Why Software Firms Build Hardware – And What Microsoft Is Doing About It

by

Ryan M. Shaffer

B.S. Electrical Engineering
Grove City College, 2008

M.S. Computer Science
Boston University, 2011

SUBMITTED TO THE SYSTEM DESIGN AND MANAGEMENT PROGRAM
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2015

Signature of Author: ____
System Design and Management Program
January 9, 2015

Certified by: ____
Michael Cusumano
Sloan Management Review Professor in Management
Thesis Supervisor

Accepted by: ____
Patrick Hale
Director, System Design and Management Fellows Program

Why Software Firms Build Hardware – And What Microsoft Is Doing About It

by

Ryan M. Shaffer

Submitted to the System Design and Management Program on January 9, 2015
in Partial Fulfillment of the Requirements for the Degree of Master of Science in
Engineering and Management

ABSTRACT

Many software companies build first-party hardware products due to the trend toward smaller, more highly-integrated devices, along with the fast pace of innovation in the technology industry. Building hardware products does not always lead to success and actually creates a financial risk for the company by significantly reducing profit margins as compared to the traditional profit margins to which large software companies are accustomed.

Three specific strategies are observed which firms have used successfully in this area. First, the “Hardware First” strategy is described, wherein a company builds devices with the primary goal of selling those devices bundled with the company’s software. Second, the “Proprietary Devices” strategy is presented, in which a company builds a device that is targeted at a particular market or function and locks in the customer to the firm’s ecosystem. This strategy has been observed to succeed in markets where the technology is not yet mature, as well as in cases where the device has a particular purpose that cannot be achieved as effectively with a general-purpose device. Third, the “Service Funnels” strategy is considered, wherein a firm builds hardware devices whose primary intent is to drive usage and revenue of its core software and services products.

Microsoft and its various hardware strategies over the years are especially considered, including products such as Xbox, Zune, Kin, and Surface, as well as its acquisition of Nokia’s devices business. Each of the three observed strategies has been used by Microsoft at various times, and analysis of these strategies is used to help explain why some products have succeeded while others have failed dramatically in the marketplace. Microsoft’s core capability is undoubtedly in software, and developing a mutually-beneficial relationship between its hardware and software products will be key to the long-term success of Microsoft in today’s technology landscape.

Thesis Supervisor: Michael Cusumano
Title: Sloan Management Review Professor in Management

Acknowledgements

I would like to extend my sincere thanks to Professor Michael Cusumano, whose advice and guidance as my thesis advisor has been invaluable. His course in Software and Internet Entrepreneurship at MIT Sloan School of Management was a source of inspiration for much of the thinking behind this paper, and his advice throughout the process of researching and writing this paper has shaped it into something of value.

I would like to acknowledge Microsoft, and especially to my team in the Office division, for allowing me the flexibility to participate in the System Design and Management program at MIT over the past two years while maintaining my job responsibilities and growing my career. This paper would not have been possible without the lessons learned through my professional experience at Microsoft over the past seven years. Likewise, this degree would not have been obtained without the consistent support of my managers, including Ryan Jansen, Gail Slapikoff, and Ed Fischer, as well as the rest of my team at Microsoft.

I also extend deep gratitude and love to my wife, Irene Shaffer, who has supported me through countless evenings and weekends of homework, projects, and research since I entered the SDM program two years ago. Her encouragement is the reason I originally applied to enter SDM, and I would not have been able to complete this degree without her.

My parents, Jerry and Lee Ann Shaffer, have also been a constant source of support, not only throughout this degree program, but throughout my life. Not only were they somehow able to raise me to become a well-balanced adult, but they encouraged my love of technology from an early age and devoted considerable time, energy, and basement space to ensuring that I was able to explore and learn as much as possible.

Thanks also to the many friends and colleagues I have gained throughout my time in the SDM program. The relationships formed throughout my time at MIT have been at least as valuable to both my personal and professional life as the academic side of the program. Special thanks go out to Pat Hale, who has made the SDM program what it is to all of us, and to Warren Seering, who served as a research advisor to me over the past year and has made the program that much more valuable for me.

Finally, I give thanks to my heavenly Father, not only for his endless love, but also for the many gifts and talents he has granted to me. In this work, and any other work that I do, my ultimate goal is to use those gifts to the best of my ability in order to bring him honor.

Table of Contents

Chapter 1: Introduction

If I look back with 20-20 hindsight, the thing I regret is that we didn't put the hardware and software together soon enough. It was almost magical the way the PC came about with an operating system from us and hardware from IBM. There was a little bit of magic too for Android and Samsung coming together. But if you really want to bring a vision to market, it's helpful to be able to conceive and deliver the hardware and software.

– Steve Ballmer, former Microsoft CEO (Cave 2014)

Devices are shrinking. This trend has existed practically since the invention of the computer, and the present era is no exception. Desktop computers are rapidly being supplanted in the market by smartphones and tablets, and still-smaller form factors like watches and other wearable devices are quickly gaining acceptance by consumers.

All of these devices need software. In the past, a clear division typically existed between hardware manufacturers and software firms. Companies like Dell or Hewlett-Packard would build assemble the devices, and companies like Microsoft or Oracle would build the software to run on the devices. But today, the lines are much fuzzier. The emergence in recent years of an extensive service fabric on the Internet – often referred to as the "cloud" – has reduced the need for processing power on individual devices, which has allowed them to become both smaller and cheaper. Software products are often a bundle of several features that includes integration with particular device features, a simple user interface, and a back-end in the cloud that is accessible anywhere the user has an Internet connection.

Most successful software companies have traditionally built only software, allowing their software to be run on as many devices as possible, in order to increase total sales and to maximize profits.

The strategy of a software company building first-party hardware has been relatively rare in the industry, but has become more common recently. The strategy was most famously pioneered by Apple in the consumer space, particularly with the launch of its Macintosh computers in 1984. It was then followed by many others, including Sun in the enterprise space, and then more recently by Microsoft in the consumer space with devices like Xbox and Surface. As coupling between hardware and software features increases, along with the pace of innovation, companies are increasingly deciding to build software and hardware products in tandem.

Afuah has suggested that vertical integration may be especially beneficial in the early life of new technologies (Afuah 2001). This builds upon the often-cited work by Henderson and Clark that explains why technological discontinuities, or "architectural innovation", often disrupt the business of established firms and can cause them to stumble or even fail completely (Henderson and Clark 1990). For a company whose core capability is building software, expanding into the hardware business may cause increased risk of such disruption.

So when does it make sense for a software company to also build first-party hardware devices? Further, how does this apply to Microsoft's strategy? This paper identifies three distinct cases where companies decide to build hardware for their own software:

1. "Hardware First": When the company's primary business goal and revenue stream is to sell hardware products. Examples of this strategy are discussed in Chapter 2.
2. "Proprietary Devices": When the company's software is proprietary or niche in nature and requires a deep level of integration with a similarly proprietary or niche hardware system to be fully functional. Examples of this strategy are discussed in Chapter 3.

3. “Service Funnels”: When both the company’s software and hardware products are simply channels through which to advertise and increase revenues in the company’s core business. Examples of this strategy are discussed in Chapter 4.

Chapter 5 deals with the issue of patents and licensing agreements in the hardware and software space and discusses how patent strategies play a role in the overall hardware strategies of software firms.

Microsoft has employed all three of the preceding strategies in various segments of its business, as we will see in detail in Chapter 6. Finally, we will summarize the arguments in the paper and draw conclusions and recommendations in Chapter 7.

Chapter 2: Hardware First

People who are really serious about software should make their own hardware.

– Alan Kay, Creative Think Seminar (Hertzfeld 1982)

2.1 Introduction

The “Hardware First” strategy is used by companies whose primary business goal is to sell hardware devices. These products are the main source of revenue for the company, and any software built by the company is solely meant to enhance and complement the overall experience of using its hardware products. Most companies that we would traditionally think of as hardware-centric – including a wide variety of firms like Sony, Sun, Apple, Samsung, or even newer firms like Fitbit or Pebble – would fall into this category.

This chapter takes a closer look at Apple, a company that exemplifies the “Hardware First” strategy. The same analysis is then performed for Microsoft and Google, two companies which are traditionally focused on software and services, but which acquired hardware divisions of other companies in no small part due to competitive pressure from Apple. A similar analysis is done for Oracle, another traditional software company that entered the hardware market via acquisition. Finally, we look at the case of IBM, which originated as a hardware company, but has since reinvented itself to become a firm focused on software and services.

2.2 Apple

Since its founding in 1976, Apple has traditionally been primarily a hardware company, and its primary products have nearly always been hardware products such as personal computing devices. It builds software and services only as required to support its hardware products.

Table 1. Revenues from Apple's software and services (Apple 2003-2013).

Fiscal Year	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Software + services as % of total revenue	10.96%	13.27%	14.28%	16.38%	16.29%	14.80%	15.03%	11.53%	8.56%	7.66%	9.39%
Total software + services revenue ($mm)	$680	$1,099	$1,990	$3,164	$4,004	$5,548	$6,447	$7,521	$9,268	$11,993	$16,051

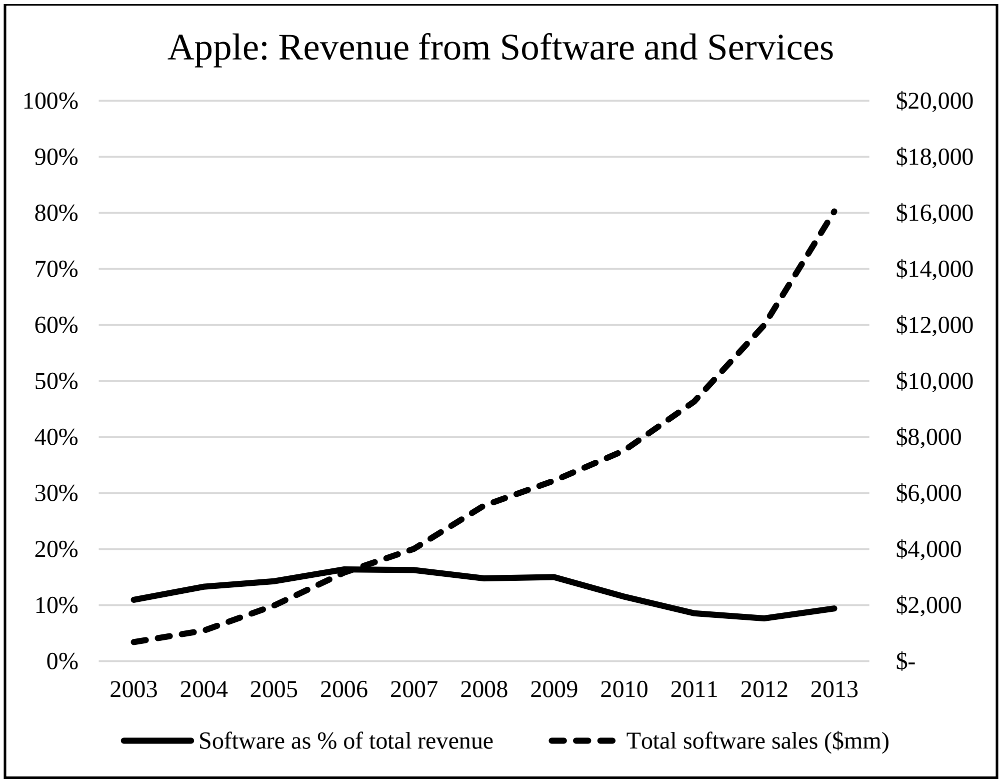

Figure 1. Revenues from Apple's software and services (Apple 2003-2013).

