nakedinnovation

uncovering a shared approach for creating value

Zachary Jean Paradis
David McGaw
IIT Institute of Design

YUYANGA/ISTOCKPHOTO

5 Technology
Platform & System Design

Platforms rule!

larry keeley, doblin inc.

Do us a favor. Take a moment to think about how many times and ways in the last 24 hours that you used the Internet. Really think about it. Our own casual count quickly shot over 100. Whether checking bank statements or paying bills, reading the New York Times or catching up on a favorite blog, buying a book on Amazon.com or finding a new restaurant, the Internet has become an immensely useful and valuable ***Platform***. It is shocking to think about the fact that just over 20 years ago, it didn't exist. What would we do today without it?

While there are many who played a part, there is no one person who "invented" it—it required many people and organization's cooperation. Similarly, there is no one person who profits from it—we all do. While you may not expect to invent an offering with the scale of Internet, you should build solutions based on standards and platforms. How can we be actively involved in important decisions related to feasibility and technology for the projects on which we work? Also, what can we learn from technology that applies to apparently non-technical products like food or books? Before we answer these two questions, let's ask something else.

Have you ever been in a meeting where an engineer (or other technical person) made you feel like you couldn't take part in the conversation because you weren't knowledgeable enough? Or, if you are a technical person, have you ever used complex terms specifically so you didn't have to explain the details of some issue? We bet you have been on one or the other side at some point. We have seen time and time again that decisions—specifically, technical ones—get made by a small number of people from one discipline who may or may not fully understand the complete ramifications on the business and the value it intends to deliver to its customers. We're not knocking engineers but we are making a call to arms to non-technical disciplines. You have to take part in these big decisions because they will indeed affect your customers! Similarly, we have hope that technologists reading this book will realize their best contribution can be to help others understand the ramifications of one decision or another. You are our guides.

Solution Architecture

The good news is that regardless of whether you are a "techie" or not, you can use some basic knowledge about how technology works agnostically to help make decisions. First, you need to understand a bit about ***Solution Architecture*** and the nature of platforms. This is the good stuff to understand even if you aren't working with a specific protocol or technical specialty. It's also worth knowing regardless of whether you're building a website or a consumer electronic device. Second, regardless of whether you are naturally inclined to or not, if you want to be an innovator you should become passionate about the technological trends that shape our world and affect your current and next project. Although we would never suggest innovation is solely driven by

technology, it is one essential part of creating a balanced breakthrough. We'll outline some high level steps you can take to understand these trends, but first let's get to some much needed definitions.

The term "Solution architecture" refers broadly to the way in which functional pieces are arranged into some end solution. The processor, hard disk, track pad, keyboard, and other functional pieces come together to make the laptops on which we write this book. There are two main types of architecture. On one hand, **integral** solutions are built to very specifically solve one problem and not necessarily relate well to other problems—their parts fit together uniquely. On the other, **modular** solutions, even when built for a specific problem, tend to be flexible in terms of their applicability to other problems—their parts could fit with others to create solutions for different problems. Each of these ways of conceiving some solution has costs and benefits associated with them. Let's illustrate these through a few examples.

Integral Solutions

Racing style motorcycles and fine Swiss watches are built for performance. For this reason, motorcycle and watch designers often choose an integral product architecture for top-end products. Integral architecture offers fast, efficient, and precise interactions between parts with little waste in materials—thus, it is appropriate for higher performance solutions. But parts for racing bikes or fine watches really aren't going to be good for much else. On the opposite end of the complexity continuum is a simple pair of scissors. Again, designed with performance for a single use in mind, cutting plain paper, the parts fit together in a way that they aren't very useful in application to other problems. Developing a new unique product in this way can get you to market really fast. This is the essence of choosing to making something in an integrated way: the pieces fit together exceptionally well—they are designed from the ground up to work together—but are of only limited utility and re-use. It would be difficult to modify them without a lot of work. More importantly, there is little value built between generations. You pretty much have to start from scratch each time you create a new product. This is less of an issue with scissors, but when you're making a motorcycle, or anything else complex, it can be costly.

