(^\text{15}\) A second equation then relates the generation of \( Z \) to numbers of platform adopters, \( S \). The two equations together imply positive feedback between \( S \) and \( Z \). This can be seen in the simplified formulation: \( S = S(Z) \) and \( Z = Z(S) \). The regression of the availability of software titles, \( Z \), on market size, \( S \), is typically motivated by and given structural interpretation with a model of differentiated market entry, where the zero-profit level of entry, \( N \), is likened to equilibrium product variety. In effect, \( Z \) is assumed equal to \( N \).\(^\text{16}\)

This paper studies the correctness of this last assumption and provides a novel examination of the *supply structure underlying network effects*. The idea is to study data that allows \( Z \), in this case software variety, to be distinguished from numbers of firms, \( N \). In studying how the generation novelty relates to supply structure, the analysis relates most closely to empirical research market structure on innovation (Cohen and Levin 1989; Nickell 1996; Aghion et al. 2005). Following this tradition, I employ panel data methods and allow for possible non-linear relationships with flexible estimators in relating a measure of innovation intensity (quarterly counts of new software title releases) to a measure of market structure (number of software firms). Distinct from this tradition, the unit of analysis is the group

\(^{14}\)The purchase of hardware built on a platform is typically the major sunk and durable investment of a consumer into a particular platform.

\(^{15}\)More precisely, the expected utility deriving from purchases of music titles over time, given the selection, prices, and quality that emerge in equilibrium.

\(^{16}\)More weakly, the assumption might be interpreted, perhaps, as a linear or at least monotonic relationship between the \( Z \) and \( N \).
of developer firms around a platform, rather than an entire industry of firms. To begin, an analytic framework is developed—not to impose constraints on the econometric model—but to anticipate identification challenges and to further clarify commonalities and departures from prior work.

### 3.1 A Characterization of New Software Title Development

Consistent with earlier research, I interpret essential features of software industry structure as consistent with a canonical model of differentiated entry.\(^{17}\) Consider \(N\) software firms, indexed by \(i \in \{1, N\}\), with products ranging from accounting software to video games and the spectrum in-between. These firms locate themselves along differentiated positions in a product space, abstracted as a circle of circumference one. For simplicity, consider demand of mass \(S\) to be equally distributed along the circumference of the circle. Market opportunity \(S\) is responsive to shocks, such as demographic changes or shifts in the broader value chain, \(\Theta\). With indirect network effects at work, the market opportunity is also responsive to the entry of software developers, \(N\), or \(S = S(N; \Theta)\).\(^{18}\)

Distinct from prior work, I assume software firms invest, \(z\), in a level of product quality (alternatively, appeal or “reach”), \(U\).\(^{19}\) The product development function is assumed to take a simple concave form, \(U_i = z_i^\gamma\), \(0 \leq \gamma < 1\). As in the canonical set-up (Salop 1979), a consumer’s preference to buy from a particular firm is determined by the quality, proximity on the circle and price, \(V_i = U_i - \alpha x - p_i\) (where “distance” to each product is \(x\) and \(\alpha\) is the cost per distance).

As this characterization is simply intended to provide a coarse characterization as a basis for the empirical specification, the set-up is solved as a one-shot symmetric equilibrium where each consumer makes one purchase (also consistent with past work). This static approach ignores dynamic demand issues such as intertemporal substitutions and expectations and resulting strategic interactions (Gowrisankaran and Rysman 2006).\(^{20}\) Finally, it is assumed

---

\(^{17}\)See, for example: Anderson, De Palma & Thisse (1989), Chou and Shy (1990), Church and Gandal (1992), Katz and Shapiro (1994) and particularly their discussion of indirect network effects in software, Park (2000), and Gandal et al. (2000) This approach abstracts from possible strategic interactions between platforms, an issue which will be revisited in specifying the empirical model.

\(^{18}\)This formulation follows the convention of emphasizing demand elasticity to supply-side changes associated with the network effect, rather than elasticity to price changes.