As can be seen from the chart in Figure 1 and accompanying data in Table 1, since 2003, software and services have accounted for approximately 10-15% of Apple's revenue[1]. This percentage has remained relatively consistent even though absolute revenue from software and services has skyrocketed by more than 1500% over that same period. It is worth noting that in Apple's case, this revenue is not "software" revenue per se, but is almost entirely what we would think of as "services" revenue, obtained primarily through iTunes music sales and App Store sales.

Apple is the most famous example of a company that uses the "Hardware First" strategy. As a company whose ultimate goal is to sell hardware devices, it is a logical conclusion that its software should run exclusively on its own hardware. But from Apple's history, we can see that this strategy has not always been effective in achieving market success. The Macintosh series of personal computing devices, despite initial success, eventually failed to beat Windows in the marketplace in large part because of this first-party exclusivity. Because the Macintosh operating system could not run on other devices, and because Apple's devices were typically more expensive and not as widely adopted as its competitors' models, it eventually faded into obscurity.

Apple has followed a similar strategy for its mobile operating system, iOS, but with much greater initial success.

The notable exception to Apple's first-party exclusivity strategy is iTunes. This software product, which allowed iPod users to purchase and manage their music, was originally developed only for

[1] From 2003-2010, the "software and services" revenue figure is estimated by adding together the revenues reported under "other music related products and services" and "software, service and other sales" from Apple's annual financial reports. From 2011-2013, the "software and services" revenue figure is estimated by using the "iTunes, software and services" category, which should give us a slightly better estimate than for earlier years. While neither is an exact measurement of revenue from software and services, it is reasonable and sufficient for our purposes.

the Macintosh operating system (released in 2001). Because the market share of Macintosh computers was so low in comparison to Windows PCs, this placed a severe limitation on the potential market for the iPod itself. Recognizing that the iPod had much greater market potential than the Macintosh at that time, Apple decided in 2003 to release iTunes for Windows. As can be seen from the chart in Figure 1, this was the point at which software and services revenue (which includes iTunes) began to skyrocket. Because iTunes was software designed to exclusively support an Apple device, and not a product in itself, Apple chose to expand the potential user base of this software rather than arbitrarily limit it to users of the Macintosh, which was a product mostly unrelated to iPod.

2.3 Nokia and Microsoft

Microsoft, as the company's name implies, has traditionally been a software company. It has primarily built hardware only for specific purposes that support the goals of its software business. The chart in Figure 2 and accompanying data in Table 2 shows a hypothetical combination of Microsoft and Nokia Devices revenue from 2003 through 2014[2]. The figures were obtained by adding the revenues of Microsoft and Nokia's devices business over that timeframe. Between 2003 and 2007, software and services accounted for just 50-55% of this hypothetical entity's total revenue – Xbox was selling well, but more importantly, Nokia's device business was booming.

[2] From FY2003 through FY2013, the "total revenues" figure is estimated by taking the overall reported revenues of Microsoft and adding the revenues of the Smart Devices and Mobile Phones division of Nokia for those years. The "total software sales" figure is estimated by taking the sum of the Windows, Server and Tools, Online Services, Office, and Corporate and Other divisions of Microsoft for those years. For FY2014, Microsoft changed its financial reporting structure, and no clear annual total was available for Nokia's devices unit, as Microsoft completed its acquisition mid-year. The numbers for FY2014 are approximated by using the sum of Consumer Licensing, Commercial Licensing, Commercial Other, and Corporate and Other as the "total software sales" figure. The approximate annual revenue of $8,715mm for the Nokia Devices unit is derived from Microsoft's assertion on page 83 of its 2014 10-K that its total revenue would have been $3,005mm greater in 2014 than 2013 had its Nokia acquisition consummated two years earlier. That assertion, together with the Nokia Devices FY2013 revenue of $12,709, allows for arrival at the $8,715mm number – which, by the way, represents a steep 31.4% decline from FY2013.

Table 2. Revenues from Microsoft/Nokia entity (Microsoft 2003-2014) (Nokia 2003-2013).

Fiscal Year	**2003**	**2004**	**2005**	**2006**	**2007**	**2008**
Software as % of total revenue	53.01%	56.33%	53.83%	50.63%	50.64%	54.66%
Total software sales ($mm)	$29,255	$33,726	$36,303	$39,474	$44,986	$52,207
Total revenues ($mm)	$55,183	$59,871	$67,441	$77,966	$88,827	$95,519

Fiscal Year	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**
Software as % of total revenue	60.29%	61.79%	65.86%	72.82%	74.69%	72.64%
Total software sales ($mm)	$52,021	$56,260	$61,047	$64,133	$67,636	$67,962
Total revenues ($mm)	$86,290	$91,054	$92,693	$88,071	$90,558	$93,563

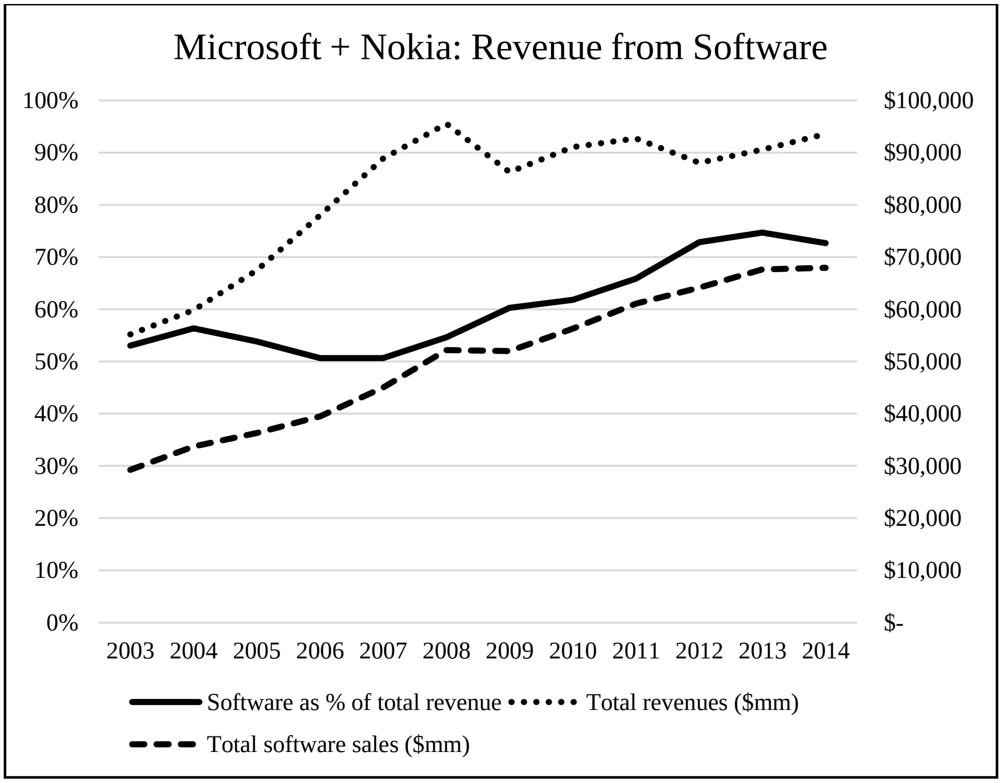

Figure 2. Revenues from Microsoft/Nokia entity (Microsoft 2003-2014) (Nokia 2003-2013).

The iPhone was released in 2007, and this coincided with the beginning of the steep decline for Nokia's devices unit. The enormous increases in the software revenue of this combined entity were almost completely offset by the decrease in hardware revenue over the following seven years. (In fact, as can be seen from the data, the hypothetical combined revenue total of $95.5 billion in FY2008 has been the highest figure to date.)

Former CEO Ballmer had the following to say about the significance of Microsoft's decisions to enter the hardware market with various devices:

> *It's important because the name of the company is "Micro-soft". As a fundamental part of the founding principles, we were a software company and yet with Xbox, then Surface and now the phones, essentially we have a profile that will wind up being far more mixed in the future. That's a pretty fundamental change in the way that we self-identify, think about and express the value we add and our innovation. (Cave 2014)*

So we can see that a fundamental difference between Microsoft and Apple is that Microsoft always has been, and likely always will be, a software-first company. In order to be consistent with this strategy, any hardware that Microsoft builds must be primarily intended to promote its software products and platforms. This strategy is exemplified by Microsoft's decision to kill off Nokia's Android hardware line, along with significantly reducing its workforce and infrastructure footprint, shortly after completing the acquisition (Warren 2014).

2.4 Motorola and Google

When Google acquired Motorola Mobility in August 2011, they gained control of the entire software and hardware stack of Motorola's Android devices. Google kept Motorola at arm's length, spun off pieces of it, and eventually sold it to Lenovo in January 2014 (see Section 5.2), and so it never really took advantage of its position to fully integrate its hardware and software operations, as Microsoft has clearly tried to do with Nokia's hardware business.

Nevertheless, it is interesting to take a hypothetical look at the financial performance of a combined Google/Motorola entity[3], as seen in Figure 3 and its accompanying data in Table 3, and compare it to the results we have seen previously with Apple and with the combination of Nokia and Microsoft. Because Google does not split out its own hardware revenues, which have traditionally been quite small compared with that of Motorola, we will analyze the percentage of the combined entity's revenue which came from Google as a whole vs. that which came from Motorola.

Like Microsoft, Google has traditionally been a software-focused company. But unlike Microsoft, Google has never made a significant portion of its revenue from actually selling its software, and it has certainly never been in the business of selling boxed software like Microsoft. Instead, it makes most of its money by selling advertisements to be displayed to the users of its free web services. And because its products are primarily hosted services, it has invested significantly in datacenter technology and infrastructure; its financial reports indicate that it is now spending over $2 billion quarterly on its hardware infrastructure. Nonetheless, this hardware investment accrues to its services revenue and does not actually lead to revenue from device sales.

[3] For FY2003 through FY2011, the "Google revenue" listed is the actual total revenues of Google, and "total revenues" is the sum of Google's revenue and the Mobile Devices line item on Motorola's annual reports. For FY2012 and FY2013, "Google revenue" represents the total revenues of Google excluding the Motorola hardware line item on their annual reports, and "total revenues" is the actual total revenues of the combined Google/Motorola entity.

Table 3. Revenues from Google/Motorola entity (Google 2003-2013) (Motorola 2003-2013).

