Entrepreneur & Innovation Exchange
Published online at EIX.org on March 23 2022
DOI: 10.32617/743-623b30b39005a

How Less Can Be More: Simplicity and Complexity in Product Design

Chris Meyer (Baruch College, City University of New York)
Scott Newbert (Baruch College)

KEYWORDS: Innovation, Product Development, Technology Commercialization.

Apple has become the standard bearer for products that are both technologically complex and simple to use. The Apple TV remote has two buttons and a ring, whereas the typical cable remote has several dozen buttons with different shapes, colors, and, sometimes, more than one function per button. Both let us navigate thousands of video content options, but most of us are happy to use one and dread picking up the other.

TV remotes didn't start with dozens of buttons - the Zenith Space Command was developed in 1956 with only 4.[i] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn1) It was only when technology allowed for a myriad of TV options that we ended up with dozens of buttons on our remotes that we didn't need, understand, or use.

How then are designers and product managers to reconcile the apparent conflict between complexity and simplicity in tech products? And how can less *really* lead to more, with highly complex products that are also simple? Perhaps it's a tradeoff that simply must be managed, as John Maeda, who founded a consortium dedicated to simplicity within MIT's Media Lab, suggested when arguing that the two must be balanced as "necessary rivals."[ii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn2)

This article shows how this Gordian knot comes undone: with a proper understanding of how complexity and simplicity arise. It tells us how less can truly mean more in tech products. In brief: complexity emerges, but simplicity must be enforced. This approach shows how products can be both complex *and* simple and helps us to understand the practical ways in which great products like the Apple TV remote benefit from the combination.

Understanding this will benefit designers, product managers, and even engineers. In addition, this approach also allows for a clear delineation of organizational roles regarding complexity and simplicity, offering guidance to senior managers as they structure their firms.

Is Less Really More?

The phrase "less is more" was made famous by Ludwig Mies van der Rohe, but it originated in a poem by Robert Browning. This poem, *Andrea del Sarto*, is about a renaissance artist with great technical skill but whose art is "empty of inspiration, energy, and religious vision."[iii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn3) Browning uses the poem to illustrate his view of great art: even when what is on the surface is well crafted, if it masks an emptiness underneath, the artist will "fail in doing."

This artistic problem is the inverse of the problem with many tech products, though. Instead of a lovely surface that's lacking underpinnings, there's often a surfeit of stuff beneath the surface that bubbles up and overwhelms the user. Either way, an artist or designer must pay attention to both the surface and to what lies beneath it.

Perhaps the most thoughtful proponent of the "less is more" ethos was Taoist philosopher Lao Tzu. In his *Tao Te Ching*, dating from the 6th century BC, he wrote:

Thirty spokes share the wheel's hub;

It is the center hole that makes it useful.

Shape clay into a vessel;

It is the space within that makes it useful.

Cut doors and windows for a room;

It is the holes which make it useful.

Therefore profit comes from what is there;

Usefulness from what is not there.

As poetic as that is, it's not particularly helpful to simply suggest to product managers that they take things out. And, technology has advanced considerably in the past two and a half millennia, leading to products that are much more complicated than wheels and clay pots, making simplicity tougher to achieve. It's in the realm of technological products that this challenge is the greatest.

It's not that simple products do less or have less utility than complex products, but rather that simple products allow the user to do more with less effort. Designers and product managers need to navigate this predicament in order to deliver great products that are complex enough to solve real problems, but simple enough for people to be able to use effortlessly. Put another way, the concept of "user experience" generally connotes just a pleasant interaction with a product. But a great user experience should not just be a pleasant interaction – it should include valuable, impressive utility. It's the sum total of the product experience that's important and drives adoption.

Yet, there is little theoretical or practical guidance on this front. To move beyond the poetry and philosophy behind "less is more," we need to unpack complexity and simplicity themselves.

Unpacking Simplicity and Complexity – And Where Each Belongs

When the terms complexity and simplicity are used as descriptors, they are opposites. This creates the sense that they're mutually exclusive: something must either be simple *or* complex. Since products are things, it's natural to describe them using adjectives such as beautiful, ugly, practical, useless, etc. But those adjectives merely describe the end state of the product, and design and product development are processes, not states. What's important is to know how to *arrive* at a given state: how to design and develop great products.

That requires a process approach. Per Alfred Whitehead, "It is true that nothing is fully understood until its reference to process is made evident."[iv] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn4) The path to a practical understanding of how less can be more and how products can be both simple and complex, then, is to focus on the *processes* of simplification and complexity. Doing so tells designers where they'll need to make simplifying the focus of their efforts.

Where Does Complexity Come From?

Complexity as a process combines specialization and cooperation, or what Herbert Simon described (using watchmaking as an example) as a "system composed of interrelated subsystems."[v] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn5) This process unfolds as different people specialize in different tasks and then combine their efforts in order to create better, more sophisticated things than they could as individuals. More complex technology and more complex assemblages of technologies demand more specialization, and making all that work demands more cooperation.

A vast academic literature explores how complexity works and what it does. It's known as Complexity Theory, or the theory of complex adaptive systems. Based on work by Herbert Simon (i.e., "The Architecture of Complexity"[vi] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn6)), Murray Gel-Mann (i.e., "Adventures in the Simple and the Complex"[vii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn7)), and many others, Complexity Theory describes complexity as a process in which a number of independent elements organize themselves such that something greater emerges. That self-organizing emergent process is how cells come together to form an organism; how wind, water, and weather come together to form hurricanes; and how traders, technologies, and banks form capital markets. It's also how engineers, designers, and managers come together to create products. Complexity involves disparate elements combining, often without a clear sense of what exactly will result, to make something that is much more than the sum of its parts.