Modular Solutions

In contrast, let's consider desktop personal computers and Swatch watches. Most of us have gone through the experience of purchasing a new mouse, hooking up a digital camera, or maybe even installing more RAM or an internal hard disk. While these hardware upgrades and modifications can be painful for non-technical people, they aren't impossible—that's mainly the result of the modular nature of their architecture. This modularity is well illustrated in the "good, better, best" versions of many desktop computers today. The basic system is maintained but for faster components or those with more capacity. It allows manufacturers to offer variety and choice to consumers with the benefits of economics of scale and re-usability.

A similar architecture was used by Swatch to revolutionize the definition of Swiss watch design. By establishing a standardized and modular platform, Swatch could offer an enormous variety of designs while still producing them at relatively low cost. Changing a piece in an integral solution can effect many different other functional pieces; changing a piece in a modular one should have little to no effect on the rest of the system. In software development, this is exactly the benefit "Object Oriented" programming is supposed to achieve: one piece can be flexibly upgraded or changed without fundamentally having to upgrade the entire solution. Unfortunately, this flexibility also comes at some cost. It generally takes more time to design

and build modular solutions than it does integrated ones. Additionally, a problem with one part or "module" used in many systems creates problems in all of the solutions making use of it.

Regardless of their issues, we believe in building modular solutions when addressing complex problems. In fact, as we noted at the beginning of this chapter, we would go so far as to suggest that we (and you) should try to build platforms whenever possible. The nice thing about platform thinking is that it is equally applicable to non-technical solutions as it is technical ones. Take the *For Dummies* book series. As pedestrian in nature as the Internet is grand, the series has been remarkably successful since it first debuted in 1991 with *DOS For Dummies*. With more than 125 million books sold, they have shown the world how to do just about everything: use computers, cook, garden, manage finances, run a business, buy a home, plan a trip, exercise, and eat right. The creators do this not through technical prowess but through a consistent organizing principle and modular parts. *For Dummies* is every bit of a platform that the Internet is. It allows books from fantastically diverse subject matter to be written, presented, and consumed in a common and an easily accessible fashion. *For Dummies* also demonstrates how integral solutions many times become modular over time. The first book wasn't viewed as a series but as single solution. Only over time did the formula of how to write, design, and produce the books become modular and standardized.

So what's really the essence of modular architectures and platform thinking? Standards. Whether closed to an organization, partially shared, or completely open, standards are the key to realizing economies of scale, allowing separate technologies to work together, and bringing together contributions from disparate teams. Realize that the further you stray from open standards, the more costs will be associated with development, and the harder it will be for others to work with you. That is not to say closed shouldn't be used. For example, open platforms and standards were and are essential to the development of personal computers overall, but smart firms within the industry still maintain control over the distinctive value they deliver. Intel, Apple, Microsoft, HP, among others have contributed in and shared the common USB peripheral platform. All have benefited without giving up control of the piece(s) of the puzzle that allows them to differentiate.

Understanding Trends

To be ahead of the curve, we need to spot the emerging and converging trends that will lead to future standards. How can non-techies and geeks alike understand the impact of emerging trends in the world? We find one particularly useful activity is the building of ***Era Maps***.[1]

Era Maps are one of our favorite tools in this kit and are equally applicable when looking at trends in culture, the competitive space, and importantly for this chapter, technology. They are great to complete at the beginning of a project to set the larger context in which your development will be found. Fewer pieces of knowledge are as important as where you've been, where you are, and where you're going. That is exactly what is great about them: they offer a wide-angle lens through which to view past, current, near-future, and future use of the technology you will use to build your solution. Let's consider an example or two.

At a *really* macro level, the personal computer was a big platform of the 1980s—think IBM PCs, Apple IIs, Commodore 64s, and the like. "Official" networks were the platform for the 1990s—think AppleTalk, corporate networks, and shared printers—all built on the previous personal computer platform). Our current

[1] IIT Institute of Design professor Vijay Kumar taught us the value of Era Maps, which we've seen our colleagues use in myriad projects.

platform is the Internet, which was built on shared networks and personal computers. So what is our technological future? The trends tell us mobile and wireless hardware and applications will dominate the next decade. We've been told for years that the "network is pervasive" but now it really is. This might sound cliché but it isn't. We would bet that your firm will use mobile applications (if it isn't building them itself). The more you account for that in your planning, the better your firm will succeed.