Fiscal Year	**2003**	**2004**	**2005**	**2006**	**2007**	**2008**
Google as % of total revenue	11.54%	15.71%	22.24%	27.20%	46.64%	64.30%
Total Google revenue ($mm)	$1,466	$3,189	$6,139	$10,605	$16,594	$21,796
Total revenues ($mm)	$12,704	$20,297	$27,598	$38,988	$35,582	$33,895

Fiscal Year	**2009**	**2010**	**2011**	**2012**	**2013**
Google as % of total revenue	76.80%	78.95%	86.38%	91.76%	92.81%
Total Google revenue ($mm)	$23,651	$29,321	$37,905	$46,039	$55,550
Total revenues ($mm)	$30,797	$37,140	$43,883	$50,175	$59,856

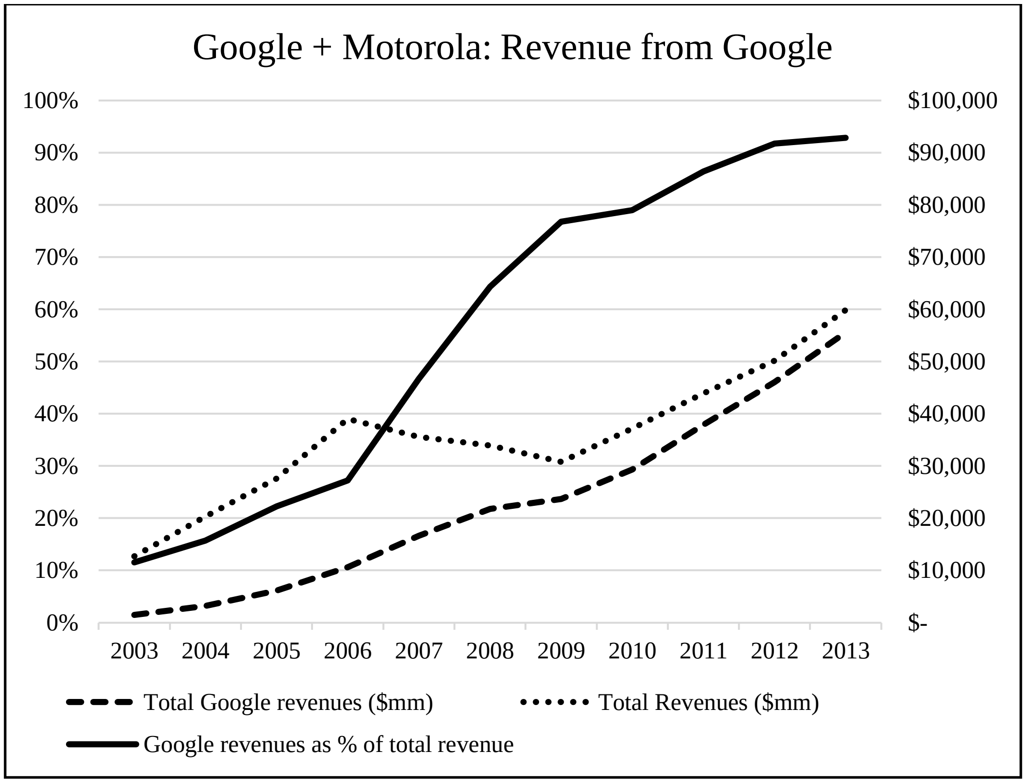

Figure 3. Revenues from Google/Motorola entity (Google 2003-2013) (Motorola 2003-2013).

Also unlike Microsoft, Google’s revenues were relatively low in the early 2000s, so when looking at the hypothetical combined entity of Google and Motorola Mobility, the devices revenue far outweighs the software and services revenue generated by Google’s products for those years.

This highlights one major difference between the Google/Motorola marriage and that of Microsoft/Nokia: namely, that Microsoft was a much older, more diversified company than Google at the time of their respective hardware acquisitions. Microsoft has a long history of acquiring and integrating large companies into its core business, whereas such occurrences are rarer with Google. Therefore, it should not be surprising that Google kept Motorola separate from its core business and eventually spun it off.

As with Microsoft, we can conclude that Google seeks to build hardware primarily in cases where it promotes its own software and services. Because Google’s consumer-oriented services are primarily web-based and device-agnostic, it makes sense that it would be less invested in the hardware business than a company like Microsoft, which has traditionally built OS software and other experiences more tightly integrated with hardware. The exception, of course, is Android – and this is an area where Google has continued to pursue its own hardware to some extent with the Nexus line of devices.

2.5 Oracle and Sun

In 2010, Oracle completed its acquisition of Sun Microsystems. For the first time in its history, Oracle was responsible for a large lineup of hardware products, in addition to its traditional business of database software. As in the previous sections, we can perform a financial analysis on a hypothetical combined entity of Oracle and Sun over the past decade, as shown in Figure 4 and the accompanying data in Table 4.

Table 4. Revenues from Oracle/Sun entity (Oracle 2003-2014) (Sun Microsystems 2003-2009).

Fiscal Year	**2003**	**2004**	**2005**	**2006**	**2007**	**2008**
Software as % of total revenue	62.7%	65.5%	68.8%	69.5%	72.5%	76.3%
Total software sales ($mm)	$13,116	$13,986	$15,743	$19,077	$23,098	$27,692
Total revenues ($mm)	$20,909	$21,341	$22,869	$27,448	$31,869	$36,310

Fiscal Year	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**
Software as % of total revenue	80.7%	80.1%	80.5%	83.0%	85.6%	86.0%
Total software sales ($mm)	$27,997	$27,509	$28,678	$30,819	$31,834	$32,903
Total revenues ($mm)	$34,701	$34,333	$35,622	$37,121	$37,180	$38,275

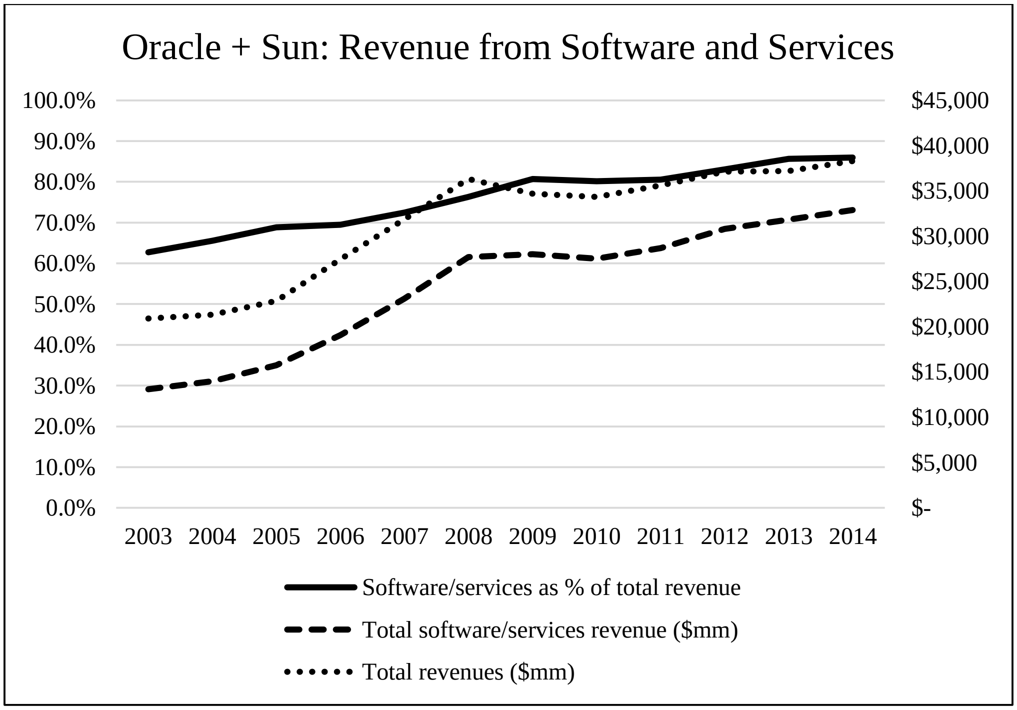

Figure 4. Revenues from Oracle/Sun entity (Oracle 2003-2014) (Sun Microsystems 2003-2009).

The analysis reveals that the software and services revenue[4] of the combined entity has been steadily and consistently increasing as a fraction of total revenue since 2003. This is similar to the trend we saw with both the Google and Microsoft cases, where software and services revenues are increasing at a faster rate than overall revenue, a trend with the implication that hardware revenue is remaining fairly stagnant. Sun was clearly in decline at the time Oracle acquired it. When looking at this fact in combination with the recognition that Oracle is a software company at its roots, we can see that Oracle has used the hardware line it acquired mainly as a complement to its software and services products, rather than treating it as a primary source of revenue on its own. Indeed, combined revenue from Oracle's software and services businesses increased by 19.6% from 2010 through 2014, while its total hardware revenue has actually decreased by 21.3% over that same span (Oracle 2003-2014).

2.6 IBM

Established in 1911 as the Computing-Tabulating-Recording company, IBM for most of its history has been a hardware firm that developed software as necessary to support its hardware products. In this sense, it was similar to Apple – it was an early example of the "Hardware First" strategy being used with great success.

However, beginning in the 1990s, financial struggles began to envelop many of the company's hardware businesses that had been successful for so long. IBM began to divest various pieces of its business, including the spinoff of its Lexmark printer business in 1993, the sale of its personal

[4] For fiscal years 2003 through 2009, the total software and services revenue was calculated by adding the Software and Services line items from Oracle's financial reports, along with the Support Services and Professional/Educational Services line items from Sun's financial reports. For fiscal year 2010, the numbers from Sun were estimated based on similar numbers from 2009 and 2011, since there was no form 10-K filed with the SEC due to the acquisition. For fiscal years 2011 through 2014, the revenues listed were simply the sum of the Software and Services items from Oracle's 10-K reports.

computer business to Lenovo in 2005, and the sale of its x86 server division to Lenovo in 2014. As a result, the company began to increasingly focus on software and services as its main source of revenue. The dramatic expansion from a "Hardware First" strategy is demonstrated in a 2002 press release announcing the acquisition of PricewaterhouseCoopers' global consulting and technology services unit:

> *The combination of IBM and PwC Consulting will create a powerful, unmatched capability to help clients solve their business issues, exploiting world-class technology for improved business performance. IBM Global Services integrates a broad range of capabilities – services, consulting, hardware, software and research – to help companies of all sizes realize the full value of information technology. (IBM 2002)*

To compare IBM to the cases of Apple, Google, Microsoft, and Oracle that we have looked at previously, we analyze its financial performance since completing the acquisition of PwC Consulting in 2002, particularly to determine the relative importance of its software and services business[5] to the company as a whole. The financial trends are presented as a chart in Figure 5, and the accompanying data are shown in raw form in Table 5.

The result here looks surprisingly similar to that of Oracle and Sun. Total software and services revenue increased 45.5% from 2003 through 2013, while hardware revenue decreased by 49.1% over that same time period (IBM 2003-2013).

[5] From 2003 through 2006, the software and services revenue is calculated by adding the Global Services and Software line items on IBM's annual 10-K financial reports. From 2007 through 2013, the software and services revenue is computed by taking the sum of the Global Technology Services, Global Business Services, Software, and Services line items from the 10-K annual reports.

Table 5. Revenues from IBM's software and services (IBM 2003-2013).