\(^{19}\)Empirically, this may be thought of as some combination of higher quality of a given product, new versions, or perhaps dispersion of greater numbers of products over a wider product space.

\(^{20}\)In contrast to markets for durable hardware, these issues are far less likely to play-out here where software is inexpensive, upgrade-able and trial-able software—and served by many small firms.
that fixed entry costs are minimal in relation to development costs, consistent with minor equipment and costs of specializing programming skills in this industry. The overall product development effort that is then exerted by the entire network of software developers around a platform is then as follows:\footnote{By direct inspection, overall product development investment, $Z$, might either increase or decrease with changing network size, $N$, depending on the relative strength of market expansion from network effects and competition effects. This can be true, even under the conventional assumption that entry always provokes market expansion, $S'_N \geq 0$.}

\[
Z = N \cdot z = \left[ \frac{\gamma S(N; \Theta)}{N^\gamma} \right]^{\frac{1}{1-\gamma}}
\] (1)

Taking natural logarithms of both sides and assuming determinants of the market opportunity are linearly separable gives the following expression. (Lower case $s$ with appropriate subscripts denotes the separated components of market opportunity).

\[
\ln Z \equiv Z = G(N) + \ln \frac{\gamma + s \Theta}{1 - \gamma}
\] (2)

Terms related to the number of software firms, $N$, are collected under $G(N) \equiv \frac{1}{1-\gamma} \{s_N(N) - \gamma \ln N\}$. This expression illustrates that the response of product development to growing numbers of developers could be either negative or positive, depending on the relative strength of market-expanding network effects and competitive crowding.

The level of entry, $N$, is likely itself to be endogenous and responsive to many of the same factors as investment in product development, $Z$.\footnote{Following this characterization, the level of entry may be thought to be the free entry level, plus some plus some deviation, $\Delta$, or $N = N^* + \Delta = S \left[ \frac{n^{1-\gamma}}{\gamma} \right]^{\frac{1}{1-\gamma}} + \Delta$, $\Delta \in \mathbb{R}^{+/-}$.} The empirical challenge is then to infer how the exogenous component of variation in $N$, such as shifts in the supply of developers in the broader computer industry or random fluctuations, affected overall product development, $Z$.

### 3.2 Estimating $G(N)$ with an Upward Bias

Studies of market structure and innovation typically concern themselves with at least two sources of biases. One is reverse causality, which does not appear to present a problem here.\footnote{Nickell (1996) and Aghion et al. (2005), for example, point to market structure evolving in response to innovation in the sense of, say, Sutton (2001) or Klepper’s (1996) work.} The second is omitted variable bias where entry, $N$, and product development, $Z$, are determined by many of the same (unobserved) factors. This latter problem is unavoidably...
worse in the presence of indirect network effects, as explicit controls for market opportunity cannot be included without “soaking up” the endogenous market-expansion resulting from entry and network effects, or \( s_N(N) \) in the earlier notation. To attempt to mitigate any bias created by uncontrolled market opportunity the model will include fixed effects for each platform, \( \eta_i \), and year, \( \nu_t \). Remaining unobserved (platform-specific and transitory) variation in market opportunity should create an upward bias, as both investment and entry should respond to market cues in the same direction.\(^{24}\)

Changes in investment productivity, \( \gamma \), might present other complications. The most intuitive problem—that development might progressively become more difficult as new opportunities are “fished out”–is addressed by including controls for cumulative past development. The opposite problem of increasing productivity with platform enhancements is controlled with an explicit count of platform version.\(^{25}\)

Less straightforward is the possibility of interactions between productivity and other factors. An interaction between productivity and the size of network, \( N \), such as \( \gamma \ln N \) as in the earlier formulation, would lead to upward biased estimates of \( G(N) \) if productivity, in terms of utility per dollar, declined as the network grew large. Unobserved interactions with control variables in the error term could exacerbate this problem. Given limits to identification, the controls for changing productivity then relied upon to “detect” any large potential distortions.