SAMPLE ERA MAP: MOBILE COMMUNICATION DEVICES

	PAST	PRESENT	NEAR FUTURE	FUTURE
Hardware	Small screens Monochromatic Limited processing power Limited entry interface	Large screens Color Reasonable processing power Flexible and intelligent entry interface	Smaller devices Rich media capabilities Multiple band connectivity GPS enabled Screen-based keboards	Haptics Augmented reality headsets Virtual keyboards Star Wars holograms Mini Projector
Applications	SMS based Limited mobile browser e.g. WAP Command line data entry	Web enabled WAP 2.0 applications Service provider embedded apps Flexible and expanded data entry Downloadable JAVA enabled	Community based applications Media aggregating Rich multimedia	POP payment Phone as identifier Universal remote control Learns user preferences Natural language recognition
Operating System	Particular to each manufacturer	Windows CE RIM blackberry Palm OS Manufacturer Specific	Windows CE RIM blackberry Palm OS Manufacturer Specific Linux	Windows CE RIM blackberry Manufacturer Specific Linux
Network	Low bandwidth Limited access Regular dropouts	2.5G/EVDO	3G Mesh networks WiFi GIS systems	WiMax Ubiquitous connectivity IP aware

Consider your work through the lens of an Era Map: what technology have people used in the past, what do they use now, and what will they use in three years or a decade? Use the generic diagram on the previous page to build on. We've filled it out with data from a project related to mobile phone restaurant reviews for demonstration purposes. We executed it quickly in November of 2006 and it is interesting to us that with very little research, we predicted several of the iPhone's platform features. Era Maps really work! You might not readily be able to complete all the boxes when you start a project. This is good—it will encourage you to do some research and engage with those who are in charge of making larger technological decisions.

One more issue with platforms needs to be addressed. This is a big one. *Platforms only become relevant when they move past a certain tipping point of adoption—either external or internal to your firm.* Simply put, enough people have to make use of it for it to achieve platform benefits. Why is this so important? Platforms, especially technological ones, can be quite expensive to build. It is much easier and faster to produce a "one-off" than to thoughtfully design a platform that will have legs for years to come.

The decision to build a platform, and, more importantly, whether it be closed, open, or partially-open, will end up, for better or worse, affecting your firm's products for a long time in the future. Sony's decision to keep Betamax (a superior solution in nearly every way) closed and proprietary doomed it to financial failure when compared to the open approach taken by VHS, offered by JVC. Microsoft beat Apple by opening Windows to run on any hardware, although the MacOS and the Macintosh were superior products, at least when Windows first debuted.

But the answer isn't always to be open. Apple's resurgence of late has largely rested on its ability to produce truly integrated experiences—think of how seamlessly the iPod/iPhone works with iTunes and the media/apps store. This thoughtful integration could only be achieved because it is at least partially closed and proprietary. A rule of thumb is to open whatever will help grow the marketplace you are working in without ultimately giving away the distinctive value that your firm delivers.

Being part of the technology decision-making process isn't easy but it is better than sticking your head in the sand and ignoring it. Understanding the implications of the different types of solution architecture and platforms within your competitive space will give you a big advantage in understanding how to balance your innovation. This is the type of vision that points to what to your team and firm should invent.

Before You Go On...

Let's review the keys to understanding technology in Naked Innovation.

› Know the different types of architecture solutions can take. Understand the difference between *integral* and *modular*.

› When ever possible, try to build your solution around *platforms* based on common *standards*. Share valuable information and development burdens when possible to grow markets but don't give away what differentiates your firm!

› If you are non-technical, have the courage, fortitude, and the humility to take part in important technical conversations. If you are geek, understand that you are more powerful and valuable to your firm when you actively engage non-technical colleagues in meaningful discussions.

RESOURCES FOR TECHNOLOGY & PLATFORMS

Gladwell, Malcolm. *The Tipping Point: How Little Things Can Make a Big Difference*. New York: Little, Brown, 2000.

Meyer, Marc H., and Lehnerd, Alvin P. *The Power of Product Platforms: Building Value and Cost Leadership*. New York: Free Press, 1997.

Ulrich, Karl T., and Eppinger, Steven D. *Product Design and Development*. 4th Rev. Ed. Burr Ridge, Ill.: McGraw Hill Higher Education, 2003.