Fiscal Year	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Software + services as % of total sales	63.9%	64.8%	70.5%	72.7%	75.7%	78.8%	80.5%	79.6%	80.1%	81.1%	83.5%
Total software + services sales ($mm)	$56,946	$62,424	$64,237	$66,451	$74,125	$80,981	$76,396	$78,909	$85,107	$84,251	$82,877
Total revenues ($mm)	$89,131	$96,293	$91,134	$91,423	$97,943	$102,827	$94,889	$99,120	$106,195	$103,930	$99,270

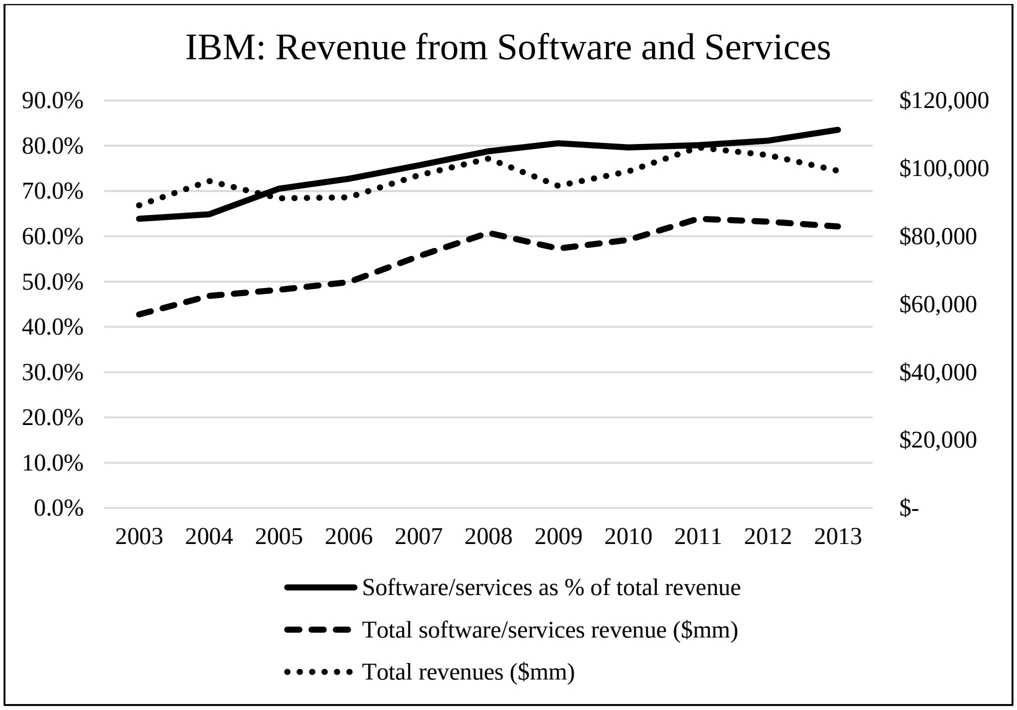

Figure 5. Revenues from IBM's software and services (IBM 2003-2013).

No matter how the facts are analyzed, it is clear that IBM has fully transitioned away from being a “Hardware First” company into a firm fully focused on providing software and services. Of course, IBM still has a sizable hardware business with annual revenues of over $14 billion, but this now serves as a complement to IBM’s larger goal of providing end-to-end technology solutions for its customers.

2.7 Summary

While a hardware-focused company like Apple produces up to 90% of its revenue from sale of hardware devices, software and services companies like Microsoft or Google typically make only a small fraction of their revenue from hardware sales – even if we analyze them as if they and one of their large OEMs were a single, unified firm. Companies like Oracle and IBM, although they have significant first-party hardware business, derive a large and growing portion of their revenues from software and services, and their hardware offerings simply serve as a complement to their primary business goals. By nature, companies like Apple design their software to drive sales of their own devices, which necessarily results in software sales being simply a small percentage of hardware sales. But software-focused firms like Microsoft and Google design their software to be accessible from a multitude of devices, and so revenues from in-house hardware are not tied to software sales in the same way. Likewise, services-focused firms like Oracle and IBM often design both their software and hardware products to serve as a conduit to increase revenues in their growing services businesses, rather than as ends in themselves.

Chapter 3: Proprietary Devices

One of the major benefits of specialization is to allow designers to place significant constraints on the system's interface. Specialized interfaces can provide direct links between the actions that users want to carry out in the task and in the interface.

– James Miller and Donald Norman (1986)

3.1 Introduction

Not all hardware devices can be built on open platforms or serve as general-purpose computers. Miller and Norman (1986) discuss the pitfalls of building purely general-purpose devices. They argue that specialization of interfaces allows the task semantics to drive the design of the interaction model, which results in a much more appropriate and intuitive way of working with the system:

> *The alternative approach [to general-purpose tools] is to develop highly specialized systems, each of which addresses one and only one task. Specialized solutions are apt to be more efficient and easier to learn than generalized ones, and will tend to be more powerful and useful for their specific application. With such systems, average users need not be aware of the computer or its interface; they need only be aware of and expert in the application. (Miller and Norman 1986)*

Despite this, the most profitable software has traditionally been the most general-purpose software, since it by nature has the largest potential market, both in terms of hardware to run it and customers to use it.

In a few situations, it makes sense for a company to build a proprietary, specialized hardware/software solution that is geared toward a specific purpose. First, in a category where general-purpose solutions are not yet widely available, like smart thermostats or fitness bands, a company may be able to grab a significant portion of the market by creating a proprietary solution that is more functional and/or more appealing than its competitors. Second, and perhaps more relevant to this conversation, software that is geared to a very specialized purpose, such as playing music or reading books, might be better suited to run on a proprietary device especially designed for that purpose. Hardware for general-purpose or platform-agnostic software is usually commoditized and is not typically very profitable, especially for a software company that is accustomed to very high operating margins.

This chapter looks at examples of several companies who have built proprietary devices over the years, starting with the early examples of Sun and IBM, and moving on to more recent examples like Oracle, Google, Apple, Microsoft, and Amazon. The chapter concludes with a look at some first-party, proprietary hardware devices that have failed in large part due to the fact that they were general-purpose devices consisting of primarily commoditized hardware and without enough differentiation to allow them to succeed in the market.

3.2 Sun workstations

Sun Microsystems traditionally developed the software and hardware for its workstation systems in tandem. The hardware itself was a commodity. Other workstation manufacturers like DEC, IBM, and HP typically had access to the same suppliers and built essentially similar machines with similar performance at similar price points. Sun viewed their software, and the way in which it integrated and took advantage of their hardware, as their differentiator (Hall and Barry, 29). Therefore, their competitive advantage in the marketplace was derived primarily from the efficiency of its software, not necessarily the power of its hardware (Hall and Barry, 52). Later, as their market share and penetration began to increase, network effects started to take hold, as many third-party software vendors began to port their software over to Sun’s operating system. On occasion, Sun would pay the software vendors for this task in order to further increase its attractiveness in the marketplace (Hall and Barry, 34).

Sun’s strategy for building workstations was initially unique among its competitors. Rather than building a closed, proprietary bundle consisting of both hardware and software, Sun chose to use an open systems concept for its hardware development. In Sun’s view, this was the path to becoming a long-term player in the workstations market; they believed that developing a closed, proprietary system led to inevitable obsolescence (Hall and Barry, 26).

When Sun began to develop workstations based on its own Scalable Processor Architecture, or SPARC, it worked to rapidly replace the older Motorola x86-based systems in the marketplace with models that were SPARC-based (Hall and Barry, 237). Because Sun owned the technology underlying SPARC, this meant higher margins because it did not need to license an instruction set architecture from another company. In addition, their workstations ran Solaris, a Unix-based operating system built in-house. By tying their success to proprietary SPARC platform and their proprietary Solaris software, rather than the Intel x86 architecture which was rapidly becoming an industry standard, Sun effectively made their systems even more proprietary, which made them highly vulnerable to changes in the market.

Over time, it became clear even to Sun that the SPARC platform had no future, and so in the early 2000s they tried to move back to developing the open systems concept that had been so successful for them initially. They began producing x86-based systems in 2003, using lower-cost chips from AMD at first in an attempt to maintain both healthy margins and competitive prices. Shortly thereafter, they began to open-source their key proprietary software products – starting in 2005 with their Solaris operating system, and following up with their flagship Java product in 2006, both of which had been closed and proprietary to that point. But these moves were simply a desperate reaction to the firmly established market trend toward x86 and open-source software; in 2003, when the transition away from its proprietary SPARC-based systems began, Sun had reported an annual loss of $2.4 billion (Brodkin 2009). Sun's failure, which eventually concluded with its acquisition by Oracle in 2010, is evidence that proprietary devices are hard-pressed to succeed in a mature market with well-established platforms and standards, which the server and workstation market had clearly become by the early 2000s.

3.3 IBM mainframes

Since the 1950s, IBM's largely proprietary mainframe products have been at the heart of its hardware business. Starting with the System/360 line of mainframes released in 1964, IBM built and shipped operating system software customized for its mainframe hardware. Even though these mainframes might be considered commodity devices by today's standard of computing, in the 1960s both the hardware and software were still in their infancy, and thus required an extraordinarily high degree of coupling between the hardware and the operating system. It would have been impractical for any company to develop an operating system without also developing the hardware on which it was designed to run.

The architecture of the System/360 and subsequent models of IBM mainframes was truly proprietary, having been developed entirely in-house. But over time, it became so established in the market that it effectively became the industry standard. An entire ecosystem of "plug-compatible mainframes", or PCMs, emerged from competitors like Fujitsu and Amdahl who built their own units by copying the IBM architecture. This created a unique danger for IBM in that its software, which had been provided for free to run on its System/360 mainframes, could also be run on these PCMs from its competitors (Takahashi 2005).

IBM’s modern System z mainframe line, introduced in 2000, effectively ended the era of PCMs. This line is targeted at highly specialized applications, such as big data processing, private cloud management, and transaction processing, rather than serving as general-purpose web servers or storage endpoints (IBM 2014). The tightly coupled nature of the whole hardware and software system adds value and allows the System z line of mainframes to be successful despite the ever-increasing commoditization of datacenter hardware. However, partly because of this commoditization, the bulk of IBM’s revenue does not come from the sale of mainframes themselves, which accounts for only about 4% of its overall revenue. Instead, IBM profits by supplying software and professional services that tightly integrate with and enhance its proprietary mainframe hardware systems. This aspect of IBM’s mainframe business is discussed further as an example of the “Service Funnels” strategy in Section 4.5.

3.4 Back-office appliances

As with mainframes, there are many enterprise back-office applications which perform much better on hardware that is dedicated to running a particular piece of software. Some companies have taken advantage of this opportunity to offer a fully integrated hardware/software solution, often called an appliance, which is dedicated to a particular function or small set of functions. Common examples of this include the Google Search Appliance (Google 2014) and the Oracle Database Appliance (Oracle 2014).

While these devices appear to be mainly the vendor’s specialized software combined with mostly commodity hardware, they have a place in the market for a couple of reasons. First, the software is extremely complex and configuration on a company’s own server hardware would be a difficult task for most IT personnel. Second, a company is likely to need a very small quantity of these machines, making the investment in IT knowledge unnecessary and unprofitable. For these companies, a standardized solution that is plug-and-play is much more economical. This means that vendors like Oracle and Google, much like technology consulting firms, can charge more for the proprietary hardware/software solution than the sum of the components’ individual costs.

3.5 Apple iPod and iPhone

The iPod is perhaps the quintessential example of a proprietary device with tightly coupled hardware and software that was extraordinarily successful in the market. According to public filings, the company sold nearly 400 million iPod devices between FY2002 and FY2014, including more than 200 million over a four-year span from FY2007 to FY2010 – and this is not even including the sales of the iPhone, which is essentially a successor to the iPod and has begun to replace it in the marketplace. The iPod’s success is even more stunning when compared to the glut of portable music players that arrived on the market around the same time. A key to Apple’s success here was their development of a true end-to-end software system to support the iPod, and not just the software on the device itself. Competitors’ devices had equivalent or superior functionality in many cases, but their downfall was a failure to consider the whole system, especially including the software used to download the music and transfer it to the device.

Apple was certainly not the first manufacturer to develop a portable music player – in fact, the iPod’s release in late 2001 came a full three years after devices like Diamond’s Rio and the

Compaq-developed Personal Jukebox entered the marketplace. But Apple’s fully-integrated system of iPod and iTunes, while often criticized for being “closed”, was precisely what the market required due to the immaturity of both the hardware and software ecosystems for music players.