Other controls are also added to account for factors abstracted from the earlier analytical framework, such as measures of software developer firms developing for multiple platforms (“multi-homing”) and market share of platforms as a whole.\(^{26}\) The resulting empirical model is then as follows:

\[
Z_{it} = G(N_{it-4}) + \eta_i + \nu_t + \beta \cdot \Phi_{it} + \varepsilon_{it} \tag{3}
\]

\(^{24}\)The direct partial effect of an increase of \( S \) on both \( Z \) or \( N \) in the earlier framework is positive. Although the net effect of entry with growing markets could intensify competition and therefore plausibly diminish investment, prior research suggests this is not likely with the large number of firms observed here (Bresnahan and Reiss 1991).

\(^{25}\)Firm-specific differences in productivity might also lead to changing levels of new title development. This might, for example, lead to progressively lower contributions with marginal entrants. However, firm heterogeneity on its own cannot account for a negative relationship.

\(^{26}\)Earlier-mentioned panel controls for time and platforms address potential variation in Handango’s own market share in relation to other distribution channels over time. They also address variation in unobserved developer populations that choose to not participate in mainstream commercial distribution channels (such as freeware and open source).
where product development intensity, $Z$, of the developer network around platform $i$, at time $t$, relates to the number of firms, $N$, by some unknown function $g$, a platform effect $\eta$, a year effect $\nu$, a vector other controls $\Phi$, and an error term $\varepsilon$. Numbers of developers are lagged one year to account for developer response to cues at the start of product development, rather than at the end.\textsuperscript{27} Strictly speaking, little can be said about the structure of the error term \textit{a priori}. Huber-White standard errors are employed to account for unknown heteroskedasticity and serial correlation of errors.

The following section presents results for both parametric estimates of $G$ and flexible, differenced-based semi-parametric estimators (Yatchew 1998, 2003) using locally-weighted regression methods (Cleveland and Loader 1996).\textsuperscript{28}

4 Analysis & Results

4.1 Rates of Product Development and Number of Firms

An initial set of regression results approximating $G$ as a simple linear relationship are reported in Exhibit R1. Consistent with common wisdom and earlier scatter plots, a simple OLS\textsuperscript{29} regression of (the log\textsuperscript{30}) of the rate of new title development on No. of Developers reveals a positive, statistically significant relationship (M1-1).\textsuperscript{31} However, this positive relationship is startlingly fragile. While controlling for time effects has minimal effect (M1-2), adding platform fixed effects reverses the sign of the first order slope estimate. The coefficient drops from the earlier .003 (M1-1) to -.0004 (M1-3). The negative coefficient is only statistically significant at $p=14\%$; but nonetheless it is telling that the coefficient is negative at all—especially in a crude first order approximation, in an upwardly-biased specification. Even a zero relationship would be an important departure from a supposed positive relationship.

\textsuperscript{27} Adjusting the lag does not affect results substantially. Dynamic issues, more generally, may be less pronounced in relation to, say, the development of new patentable inventions in pharmaceuticals or process and organizational innovations within a firm. By implication, they may be less drawn out, less subject to "adjustments" and pose fewer dynamic problems for the econometric analysis.

\textsuperscript{28} A variety of more and less flexible nonparametric methods generate essentially the same results.

\textsuperscript{29} This and other regressions are insensitive to restrictions of variance such as Poisson or Negative Binomial distributions.

\textsuperscript{30} Results are not sensitive to the logarithmic transforms of either dependent or main explanatory variables. The log-linear speciation simply more directly supports assumptions of separability (Section 3).