These same factors played a large part in the success of the iPhone when it was released a few years later. Indeed, the original iPhone was not much more than an enhanced version of iPod with telephony features. But with the introduction of the App Store for third-party applications, Apple continued the strategy that had been so successful in the iPod/iTunes world by providing a unified marketplace for content (previously music, and now apps) and a continuous end-to-end software experience via both iTunes on the computer and the App Store interface on the device. The software and services system was entirely closed, and yet much like iTunes was open to any content creator, the marketplace of apps was open to any third-party developer. In this way, Apple was able to capitalize on the immaturity of the smartphone market by providing a complete, extensible experience that was superior in nearly every way to its competitors.

3.6 Game consoles

The game console is another type of proprietary device which has proven to be extremely lucrative, and which has been traditionally dominated by companies producing integrated solutions consisting of both software and hardware. Sony, Microsoft, and Nintendo have produced several generations of successful consoles by producing proprietary systems where the hardware and software have been developed in tandem. The volume of game console sales since the early 1990s is remarkable – Sony has sold over 300 million PlayStation consoles, Nintendo has delivered over 200 million of its systems, and Microsoft has shipped over 100 million Xbox units. Meanwhile, there has been very little competition from non-proprietary devices – PC gaming has maintained a presence in the market, but has never approached the levels of popularity or revenue that console gaming has achieved.

Two key aspects of these devices make it beneficial for a firm to control both the hardware and software development, and therefore make proprietary devices more attractive in the marketplace. First, the end-to-end user experience is even more crucial on these devices than on a standard PC, because there is much less configurability. It needs to “just work”. A company can have more control over the entire experience by developing the hardware and software in tandem. Second, the software written on top of these devices is typically limited in nature, and can easily be written to run in a sandboxed or restricted environment. Originally games were the primary software written for these devices, though more recently, Internet video streaming apps have become a primary use case. In both cases, the applications are focused on consumption and entertainment, and they can use a combination of specialized input devices like game controllers, motion sensors, or voice control to accomplish their goals.

3.7 Amazon Kindle

The e-reader category is another example of a largely proprietary device category, and the Kindle from Amazon is the most successful example. Since 2007, Amazon has released seven generations of Kindle devices with e-ink displays, in addition to its line of Android-based Kindle tablets. By some estimates, Amazon generates approximately $4 billion in annual revenue from sales of these devices (Trefis 2014). This figure excludes sales of e-books and other digital media, which we will discuss further in Section 4.2 as an example of the “Service Funnels” strategy.

The Kindle has been successful by offering an all-in-one e-reader solution which, especially in the early days, was at a price point far below the cost of a general-purpose device with equivalent functionality, such as a small laptop or a tablet computer. In addition, its display and user interaction model were far superior for the activity of reading than a general-purpose device could have been, thus lending credence to the ideas of Miller and Norman described in the introduction to this chapter.

3.8 Failures of proprietary devices

Proprietary devices are not always successful, despite the many examples listed in this chapter. General-purpose computing devices are part of a highly competitive and highly commoditized market where it is very hard to gain a significant market share or to generate a large operating margin, and when a company attempts to build a proprietary general-purpose device, it often struggles to maintain the necessary differentiation to succeed.

Apple’s dramatic success with its proprietary iPod and iPhone devices were in contrast to many of its earlier hardware manufacturing efforts. With its original Macintosh computer, Apple produced a tightly-integrated, proprietary system of software and hardware that was, by most accounts, technically superior to its competition. But its prices were high by comparison, and as the personal computer market became more crowded and began to standardize around the IBM PC, Apple was left with an incompatible system that never gained enough market share to maintain its success. When IBM branched out to produce its own software and hardware combination using its OS/2 operating system, it achieved even less success.

Even general-purpose software can occasionally become commoditized. This is particularly the case with the Android operating system produced by Google. Because the operating system is free and open-source, hundreds of device manufacturers have released various types of devices running Android, and several vendors have even created their own “forked” versions of the Android operating system for use on their own devices. When Google acquired the smartphone business of Motorola Mobility, many thought that it could use its position to build an Apple-like integrated hardware/software package that was better than what other Android handset manufacturers could produce. But in reality, due to the nature of the OS, Google had very little opportunity for competitive advantage because it was simply working with a combination of commodity software (Android) and what had become by that time commodity hardware (smartphones).

Another interesting failure is Microsoft's first attempt at a first-party proprietary tablet computer, the Surface RT, which was released in 2012. Its problems have been well-documented, including lack of developer and user ecosystems and failure to gain any significant market share or associated network effects. But perhaps a significant reason for these failures was the fact that Surface RT, while purportedly developed to compete with the proprietary Apple iPad, actually failed to differentiate itself in any significant way. It was marketed as a general-purpose device that would offer more flexibility and customization than the iPad, when in reality, it was the same type of proprietary device with a much smaller ecosystem and similarly small user base[6].

The one supposed point of differentiation for the Surface RT was its ability to run the new RT version of the Microsoft Office suite. But poor execution combined with poor marketing led to consumer confusion around compatibility. Indeed, Office RT did not even support the multitude of third-party plug-ins that had been developed over the years and that are core to Office's value proposition. So because its ecosystem gave it no advantage, it essentially placed itself in

[6] Because the Surface RT and its included Windows RT operating system could not run any of the legacy Windows applications that had been written over the decades, it was essentially starting from zero in terms of developer ecosystem, third-party application support, and installed user base.

competition with commodity tablets and laptops, and its limited functionality and performance gave it no chance of success in that market. The Surface is discussed in more depth in Section 6.5.

3.9 Summary

For software companies that want to produce a fully-integrated, proprietary, first-party hardware solution, it is critical to consider the specialization of the device, and whether it meets a need that cannot be met by combinations of existing commodity hardware and software. As we have discussed in this chapter, several categories of devices are good candidates for this type of solution, including mainframe computers, back-office appliances, portable music players, video game consoles, and e-readers. But more commoditized devices, including personal computers, Android smartphones, or even general-purpose tablets, are very difficult to differentiate without some specific added value or specialization that cannot be produced without the tight coupling enabled by first-party integration of software and hardware.

Chapter 4: Service Funnels

What we are doing is offering premium products at non-premium prices. [Other tablet makers] have not been competitive on price [and] have just sold a piece of hardware. We don't think of the Kindle Fire as a tablet. We think of it as a service.

– Jeff Bezos, Amazon CEO (Stone 2011)

4.1 Introduction

As we discussed in Chapter 2, hardware companies and software companies have significantly different revenue proportions between their respective hardware and software businesses. This leads to vastly different motivations when building a consumer hardware device. For a hardware company like Apple, selling the device is an end in itself. Apple wants as many people as possible to own an iPhone, an iPad, and even a MacBook. Of course, Apple has additional revenue streams post-purchase that it also seeks to maximize, but these are minimal compared to the sale of the hardware itself.

For companies that specialize in software and services, another strategy has become prominent, with some especially high-profile examples in recent years. These companies produce devices that funnel customers toward their software and services, with the ultimate goal of driving licensing and subscription revenue for their core products. Of course, the devices themselves are a revenue stream that must be managed and grown, but hardware revenue is not the end goal. Rather, the primary objective is to drive software and services revenue by acquiring users and maintaining mindshare of the customer. In this chapter, we will discuss a few examples of this in more depth, beginning with Amazon and its Kindle and Fire series of devices, and also touching on other companies' efforts in the smartphone and game console markets. We will also look at how IBM's mainframe business drives software and services revenues in a similar way.

4.2 Amazon Kindle and Fire

Amazon's Kindle, which we previously discussed through the "Proprietary Devices" lens in Section 3.7, is also one of the prime examples of a device that was developed with the primary purpose of promoting the company's own services – in this case, Amazon's Kindle library. The clear strategy with Kindle has been to sell the devices at very small margins with the primary goal of developing an engaged user base and driving revenue of Kindle e-books, which like other types of electronic goods, have an incremental production cost of essentially zero.

This strategy is even clearer with the more recent Fire series of devices, including the Kindle Fire tablets and the Fire Phone. From a hardware perspective, these devices are very similar to other Android devices and are essentially commodities. Even from a software perspective, there is nothing that functionally differentiates these devices from Apple's devices or even other Android devices. But Amazon has been very transparent about the fact that these devices are intended as funnels to push customers to its services – not only the e-commerce destination that is Amazon.com, but also its growing Prime offerings of digital movies, music, and other media. And because these services are indeed desirable to many users, Amazon has an opportunity to succeed in hardware if it is able promote its services in a way that make them more accessible, engaging, and compelling on its own devices than on similar competitive devices.

4.3 Google Nexus and Android One

In early 2010, Google released the Nexus One smartphone, the first of a long series of Nexus-branded handsets and tablets. For the most part, these devices have failed to differentiate themselves functionally or technologically from their competitors, since the hardware is mostly commoditized, and the software is simply stock Android. If we look at these devices through the lens of the “Proprietary Devices” of Chapter 3, they fail miserably.

But Google did not develop Nexus to fill a gap in the market that could not be served by existing devices. Instead, Google intended Nexus as a way to promote the Android ecosystem and drive customers to its own services. In a 2012 interview, Larry Page explained that one reason for Nexus “is that we want to build an amazing device that kind of showcases what’s possible on Android, gives a way for the programmers to get early builds, [and] does a whole bunch of things that are important” (Helft 2012). The goal is not to sell the hardware. The goal is to promote the ecosystem, which will in turn drive software, services, and advertising revenue for Google.

Google launched its Android One line of smartphones in 2014, priced around $100 and aimed at emerging markets. The hope is that by designing both the hardware and the software experiences end-to-end, Google can provide a more consistent and fluid experience than the existing low-end Android devices. Some believe that such a strategy also allows Google to protect its brand by maintaining more control over the Android handset experience than if it relied solely on OEMs (Krishna and Winkler 2014). One major piece of this is driving customers to Google’s own default services, including Google search, Gmail, YouTube, and so on – and keeping them engaged once they are there. By controlling the end-to-end experience of both hardware and software, Google may be attempting to optimize the experience for users of its services in a way that they believe other device manufacturers would be unable to do at such a low price point.

4.4 Game consoles

In Section 3.6, we discussed the game console, such as Microsoft's Xbox and Sony's PlayStation, as a type of proprietary device that is tailored for a specific use case. But we can also view these devices through the lens of the "Service Funnels" strategy. Revenues from game console units, while significant, do not produce large margins – the consoles are typically sold at very close to production costs, or even at a slight loss.

As we saw with the Kindle, the ultimate goal for the console manufacturers here is not necessarily device sales, but rather sales of consumable goods to be used on the device – in this case, video games, or more recently, subscriptions to online entertainment services like Xbox Live or PlayStation Network. Because video games and online subscriptions have an extremely low incremental cost per unit sold, the profit margins are extremely high, as is typical with software products. As evidence of this, Microsoft reported in 2013 that over 50% of the revenue from its Xbox group, or around $1.4 billion, came from Xbox Live subscriptions and related transactions (Wilhelm 2013).

4.5 IBM mainframes

Another somewhat unintuitive example of the "Service Funnels" strategy is IBM's mainframe business, which we discussed in Section 3.3 as an example of a "Proprietary Devices" business. Mainframe hardware is no longer the primary source of revenue for IBM as a whole, accounting for only about 4% of the company's overall sales. But the mainframe business drives an entire ecosystem of software and services, which as of 2012 accounted for about 25% of the company's overall revenue and over 40% of its total profit (Lohr, I.B.M. Mainframe Evolves to Serve the Digital World 2012).

One significant difference between the mainframe business and the world of consumer devices like e-readers and game consoles is the volume of units sold. While sales of game consoles are measured in millions – Microsoft's Xbox One and Sony's PlayStation 4 both reached sales volume of over 10 million in their first year of availability (Evangelho 2014) – sales of mainframe computers are measured only in thousands annually, as rough financial calculations show[7]. But the mainframe market is now very mature, and given that customers are essentially locked-in due to extraordinarily high switching costs, IBM can drive increasing revenues and strong profits by offering software and services to support the mainframes they have sold – thus making their mainframe business an excellent example of the "Service Funnels" strategy.

4.6 Summary

From looking at products like Amazon's Kindle, Google's Nexus, and video game consoles, we can identify three factors that make the "Service Funnels" model likely to lead to success for a given hardware device. First, low device price – these devices should be priced competitively, at or near cost, in order to drive unit sales. Second, competitive functionality in hardware and software – no matter the price, the device must offer value both functionally and technologically, or it will usually not be adopted by users. Third, desirable and compelling services – the company's services offerings must be offered in a better or more convenient way than they are offered in existing commodity devices, since this is both the differentiator for the device and the profit driver for the company.

[7] IBM's total FY2013 revenue was $99.27B. Mainframe revenue accounts for roughly 4% of this, or $3.97B. Given that the unit cost of each mainframe system is at least $1M, we can estimate that at most ~4000 mainframe systems were sold in FY2013.

The IBM mainframes offer a slightly different perspective on the “Service Funnels” model. This example shows that given a certain level of functionality in the hardware that is unmatched by any competitor, as well as a suite of desirable services, a firm can be successful producing high-cost hardware that also drives high-margin software and services sales.

Chapter 5: Patents

We bought Motorola for the sum of patents, products and people.

– Eric Schmidt, Google chairman (Gallagher 2012)

5.1 Introduction

Patent strategy has long been important to leaders of technology companies, and IBM founder Thomas J. Watson was no exception. Even at the beginning of the twentieth century, Watson's National Cash Register Company was actively filing patent infringement actions against smaller competitors, harassing them in order to obtain concessions and limit competition (Rodgers 1969, 36), and eventually drawing the ire of federal antitrust regulators. In 1933, Watson's IBM acquired Electromatic Typewriters, Inc. with the primary purpose of gaining control of important patents (Rodgers 1969, 108).

Patents, while undeniably valuable, have different underlying value propositions depending on the business strategy of their owners. In this section, we will take a look at a handful of anecdotal evidence about patents from several large technology companies and consider what impact these issues may have on the larger hardware strategies of these companies.

5.2 Google and Motorola

In 2012, Google acquired the mobile devices division of Motorola for a sum of nearly $13 billion. At the time, many suspected that this was mainly a move to acquire valuable patents in order to protect aspects of its Android mobile operating system, which was becoming more and more critical to the company's overall strategy.

The following is a timeline of events in Google's acquisition and later divestiture of the Motorola Mobility business.

August 15, 2011: Google announces acquisition of Motorola Mobility for $12.5 billion. The deal closed in May 2012 and ultimately cost closer to $13 billion when adjusting for changes in market value. The acquisition included some 17,000 patents.

June 21, 2012: Chairman Eric Schmidt noted at Google's annual shareholders' meeting: "We bought Motorola for the sum of patents, products and people." However, Google later added that it did not plan to deeply integrate Motorola, and that it would be run separately from its core business (Gallagher 2012).

December 10, 2012: Google begins its divestiture of Motorola asserts by selling off manufacturing operations in China and Brazil. The deal included "the bulk of Motorola's manufacturing capacity" (Reuters 2012).

December 20, 2012: Google sells Motorola Home, the set-top box division of Motorola Mobility, for $2.35 billion. The deal included 1,000 of Motorola's patents and 29 percent of its revenue (Dignan 2013).

January 3, 2013: FTC rules that Google must license key Motorola patents that it had previously hoped to use as a weapon in the ongoing patent litigation battles between the tech giants (Federal Trade Commission 2013).

April 25, 2013: U.S. federal court issues defeat to Motorola Mobility in its $4 billion patent lawsuit against Microsoft for use of standards-essential patents related to the 802.11 wireless standard and the H.264 video standard, instead saying it is entitled to only $1.8 million per year (Fried and Paczkowski 2013). This led some to conclude that Google had grossly overpaid for Motorola Mobility (Whittaker 2013), given that the ruling significantly reduced the apparent value of the patent portfolio acquired as part of the deal.

January 24, 2014: Google announces the sale of Motorola Mobility to Lenovo for $2.91 billion (Google 2014). Google maintained control of most of the patent portfolio, selling only around 2,000 of the patents to Lenovo as part of the deal, and granting Lenovo a license to use the remainder of the patent portfolio as part of its business.

One journalist summarized the entire sequence of events as follows: “The bottom line here is that Google bought Motorola for the patents and has largely dismantled it” (Dignan 2013).

Steve Lohr of New York Times wrote at the time of the initial Motorola acquisition by Google: “The patent portfolio, some analysts estimate, could represent more than half of the value of the deal, or more than $400,000 a patent” (Lohr 2011). Indeed, the total dollar amount recuperated from the sale of each piece of Motorola was only about half of the overall value – implying that Google did indeed end up paying in the $6 billion range for the approximately 15,000 patents that remained in its control after completing the deal with Lenovo.

5.3 Samsung and Microsoft

Patents are not only valuable as a defense mechanism against potential litigation. Microsoft has shown that they can be used to generate a significant revenue stream by striking licensing deals with hardware manufacturers whose devices could infringe on the patent holder’s rights. From 2010 through 2014, Microsoft announced patent licensing agreements with at least 21 manufacturers of Android devices, with many more such agreements likely signed but never announced (Mueller 2014). These agreements give the OEMs the freedom to sell their devices without the danger of patent infringement lawsuits from Microsoft; in return, Microsoft collects a royalty for each device sold, which is thought to be as high as $10 in some cases.

One of the most interesting and contentious of Microsoft's Android licensing deals is their agreement with Samsung. In 2011, Samsung agreed to license Microsoft's global patent portfolio for its Android patents over a period of seven years (Halliday 2011). By 2013, Microsoft was collecting royalties from Samsung of $1 billion annually. After Microsoft acquired Nokia in 2014, Samsung claimed that Microsoft had broken the terms of its own licensing agreement and refused to pay any royalties due after the acquisition was announced, which Microsoft claimed to be even greater than the amount paid in 2013 (Foley 2014).

The implication here is not that Microsoft has patents that are uniquely related to Android, or that Android manufacturers are willfully infringing patents. Rather, it is the sheer volume of Microsoft's patent portfolio related to smartphones and mobile devices, combined with the inherent interrelatedness of the various hardware and software technologies, which make it essentially impossible to build a device that does not infringe on an unknown number of patents issued to other firms. A deep-pocketed company like Microsoft can often pressure a smaller hardware manufacturer into a licensing agreement with the threat of future litigation and thus produce a revenue stream – even if, as in the case of Microsoft and Android, that revenue stream comes from the sale of products that belong to a competitor's ecosystem.

5.4 Patent strategy

Lohr notes that the estimated $400,000-per-patent paid by Google for Motorola's portfolio is actually a low price compared to the $750,000 per patent paid just two months earlier for the patent portfolio of Nortel Networks. But it was far above the average price of $200,000 per patent in the preceding few years. Only a few months later, in April 2012, Microsoft purchased 800 patents from AOL for over $1 billion – about $1.3 million per patent, over three times the estimated valuation of the Motorola patents – in an effort to strengthen its position among makers of software and hardware for smartphones. Facebook also acquired 750 patents from IBM in the wake of a patent suit filed by Yahoo (Lohr, Microsoft's AOL Deal Intensifies Patent Wars 2012).

John Amster, chief executive of a start-up founded in 2008 called X, said that "the Nortel and Motorola deals were mainly special cases. The high prices reflected not only an assessment of the value of the patents, but the much larger objective of trying to gain the upper hand in the smartphone business" (Lohr 2011). But the continued activity in this area suggests that these deals may not be exceptional cases after all.

We can see that patents are valued not just for the associated intellectual property rights, but actually as a strategic business maneuver – an attempt to defend against or even prevent potential patent infringement lawsuits from other device manufacturers or software vendors. Scott Stern, a professor at MIT Sloan School of Management, notes: "The trouble is that in this industry so often a patent is not a clearly defined property right, but a lottery ticket of uncertain value. That uncertainty can carry a lot of risk and cost" (Lohr 2011).

This is particularly true for the creators of mobile operating systems like Apple's iOS, Google's Android, or Microsoft's Windows Phone, as evidenced by our discussion in Section 5.3 of

Microsoft’s licensing agreements with an undisclosed number of Android device manufacturers. Because device manufacturers using these operating systems are often targeted for infringement lawsuits by patent holders, the software companies are often pressured to help defend the device manufacturers in order to keep their products in the market (Lohr 2011).

5.5 Summary

Patent strategy at the convergence of software and hardware is a complex situation with many different side effects to consider. In general, of course, it is always advantageous to own as many patents as possible, but we have seen over the years that it is very difficult to accurately evaluate their worth.

The division between hardware and software patents is very murky, especially in the arena of mobile devices, where hardware and software are tightly integrated into a single experience. Some patents, such as those related to multi-touch interfaces, might be closely related to both the hardware and software that implements the feature in the device, and perhaps inseparable from either. And in some scenarios, such as the Android case discussed previously, hardware manufacturers must bear the burden of licensing agreements for features found in components of the open-source operating system they use, even if they did not develop it.

Typically, only the large companies like Apple, Google, Samsung, and Microsoft can afford serious patent litigation. Because of this, patents may be valued primarily as a strategic tool in battles between large corporations, or in an effort to defend smaller vendors who are supporting a company’s larger platform or ecosystem. Microsoft’s collection of royalties from Android device manufacturers also show that effective use of litigation and licensing agreements can result in a sizable revenue stream, given a large enough portfolio of relevant patents.

Patents are valuable to any company with a platform and an ecosystem to protect, regardless of whether they actually produce the software or hardware on which their patents are based. A firm's patent portfolio, as well as those of its competitors, are simply additional factors that must be taken into account in a complex decision-making process of whether, and how, to build hardware.

Chapter 6: Hardware Strategies at Microsoft

The co-evolution of hardware and software is going to define what's going to happen. We have a particular definition of mobile which is perhaps skewed to the mobile phone. But you take the industrial Internet or the Internet of everything, everything's going to be connected to the cloud, and data. So that's the world that we are building for.

– Satya Nadella, Microsoft CEO (Hachman 2014)

6.1 Introduction

Beginning in 2014, the "devices and services" strategy espoused by former Microsoft CEO Steve Ballmer took a back seat to the "mobile-first, cloud-first world" described by Satya Nadella, who took over the CEO post in February 2014. As a result, building first-party devices took on much less strategic importance to the company, with the emphasis shifting to cloud services and mobile applications to enable access to Microsoft's productivity software and services from any platform or device. However, Microsoft remained very much committed to building a certain set of hardware to complement its own ecosystem, including the Surface line of tablets and the mobile phone business it acquired from Nokia.