\textsuperscript{31} Perhaps easier to interpret, an OLS regression of raw counts of new title releases on number of developers produces a coefficient estimate of .8 new titles for every developer, every quarter.
<table>
<thead>
<tr>
<th>Unit of Analysis: Platform-Quarters (N=135)</th>
<th>All New Title Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model: (M1-1) (M1-2) (M1-3) (M1-4) (M1-5)</td>
<td></td>
</tr>
<tr>
<td>No. of software firms</td>
<td>0.003** (.0003)</td>
</tr>
<tr>
<td></td>
<td>0.003** (.0003)</td>
</tr>
<tr>
<td></td>
<td>-0.0004 (.0003)</td>
</tr>
<tr>
<td></td>
<td>-0.003** (.0005)</td>
</tr>
<tr>
<td></td>
<td>-0.003** (.0007)</td>
</tr>
<tr>
<td>Platform Version</td>
<td>1.3** (.2)</td>
</tr>
<tr>
<td></td>
<td>1.2** (.2)</td>
</tr>
<tr>
<td>Past cum. titles, same platform</td>
<td>0.0003** (.0001)</td>
</tr>
<tr>
<td></td>
<td>0.0002** (.0001)</td>
</tr>
<tr>
<td>Platform market share</td>
<td>2.5* (1.1)</td>
</tr>
<tr>
<td>Percent multi-homing</td>
<td>2.4 (1.5)</td>
</tr>
<tr>
<td>Constant</td>
<td>3.5** (.2)</td>
</tr>
<tr>
<td></td>
<td>3.4 (.1.9)</td>
</tr>
<tr>
<td></td>
<td>1.1** (.6)</td>
</tr>
<tr>
<td></td>
<td>3.0** (.4)</td>
</tr>
<tr>
<td></td>
<td>2.4** (.4)</td>
</tr>
<tr>
<td>Year fixed effects</td>
<td>Sign.</td>
</tr>
<tr>
<td></td>
<td>Sign.</td>
</tr>
<tr>
<td>Platform fixed effects</td>
<td>Sign.</td>
</tr>
<tr>
<td></td>
<td>Sign.</td>
</tr>
</tbody>
</table>

| R^2 | .22 | .41 | .91 | .94 | .95 |

Notes: * significant at 5%; ** significant at 1%; natural logarithm taken of dependent variable; number of software developers lagged one year; data include 8 platforms; standard errors in brackets, robust to heteroskedasticity and serial correlation of errors.

Exhibit R1: Parametric Regression Results

Adding other controls to ensure robustness of this result only leads to increasingly negative estimates of G. Adding controls for prior development (past cumulative title releases) and advances in the platform itself (platform version release) leads the coefficient on No. of Developers to drop to -.003 and to become statistically significant at conventional levels (M1-4).\(^{32}\) Adding controls for potential strategic interactions among platforms (market share of platforms) or the effect of developers developing on more than one platform (multi-homing) does not change results (M1-5).

To more precisely determine the nature of the negative relationship a flexible, non-linear estimator of G was used. Results of differenced-based semi-parametric estimates (Yatchew 1998, 2003) of G using locally-weighted regression methods (Cleveland and Loader 1996) are reported graphically in Exhibit R2.\(^{33}\) Panel I of Exhibit R2 shows the regression of new

\(^{32}\)To note, the sign on the lagged cumulative variable is positive rather than negative, as might be suggested by diminishing productivity with past releases. However, this sign should be understood as related to other factors factors correlated with the lag of cumulative new title releases, such as unobserved market opportunity. Any negative effect of diminishing productivity would be added to the other marginal effects that the cumulative variable is capturing.

\(^{33}\)Other methods used, revealing similar results, included parametric models of g, kernel estimators, and estimating separate means separate portions of the sample domain.
titles developed on No. of Developers with year and platform fixed effects removed. Panel II of Exhibit R2 shows the same regression, but with the natural logarithm taken of the dependent variable, as was motivated in Section 3 (analogous to M1-3). Panel III of Exhibit R2 shows results when the other control variables are added (analogous to M1-5). Each of these specifications reveals a negative-sloping portion of $G$ at high levels of entry. Findings are robust to dropping data from the platforms with the largest numbers of developers (Palm and/or Microsoft) or dropping data from any given year (not reported).