Even Microsoft recognizes that its general-purpose software platforms are struggling in the face of the trend toward more integrated devices, specifically phones and tablets. The following quote is a message from Microsoft to its investors in its Form 10-Q from January 2014:

> *Users are increasingly turning to these devices [smartphones and tablet computers] to perform functions that in the past would have been performed by personal computers. Even if many users view these devices as complementary to a personal computer, the prevalence of these devices may make it more difficult to attract applications developers to our platforms.*

In this chapter, we will discuss Microsoft's historic approaches to first-party hardware, including products like Xbox, Zune, Kin, and Surface. We will discuss the successes and failures associated with these product lines, including major problems like the large write-down from unsold inventory of the first-generation Surface. We also consider how these approaches compare to Microsoft's acquisition and subsequent integration of Nokia's mobile device unit. Finally, we will summarize the approaches Microsoft is currently taking in the market and make recommendations for future strategies based on the framework we have set forth in previous chapters.

6.2 Xbox

Microsoft announced its new Xbox gaming console on March 10, 2000 and officially launched it on January 6, 2001. Don Coyner, director of marketing for Microsoft Games, described the device in this way: "The PC and Xbox are complementary devices. Each has very distinct audiences. PC games are more cerebral, while console games are more visceral" (Microsoft 2000).

The Xbox is one of the primary success stories of Microsoft's hardware ventures, but it was not an immediate success from a business perspective. Microsoft has invested in the Xbox for well over a decade, but only since 2008 has it been a profitable business. It is important to note that Microsoft has never viewed the Xbox as an end to itself, but rather as a way to expand its Windows platform, and more recently, its cloud services like Skype and Bing.

Microsoft has long viewed the living room as an important place to have a presence, even going back to their purchase of WebTV in 1997. These living room devices have traditionally been specialized, proprietary devices focused on entertainment and a limited set of productivity scenarios. Norman Young of Morningstar writes of Microsoft's strategy with Xbox:

> *The Xbox console [is] another device to help tie consumers into Microsoft's cloud services. As the link to consumers and their living rooms, the Xbox can help the firm reinforce network effects and increase switching costs through services such as Xbox Live, Xbox Music, Skype, and Bing. (Young 2014)*

So we can see that the Xbox has encompassed, at least to some degree, all of the strategies that we described in preceding chapters. Microsoft initially conceived of Xbox as a complementary device to the PC, which implies that they were at least partially viewing it from the "Hardware First" lens described in Chapter 2 – that is, they wanted their Xbox division to be focused on hardware without the encumbrances of the divisions of the company that were focused on software and hardware for the PC. It is also clearly an example of the "Proprietary Devices" strategy. We have already discussed Xbox through this lens in Section 3.6, and the strategy is also discussed above in relation to Microsoft's WebTV acquisition. Finally, Xbox is also a prime example of the "Service Funnels" strategy, as we discussed in detail in Section 4.4.

6.3 Zune and PlaysForSure

In 2004, as iTunes and iPod began to capture a large part of the market for portable music players, Microsoft teamed up with OEMs to create a system called PlaysForSure. At the time, Apple's music offerings were protected by a proprietary digital rights management (DRM) system, which prevented users from copying iTunes music to any device other than iPod players, effectively locking them in. PlaysForSure was an open DRM standard that was intended to allow users to obtain music from any supported online music reseller and copy the files to any device that supported the PlaysForSure standard.

By pursuing this strategy, Microsoft was attempting to use an open standard, general-purpose software, and partly-specialized hardware devices from OEMs to provide a solution in the portable music player market. But the problem space and the associated technology, including both portable device hardware and DRM standards, were far too immature to allow this open system to provide an end-to-end experience that was competitive with the iPod/iTunes integration offered by Apple. Compounding the problem for Microsoft, iPod/iTunes had its own proprietary DRM and did not support PlaysForSure music. To top it all off, the iTunes music store had a stranglehold on the digital music market in 2007, with a market share around 70 percent (Ahrens and Musgrove 2007) – and music purchased from iTunes was not usable on a PlaysForSure device primarily because of the different DRM scheme. As a result, the PlaysForSure standard failed to gain a significant foothold in the market and was officially discontinued by Microsoft in 2007[8].

This scenario had the characteristics that would have caused the "Proprietary Devices" strategy to be a more successful approach, but Microsoft recognized this too late. In 2006, Microsoft released their Zune portable music player and associated music download service called Zune Marketplace, directly competing with iPod and iTunes. Again, this was a clear instance of the "Proprietary Devices" model – Microsoft controlled the entire experience, from the music search and download process all the way to the music listening experience. But because Apple's proprietary DRM prevented iTunes music from playing on any non-Apple devices – and similarly, because Zune's DRM system prevented its music from playing on the iPod – Zune utterly failed to overcome the network effects associated with Apple's existing market dominance. Somewhat unbelievably, Zune's DRM scheme was even incompatible with Microsoft's own PlaysForSure standard, which further reduced its attractiveness to consumers with existing music libraries.

[8] In fact, DRM for music generally began to disappear around this time as the voice of the market became clearer. Amazon launched its DRM-free MP3 store in 2007, and more and more music retailers began to abandon DRM. Apple's iTunes also began offering some DRM-free songs in 2007, and dropped DRM entirely in 2009.

So despite a few novel features in the Zune hardware, there was no real differentiation from the Apple solution, and because the iPod had already become dominant, the Zune failed to gain any appreciable market share. After only about five years on the market, both the Zune hardware line and the Zune services were retired. If Microsoft had truly wanted to create differentiation here, it may have needed to get ahead of the curve and produce a DRM-free solution before Apple could react – but given the music industry's attitude toward piracy at the time, this could very well have been an impossible task.

6.4 Kin

In 2008, Microsoft acquired Danger, a phone manufacturer which had previously produced the popular Sidekick phone for T-Mobile, and began an in-house project to develop a device that would be based on the next iteration of the Windows Mobile operating system.

This project, which ultimately became Kin, started in 2008 with the goal of creating "a platform agnostic, cloud-centric feature phone" (Topolsky 2010), and along with Kin came a suite of services primarily targeted at social networking scenarios. This vision sounds surprisingly similar to Microsoft's "mobile-first, cloud-first" mantra of 2014. But after an internal battle between Microsoft executives, Kin was pulled into the Windows division and suffered massive delays as the platform was rewritten (Ziegler 2010). In addition, once the phone was finally released, Verizon priced the phone and its required data plan much higher than Microsoft had anticipated. A phone that was essentially two years late and overpriced had no chance to succeed in the market, and Microsoft discontinued the phone in June 2010, less than two months after it had launched.

Kin can be viewed as a failed attempt at a "Service Funnels" model. In addition to the more obvious reasons for its failure described above, it lacked the key elements that allow such a model to succeed: namely, low device price, competitive functionality in hardware and software, and services that are desired by the user and that can be offered in a better or more convenient way than in existing commodity devices. Certainly, had the phone been released on schedule and with a more aggressive pricing scheme, it would have been much more competitive with other offerings on the market at the time and been more successful in driving users to the Kin cloud services.

6.5 Surface

The Surface line of tablets was launched in 2012 as a direct response to the overwhelming success of Apple’s iPad. Unquestionably, this was an example of the “Hardware First” strategy – Microsoft, like Apple, was attempting to build a general-purpose computing device where it controlled the entire hardware and software experience.

In statements from its Form 10-Q in January 2014, Microsoft describes its own motivation for creating the Surface line of tablets:

> *A competing vertically-integrated model, in which a single firm controls the software and hardware elements of a product and related services, has been successful with some consumer products such as personal computers, tablets, mobile phones, gaming consoles, and digital music players. Competitors pursuing this model also earn revenue from services that are integrated with the hardware and software platform.*

In the same Form 10-Q, Microsoft also points out the obvious downsides to building its own hardware devices: “Efforts to compete with the vertically integrated model will increase our cost of revenue and reduce our operating margins. … In addition, Surface competes with products made by our OEM partners, which may affect their commitment to our platform.”

Despite a massive marketing campaign, the Surface was met with tepid reaction by consumers. Microsoft took a $900 million charge to write down the value of unsold Surface inventory in July 2013. This caused many observers, including some Microsoft board members, to question whether it was wise for Microsoft to expand into hardware even more aggressively with its purchase of Nokia's handset business (Cave 2014). If nothing else, it was quite apparent that Microsoft had not yet developed the core capabilities of a hardware company like Apple.

6.6 Nokia

After several years of close partnership, Microsoft completed its acquisition of Nokia's devices business in early 2014, as discussed in more detail in Section 2.3. Unlike the arm's-length strategy Google used with its Motorola acquisition and later divestiture, Microsoft has shown that it intends to fully integrate Nokia into its corporation in order to take full advantage of its position to gain efficiencies and synergies that were not possible while the two companies were separate. In doing so, Microsoft has again shown a desire to imitate Apple's model of owning the end-to-end hardware and software experience of its devices – perhaps an example of the "Hardware First" approach, at least within the confines of the Microsoft Devices division.

At the time of this writing, not enough data exists to make a judgment call on the success or failure of the Nokia acquisition. Based on past precedent, we can expect the Nokia strategy to mostly align with the Surface strategy, and likely with similar results – especially since Microsoft has decided to tie the fortunes of its device families together by unifying its operating system and developer ecosystems with its Windows 10 release due in late 2015.

6.7 Summary

Microsoft has a long history of building hardware devices with decidedly mixed results. Indeed, this section has only covered Microsoft's more well-known hardware ventures, not even touching on less-publicized projects like the acquisition of Perceptive Pixel as an entry into the niche market of extremely large-screen touch-enabled displays. The one clear success story is Xbox, a case where Microsoft has used the "Service Funnels" strategy to drive significant revenue through sales of games and subscriptions. But in nearly every other case, Microsoft has been late to market and has followed a combination of the "Hardware First" and "Proprietary Devices" strategies where they have not proven themselves particularly adept. Indeed, our set of examples throughout this paper have shown that software companies tend to be more successful when their devices promote their own software and services, rather than trying to be an end in themselves.

Microsoft may certainly be successful with its current hardware device lineup, but as a software company at the core, it must carefully consider how to use its devices to promote its ecosystem of software and services in a way that is not achievable on its competitors' devices. The spectacular failures of devices like Zune and the first-generation Surface are examples that warn against a software company like Microsoft building devices for their own sake out of perceived competitive pressure, rather than crafting a device that is advantageous to the company's core business.

Chapter 7: Conclusions

We have seen that the trend toward smaller, more highly-integrated devices, along with the fast pace of innovation in the technology industry, has caused software companies to build hardware products to showcase their own software. We have also seen that building hardware products does not always lead to success, and in fact this strategy creates a financial risk for the company by significantly reducing profit margins as compared to the traditional profit margins to which large software companies are accustomed.

Because of these risks, a firm must carefully consider whether building hardware will be advantageous to its business objectives and ultimate financial success. We have observed three specific strategies that firms have used successfully in this area. We initially looked at the “Hardware First” strategy, wherein a company builds devices with the primary goal of selling those devices – bundled, of course, with the company’s software. Apple is the classic example of this strategy, and firms that have attempted to copy their model, such as Google with its Motorola acquisition or Microsoft with its Nokia acquisition, have not yet been met with great success. Second, we considered the “Proprietary Devices” strategy, in which a company builds a device that is targeted at a particular market or function and locks in the customer to the firm’s ecosystem. This strategy has been observed to succeed in markets where the technology is not yet mature, such as with Apple’s iPod or iPhone, as well as in cases where the device has a very particular purpose that cannot be achieved as effectively with a general-purpose device, such as with IBM’s mainframe computers or Amazon’s Kindle series of devices. Third, we considered the “Service Funnels” strategy, wherein a firm builds hardware devices that are primarily intended to drive usage and revenue of its core software and services products. Successful examples of this strategy include the Kindle, as well as gaming consoles like Microsoft’s Xbox or Sony’s PlayStation.