\footnote{The framework presented in Section 3 suggests this specification may violate the assumption of linear separability, but is included to simply describe the data.}
I. Linear with fixed effects

II. Log-linear with fixed effects

III. Log-linear with fixed effects, controls

Notes: In each of the regressions, quarterly measures of the number of new software titles generated in the entire set of developer firms around a given platform within a quarter is regressed on the number of developer firms. The estimates presented are based on differenced-based semi-parametric estimators (Yatchew 1998, 2003) using locally-weighted regression methods (Cleveland and Loader 1996). The bandwidth chosen was .2, but results were not particularly sensitive to parameter choices nor to which non-parametric approach was employed. Each model includes fixed effects for the platform and time effects. The third model also includes controls for past cumulative product releases, the version of platform release, the market share of the platform and the percentage of firms developing products for more than one platform in the given platform. All models are estimated using the unbalanced panel of eight platforms, described in Section 2.

Exhibit R2: Flexible, Semi-Parametric Regression Results
Therefore the analysis shows that the relationship is sufficiently negative to be detected with a coarse, simple first order model of $g$. But more flexible methods show this negative relationship to be apparent only at high levels of entry. As the introduction of simple fixed effects was sufficient to produce a negative-sloped estimate of $G$, it seems the casual observation of “larger networks, larger software libraries” can be largely imputed to a spurious correlation with characteristics of the platform itself, rather than a causal effect.

In the non-linear estimates, the maximum level reached with several hundred developers—far fewer than the observed thousands of software firms in these developer networks.\textsuperscript{35} Given that estimates of the curve are relatively “flat” at the peak, the precise estimated location varies a good deal depending on the particular specification used.

4.2 Firm-Level Patterns and Crowding-Out of Incentives

If faltering investment incentives are the cause of this negative effect, then the negative response to high levels of entry should be found in firm-level behavior. New title development at the level of firms ($z$ in the earlier framework) and numbers of developers is studied in an analogous log-linear model that is implied by the earlier framework development (Model 3). The unit of analysis used here is the firm’s product development within a particular platform, within a particular category of software genre (more on this later), within a given quarter. Therefore, firms diversified across multiple categories or platforms appear as multiple data points.\textsuperscript{36} The analysis faces the same sort of identification challenges as discussed in Section 3.

The variety of model specifications assayed\textsuperscript{37} each find a concave relationship that turns negative at high levels of entry (and in some specifications at much lower levels of entry than in platform-level regressions).\textsuperscript{38} Exhibit R3 shows the estimated partial relationship between

\textsuperscript{35} Regressing estimated error terms on lags finds a significant positive correlation with the single period lags (not higher lags). However introducing lags of the dependent variable or averaging the data over longer periods of time to reduce the effect of dynamic issues does not affect the basic findings.

\textsuperscript{36} Separating individual firm development by categories does not substantially change results, but has the advantage of allowing the effect of close substitutes to be examined at the individual firm level.

\textsuperscript{37} This included: log-linear specifications, as earlier, and untransformed linear-linear regressions with varying combinations of control variables; and a count model (Poisson with fixed effects) to explicitly account for the non-negative integer values taken by the dependent variable. Alternative weights of data points, including inverse residuals from a first-stage regression, were also assayed.

\textsuperscript{38} While we should expect that firm responses turn negative than the overall network of developers, given the direct effect of entry is positive, the possible biases in estimates and imprecision with which the peaks can be identified preclude a formal test of this hypothesis.
the rate of new title development and numbers of developer firms from semi-parametric and quadratic specifications. For consistency, the estimates shown are for models that most closely mirror the earlier models platform-level models (parametric model M1-5 and semi-parametric model III reported in Exhibit R2). The precise shape of relationship, location of the peak and severity of negative response are sensitive to model specification.39

---

Response of Developer Firms to Network Size

![Graph showing the relationship between the natural logarithm of the rate of new title development and the number of developer firms.](image)

Notes: In each of the regressions, quarterly measures of the number of new software titles generated in the entire set of developer firms around a given platform within a quarter is regressed on the number of developer firms. The estimates presented are based on differenced-based semi-parametric estimators (Yatchew 1998, 2003) using locally-weighted regression methods (Cleveland and Loader 1996). The bandwidth chosen was .2, but results were not particularly sensitive to parameter choices nor to which non-parametric approach was employed. Each model includes fixed effects for the platform and time effects. The third model also includes controls for past cumulative product releases, the version of platform release, the market share of the platform and the percentage of firms developing products for more than one platform in the given platform. All models are estimated using the unbalanced panel of eight platforms, described in Section 2.

Exhibit R3: Firm-Level Semi-Parametric Regression Results

If it is true that crowding out of incentives is the cause of the negative response to entry, it should also be true that the negative response should also be most pronounced among firms selling close substitutes. To study this possibility, the data were divided into categories of software types based on Handango’s own broad categories: business and professional, databases, education and reference, games, medical, multimedia and entertainment, personal productivity, translation and travel, and utilities. Higher entry to the same category should lead to a more negative effect than will entry to the broader network of software firms. (Spurious correlation with unobserved factors would, by contrast, find a positive correlation.). Given

39 The most sensitive portion of the curve is at low levels of entry, where alternate weightings of data points, model specifications or excluding different subsets of the data lead to a more or less upwardly sloping portion of the curve.
the relatively smooth shape of earlier estimates and given the within-category count of software firms will be somewhat collinear to total counts of software firms in a network, a more constrained parametric model (employing both a linear and quadratic term) is estimated.40

<table>
<thead>
<tr>
<th>Unit of Analysis: Individual Developers in a Given Category-Quarters (N=80,717 platform-firm-category-quarters)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent Variable:</strong> New Title Releases Average Price</td>
<td><strong>Model:</strong> (M2-1) (M2-2) (M2-3)</td>
<td></td>
</tr>
<tr>
<td>No. of software firms [000s]</td>
<td>.01</td>
<td>-.001</td>
</tr>
<tr>
<td></td>
<td>(.03)</td>
<td>(.02)</td>
</tr>
<tr>
<td>No. of software firms(^2)</td>
<td>-.03**</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(.01)</td>
<td></td>
</tr>
<tr>
<td>No. of software firms, same category [000s]</td>
<td></td>
<td>-.5**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.1)</td>
</tr>
<tr>
<td><strong>Platform Version</strong></td>
<td>-.02**</td>
<td>-.02**</td>
</tr>
<tr>
<td></td>
<td>(.01)</td>
<td>(.01)</td>
</tr>
<tr>
<td>Past cum. titles, same platform [000s]</td>
<td>-.01**</td>
<td>-.003</td>
</tr>
<tr>
<td></td>
<td>(.002)</td>
<td>(.002)</td>
</tr>
<tr>
<td><strong>Platform Share</strong></td>
<td>.06*</td>
<td>.06*</td>
</tr>
<tr>
<td></td>
<td>(.03)</td>
<td>(.03)</td>
</tr>
<tr>
<td><strong>Percent Multi-Homing</strong></td>
<td>.2</td>
<td>.1</td>
</tr>
<tr>
<td></td>
<td>(.1)</td>
<td>(.1)</td>
</tr>
<tr>
<td><strong>Constant</strong></td>
<td>-.03</td>
<td>.004</td>
</tr>
<tr>
<td></td>
<td>(.08)</td>
<td>(.08)</td>
</tr>
<tr>
<td><strong>Firm fixed effects</strong></td>
<td>Sign.</td>
<td>Sign.</td>
</tr>
<tr>
<td><strong>Platform x category F.E.</strong></td>
<td>Sign.</td>
<td>Sign.</td>
</tr>
<tr>
<td><strong>Year x category F.E.</strong></td>
<td>Sign.</td>
<td>Sign.</td>
</tr>
<tr>
<td><strong>R(^2)</strong></td>
<td>.16</td>
<td>.16</td>
</tr>
</tbody>
</table>