Finally, we summarized the various hardware strategies that have been used by Microsoft over the years, including products such as Xbox, Zune, Kin, and Surface. We have shown that each of the preceding three strategies have been used by Microsoft at various points in its history, and we have used the analysis of these strategies to help explain why some products have succeeded and why other products have failed dramatically in the marketplace.

In 2015 and beyond, the biggest unanswered question for Microsoft's hardware strategy is what it will do with the recently-acquired Nokia devices business. Microsoft has already begun the process of fully integrating Nokia and bring it under its corporate umbrella, including combining it with its existing hardware division and eliminating the associated redundancies by laying off nearly half of the acquired workforce. Smartphones and tablets will be branded with Microsoft's name, not Nokia's, which further indicates the extent to which Microsoft aspires to fully integrate Nokia's hardware business into its own. But as Google did with Motorola, there remains a high likelihood that Microsoft will integrate the most desirable pieces of Nokia's business – namely, the Windows Phone ecosystem that it has produced – and simply sell off or even abandon the rest.

For Microsoft, the key takeaway is that it must continue to build only those devices that are advantageous to the core revenue creators of the company – namely, its software and services products such as Windows and Office. Microsoft's core capability is in software, and it must learn from both its own history and the history of other companies in the industry when making decisions about how to market its hardware products. Developing a mutually-beneficial relationship between its hardware and software products will be key to the long-term success of Microsoft in today's technology landscape.

References

Afuah, Allan. 2001. "Dynamic Boundaries of the Firm: Are Firms Better Off Being Vertically Integrated in the Face of a Technological Change?" *Academy of Management Journal* 44 (6): 1211-1228.

Ahrens, Frank, and Mike Musgrove. 2007. "Music-Selling Rivals Take Aim at iTunes." *The Washington Post.* August 22. http://www.washingtonpost.com/wp-dyn/content/article/2007/08/21/AR2007082100996.html.

Apple. 2003-2013. "Form 10-K." *SEC Filings.* http://investor.apple.com/sec.cfm?DocType=Annual.

Brodkin, Jon. 2009. "The downfall of Sun Microsystems." *Network World.* April 24. http://www.networkworld.com/article/2268096/servers/the-downfall-of-sun-microsystems.html.

Cave, Andrew. 2014. "Former Microsoft CEO Steve Ballmer's Biggest Regret." *Forbes.* March 4. http://www.forbes.com/sites/andrewcave/2014/03/04/former-microsoft-ceo-steve-ballmer-my-biggest-regret.

Dignan, Larry. 2013. "Google's Motorola purchase: Was it worth it?" *ZDNet.* January 4. http://www.zdnet.com/googles-motorola-purchase-was-it-worth-it-7000009356/.

Evangelho, Jason. 2014. "Xbox One Outselling PS4, Microsoft Celebrates 10 Million Units Shipped." *Forbes.* November 12. http://www.forbes.com/sites/jasonevangelho/2014/11/12/xbox-one-outsells-ps4-in-november-microsoft-touts-10-million-shipped/.

Federal Trade Commission. 2013. "Google Agrees to Change Its Business Practices to Resolve FTC Competition Concerns In the Markets for Devices Like Smart Phones, Games and Tablets, and in Online Search." *Federal Trade Commission.* January 3. http://www.ftc.gov/news-events/press-releases/2013/01/google-agrees-change-its-business-practices-resolve-ftc.

Foley, Mary Jo. 2014. "Samsung paid Microsoft $1 billion in Android patent-licensing royalties in 2013." *ZDNet.* Octiber 4. http://www.zdnet.com/article/samsung-paid-microsoft-1-billion-in-android-patent-licensing-royalties-in-2013/.

Fried, Ina, and John Paczkowski. 2013. "Court Denies Motorola the Billions It Wanted From Microsoft for Standards-Essential Patents." *AllThingsD.* April 25. http://allthingsd.com/20130425/court-denies-motorola-the-billions-it-wanted-from-microsoft-for-standard-essential-patents.

Gallagher, Dan. 2012. "Google says Motorola is more than patents." *MarketWatch.* June 21. http://www.marketwatch.com/story/google-says-motorola-is-more-than-patents-2012-06-21.

Google. 2003-2013. "Form 10-K." *SEC Filings.* http://google.client.shareholder.com/investorkit.cfm.

—. 2014. "Google Search Appliance 7.2." *Google Search for Work.* https://www.google.com/work/search/products/gsa.html.

—. 2014. "Lenovo to Acquire Motorola Mobility from Google." *Google Investor Relations.* January 29. http://investor.google.com/releases/2014/0129.html.

Hachman, Mark. 2014. "Mobile first, cloud first: Nadella's new mantra." *PC World.* February 4. http://www.pcworld.com/article/2094281/mobile-first-cloud-first-microsofts-new-mantra.html.

Hall, Mark, and John Barry. 1990. *Sunburst: The Ascent of Sun Microsystems.* Chicago: Contemporary Books.

Halliday, Josh. 2011. "Samsung and Microsoft settle Android licensing dispute." *The Guardian.* September 28. http://www.theguardian.com/technology/2011/sep/28/samsung-microsoft-android-licensing-dispute.

Helft, Miguel. 2012. "Fortune Exclusive: Larry Page on Google." *Fortune.* December 11. http://www.fortune.com/2012/12/11/fortune-exclusive-larry-page-on-google/.

Henderson, Rebecca M, and Kim B Clark. 1990. "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms." *Administrative Science Quarterly* 35 (1): 9-30.

Hertzfeld, Andy. 1982. "Creative Think." *Folklore.* July 20. http://www.folklore.org/StoryView.py?project=Macintosh&story=Creative_Think.txt.

IBM. 2003-2013. "Form 10-K." *Financial reporting.* http://www.ibm.com/investor/financials/financial-reporting.html.

—. 2002. "IBM to Acquire PwC Consulting." *IBM.* July 30. https://www-03.ibm.com/press/us/en/pressrelease/585.wss.

—. 2014. "Mainframe software for IBM System z." *IBM Software.* http://www-01.ibm.com/software/os/systemz/en_US/.

Krishna, R Jai, and Rolfe Winkler. 2014. "Google Unveils $105 Smartphone for Emerging Markets." *The Wall Street Journal.* September 15. http://online.wsj.com/articles/google-unveils-105-smartphone-for-emerging-markets-1410785613.

Lohr, Steve. 2011. "A Bull Market in Tech Patents." *The New York Times.* August 16. http://www.nytimes.com/2011/08/17/technology/a-bull-market-in-tech-patents.html.

—. 2012. "I.B.M. Mainframe Evolves to Serve the Digital World." *The New York Times.* August 28. http://www.nytimes.com/2012/08/28/technology/ibm-mainframe-evolves-to-serve-the-digital-world.html.

—. 2012. "Microsoft’s AOL Deal Intensifies Patent Wars." *The New York Times.* April 9. http://www.nytimes.com/2012/04/10/technology/microsoft-to-buy-aol-patents-for-more-than-1-billion.html.

Microsoft. 2003-2014. "Form 10-K." *Microsoft SEC Filings.* http://www.microsoft.com/investor/SEC/default.aspx.

—. 2000. "Xbox Brings "Future-Generation" Games to Life." *Microsoft News Center.* March 10. http://news.microsoft.com/2000/03/10/xbox-brings-future-generation-games-to-life/.

Miller, James R, and Donald A Norman. 1986. "Arguments against general-purpose interfaces: The case for specialization." *Miramontes Interactive.* August. http://www.miramontes.com/writing/specialpurpose/.

Motorola. 2003-2013. "Form 10-K." *SEC Filings.* http://investors.motorolasolutions.com/sec.cfm?DocType=Annual.

Mueller, Florian. 2014. "Microsoft announces its 21st Android patent licensee, 26th known Android patent deal in total." *FOSS Patents.* February 14. http://www.fosspatents.com/2014/02/microsoft-announces-its-21st-android.html.

Nokia. 2003-2013. "Form 10-K." *SEC Filings.* http://company.nokia.com/en/investors/financial-reports/sec-filings.

Oracle. 2003-2014. "Form 10-K." *SEC Filings.* http://investor.oracle.com/financial-reporting/sec-filings/default.aspx.

—. 2014. "Oracle Database Appliance X4-2." *Oracle Technology Network.* http://www.oracle.com/technetwork/database/database-appliance/overview/index.html.

Reuters. 2012. "Flextronics takes over Motorola assembly plants." *Reuters.* December 10. http://www.reuters.com/article/2012/12/11/flextronics-motorola-idUSL1E8NB08Y20121211 .

Rodgers, William. 1969. *THINK: A Biography of the Watsons and IBM.* New York: Stein and Day.

Stone, Brad. 2011. "Amazon's Bezos Fuels Tablet Wars with $199 Kindle." *Bloomberg News.* September 28. http://www.bloomberg.com/news/2011-09-28/bezos-portrays-pocket-sized-fire-as-service-not-tablet-in-ipad-challenge.html.

Sun Microsystems. 2003-2009. "Form 10-K." *U.S. Security and Exchange Commission.* https://www.sec.gov/cgi-bin/srch-edgar?text=Sun+Microsystems+10-K&first=2003&last=2014.

Takahashi, Shigeru. 2005. "The Rise and Fall of Plug-Compatible Mainframes." *IEEE Annals of the History of Computing* 27 (1): 4-16.

Topolsky, Joshua. 2010. "What killed the Kin?" *Engadget.* June 30. http://www.engadget.com/2010/06/30/what-killed-the-kin/.

Trefis. 2014. "Estimating the Value of Kindle Hardware Sales for Amazon." *Forbes.* March 11. http://www.forbes.com/sites/greatspeculations/2014/03/11/estimating-the-value-of-kindle-hardware-sales-for-amazon/.

Warren, Tom. 2014. "Microsoft kills off its Nokia Android phones." *The Verge.* July 17. http://www.theverge.com/2014/7/17/5911909/microsoft-kills-off-its-nokia-android-phones.

Whittaker, Zack. 2013. "Microsoft, Motorola ruling: Google patents not worth billions." *ZDNet.* April 26. http://www.zdnet.com/in-microsoft-patent-spat-ruling-hints-that-google-grossly-overpaid-for-motorola-7000014582/.

Wilhelm, Alex. 2013. "Inside Microsoft's earnings: Windows 8, and the Xbox money machine." *The Next Web.* April 19. http://www.thenextweb.com/microsoft/2013/04/19/inside-microsofts-earnings-windows-8-and-the-xbox-money-machine/.

Young, Norman. 2014. "Microsoft's (Mis)Adventures in Hardware." *Morningstar.* October 15. http://news.morningstar.com/articlenet/article.aspx?id=668548.

Ziegler, Chris. 2010. "Life and death of Microsoft Kin: the inside story." *Engadget.* July 2. http://www.engadget.com/2010/07/02/life-and-death-of-microsoft-kin-the-inside-story/.