Notes: * significant at 5%; ** significant at 1%; natural logarithm taken of counts of new title releases (not price); number of software developers lagged one year; data include 8 platforms, 8 categories, 3,893 software developer firms; standard errors in brackets, robust to heteroskedasticity and serial correlation of errors

Exhibit R4: Firm-Level Parametric Regression Results

Regression results reveal that the negative effect of entry is caused only by firms selling close substitute software products. The coefficient on counts of firms within the same platform and category is negative, while the response in relation to total numbers of firms in the network is positive (M2-2). The negative relationship with close substitute firms is precisely the opposite finding of what either the “more developers, more software” view or simply

40 Using parametric relationships to estimate these apparently smooth curves also avoids possible distortions caused in estimating of parametric components of the model, in this case a differencing procedure (Yatchew 2003).
spurious correlation.\textsuperscript{41} These distinct—positive and negative—effects of entry offer further evidence consistent with a tension between network effects and competitive crowding in the developer network.

A further implication of growing competition, lowered appropriability and lower investment incentives should be the decline of prices with entry. To caution, relationships involving price are yet more difficult to study and identify in this context. Price should be highly endogenous and influenced by numerous factors and the direction of bias in estimates cannot be anticipated. Nonetheless, analysis at the level of firms within individual categories while controlling for fixed effects does much to at least mitigate these problems. The average price offered by a firm in a given category of software is modeled as the dependent variable in a regression framework similar to earlier regressions.\textsuperscript{42} Consistent with past studies of price levels in the context of network effects (Gandal 1994; Brynjolfsson and Kemerer 1995), price is found to be positively associated with the overall size of the network (M2-3). However, the effect of close substitutes is again found to be negative, consistent with intensifying rivalry and crowding.\textsuperscript{43}

5 Discussion

The findings most directly contribute to research on network effects and two-sided markets (Tirole and Rochet 2003, Armstrong 2006) by showing how traditional concerns for strategic investment incentives (Arrow 1962; Dasgupta and Stiglitz 1980) continue to play a role in contexts with platforms and network effects. Beyond managing entry and adoption patterns, platform suppliers might also pay heed to complementors incentives to invest. While the precise nature of strategic interactions created by growing entry is likely subtle and multifaceted,\textsuperscript{44} the strongest patterns found in these data appear to be consistent with prior claims

\textsuperscript{41}This pattern was also found when performing the regression at the level of total software produced per category within a platform. Repeating the regression, but on individual categories one-at-a-time (equivalent of allowing individual categories to take their own unique coefficients in addition to fixed effects), finds that 6 of 8 categories follow this same pattern, with a negative effect of close substitutes (not reported).

\textsuperscript{42}Given possible endogeneity bias in estimates, pricing decisions are modeled separately from investment decisions so as to avoid worsening any biases.

\textsuperscript{43}Prior empirical work on indirect network effects has generally discounted the value of diminished price in relation to growing variety. In this case, decreasing price may offset a loss of product selection. Although the magnitude of effect, 0.4 cents per thousand entrants, appears exceedingly small, this likely contains some unknown bias. Therefore the question remains mutable.

\textsuperscript{44}See Aghion and Griffith (2005) for a wide review of effects of intensifying rivalry on innovation.
that entry can stimulate higher rivalry and rent dissipation–eventually overwhelming network effects (Church and Gandal 1992; Economides 1996; Ellison and Fudenberg 2003; Augereau, Rysman et al. 2004). The findings–first an increasing and then decreasing slope of $G(N)$–are also consistent with network effects dominating a low levels of entry, but competitive crowding dominating at high levels of entry.\footnote{The combination of indirect network effects with competitive crowding might also provide another rationale for findings of “inverted-U relationships” between competition and innovation (Scherer 1967; Aghion et al. 2005).} This paper clarifies that the growing competition from entry can eventually result in inefficiencies, a dark side to larger ecosystems. Restricting entry to just several hundred developers\footnote{The flatness of the curve at its peak would make the assertion of a more precise estimate misleading.} software firms would have led to more active software development. The finding of sharp crowding among close substitutes further suggests that careful selection of partners and avoiding excessive rivalry could have further enhanced development.

In a manner, these findings also contribute to the vein of research considering how platforms should be “openened” to outsiders. Prior research has identified a growing number of modes and degrees of opening a platform including: selective revealing of software code (Henkel 2006), varying levels of integration and concentration of control (Farrell and Katz 2000; Gawer and Henderson 2007) and varying duration of property rights (Parker and Van Alstyne 2006). This paper points out that platform suppliers might also regulate the extent of access granted and resulting levels of entry. Consistent with the gist of most prior work in this area, the findings here show that intermediate “degrees” of openness might often be optimal.

The virtues of restricting entry, illustrated here, contrast with findings from research on software developer entry into more highly socialized, cooperative environments such as open source (von Krogh et al. 2003). In this case of these competing software developers, the bottom half of software firms contributed just 10% of software titles (also the least popular titles) and had an overall net negative effect on innovation. Because there appears to have been little \textit{ex ante} uncertainty regarding which firms were part of this competitive fringe, platform suppliers could have potentially excluded these firms from receiving a license. In contrast, “peripheral” contributors in open source software development have been found to contribute a large positive effect on innovation. This might be explained by features of open source such as heterogeneity of motivations, norms of cooperation, and clear differences in contributing to the same software project rather than multiple competing projects. I also speculate that the rites of passage to an open source project (von Krogh et al. 2003) may be
more selective than the rigors of competition—in this context with such low barriers to entry.

6 Conclusion

*When should a platform supplier allow or even promote the entry of complementors? To what extent?* This paper empirically analyzed the case of software firms around mobile computer platforms. In this case, it was widely believed that high levels of software developer entry led to the innovation of vast libraries of software programs and network effects. Prior research by Nair et al. (2004) established there was positive feedback between the available selection of software titles and market demand. This paper focused on the relationship between the available selection of software titles and the underlying software supply structure. Using point-of-sale data from the leading retailer allowed a relatively comprehensive sample of commercially available software to be assembled. This also allowed product selection to be distinguished from the underlying supply structure so the relationship between the two could be studied.

Patterns in the data reveal that repeated product development by incumbent software developers accounted for vastly more new software titles than did the entry of new software firms. Therefore, factors influencing incremental product development investment decisions should have had (and did have) the greatest impact on overall product development. Given that many product releases did not just serve new markets but were often close substitutes of existing products raised the possibility that entry might not just lead to a market-expanding network effect, but might also (or instead) lead to competitive crowding.

Simple correlations between numbers of software firms and numbers of available software titles or rates of new software title development in different platforms initially corroborated the intuition of “more developers, more software.” But this positive relationship was found to be exceedingly fragile to the introduction of even the most basic control variables. Once control variables were included in regressions, the relationship between numbers of software firms and rates of new product development turned negative at high levels of entry. An analytical framework was developed to argue that this slope is in fact an upwardly biased estimate of the true causal relationship and therefore the negative slope was interpreted as a negative causal effect. The results are consistent with growing entry leading competitive crowding to eventually overwhelm network effects at high levels of entry.

The product development and price-setting decisions of individual firms corroborate this interpretation. Individual firm product development also decreased with high levels of entry.
The negative response was found to relate just to the entry of other software firms who sold close substitute products, as both product development and price declined in response to entry by close substitutes, but did not for entry in the same software developer network as a whole. Thus, the patterns suggest that even in platform contexts with network effects that traditional concerns for maintaining investment incentives and dynamic efficiency remain paramount.

References


[52] Salop


Figure 1: The Generation of an Extraordinary Selection of Software Titles for Leading Platforms

Source: Estimates based on company press releases
Figure 2: Variety of Software Titles

Figure 3: “Long Tail of Demand”
Figure 4: More Developers, More Software?

Figure 5: More Developers, Faster Development?
Figure 6: Software Market Structure