Not surprisingly, the results are emergent: how the various subsystems organize and adapt develops via experimentation and interaction across the entire system. This is a living, breathing, dynamic process. A paradoxical element of all of this is that designers need to understand this process that creates without a designer. Yet, this is the nature of complexity: outcomes that can't be readily predicted emerge as different elements combine.

On a practical basis, the process of building complexity in product development is a matter of increasing technological sophistication and, subsequently, modularity. Advances in research require individuals to go deeper into the technology itself, increasing the number of sub-systems and the needed coordination. With more tech, more elements can go into a given product.

Individuals, teams, and stakeholders coalesce around a given technology or piece of functionality, becoming invested in seeing it placed into any and every product they can, preferably in a prominent spot.

Then, individuals, teams, and stakeholders coalesce around a given technology or piece of functionality, becoming invested in seeing it placed into any and every product they can, preferably in a prominent spot. They're competing with other teams and groups that are pushing their own product element(s).

For example, the number of buttons on remotes exploded with cable TV, as it offered dozens of channels instead of a handful. Cable TV developed the ability to transmit metadata, schedules, and all sorts of other information that viewers never had -- with commensurate functionality for scheduling, recording, and storing programs. When this happened, engineers designing remotes felt compelled to add, and advocate for, dozens of new buttons to access all of that complexity, most of which people never used (or even knew how).

Of course, this excess complexity we see in products today extends well beyond the TV remote and serves to illustrate that as technical specialization increases, so do the number of potential product elements, and that those responsible for each one will (perhaps justifiably) work hard to ensure that their work sees the light of day.

The end result is growing product complexity pushing up through the surface of the product. Products compete based on functional elements, with engineers and marketing personnel pushing to make them seen. The added functionality underneath is a generally good thing, of course. Televisions and video options provide far more utility than they did in 1956. Still, we are left with the question of what to do about all those buttons.

Enforcing Simplicity

If complexity as a process can be a virtuous circle that builds utility and value, what, then, of the need for simplicity? We all want sophisticated tech functionality that's nonetheless easy to use. How can firms deal with the organic growth of complexity so as to give customers the great experience they want?

While complexity can and should be allowed to emerge to drive functionality in the core product, it must be delivered at the interface with simplicity so that the *usefulness* of the product can shine, rather than its complexity. As Lao Tzu saw, profit lies in what is there in the complex core, but the usefulness stems from what is not there at the surface.

Simplicity should be the rule in interfaces between the product and the customer, and that can be done without unnecessarily reducing complexity within the guts of the product itself. Customers want the utility of complex functionality but they don't want to see it or struggle to use it. This requires seeing the two pieces as distinct and recognizing that not everything under the hood has to come poking out.

An apt illustration comes from watchmaking - Simon's example of a complex system. Rolex's signature technology, the "perpetual motion" mechanism, is proudly promoted on its website while being noted as "invisible to the wearer of the watch." Customers appreciate the engineers' work, but they don't necessarily care to see it – they just want to know what time it is.

Customers want the utility of complex functionality but they don't want to see it or struggle to use it. This requires seeing the two pieces as distinct and recognizing that not everything under the hood has to come poking out.

This is where simplicity must be enforced. If the natural tendency of technical systems is *more*, due to emergent complexity and subsystems' stakeholders wanting to show their work, there is no organic or natural force in favor of *less*. Unlike complexity, simplicity isn't emergent. That's where the design challenge comes in.

This design challenge is both an organizational and management challenge. The architect Walter Gropius, who served as the main force behind the Bauhaus School that exerted a significant influence on the design of buildings in the 20th century, argued that "architecture begins where engineering ends."[viii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn8) Engineers specialize and cooperate, such that complexity emerges from their work, delivering the utility and functionality that customers crave. Then, it is up to architects to step in and enforce simplicity by eliminating what is not needed at the customer interface in order to make room for what is.

Simplicity Begins Where Complexity Ends

Gropius' statement highlights the fact that complexity and simplicity largely come about at different places and at different times. Complexity emerges from within, the result of research and engineering work, to create the core product functionality. Simplicity is then enforced where that functionality meets the user – at the interface – by product designers that use the technology provided to them to envision and enact something useful for customers. And, this is typically done after the core functionality has emerged. Thus, we might rephrase Gropius' adage as "simplicity begins where complexity ends." This is no easy task, because complexity is sophisticated and shows a depth of knowledge. It's hard not to show that off. What's new and exciting naturally rises to the surface.

Conversely, simplicity can only be achieved through *discipline*. Harmut Obendorf, in his book on simplicity, reduction, and minimalism, states that reduction "is almost always hard work" because "development processes are often driven by features, internally motivated by the developers… and externally by marketing demands—reduction is hard to sell as a feature."[ix] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_edn9)

Steve Jobs had to battle for years to keep just a single button on the mouse of a Mac.

Indeed, Steve Jobs had to battle for years to keep just a single button on the mouse of a Mac. Simplicity depends on that hard work, though. It requires reduction and battling back against what the engineering and marketing teams want to present so that what is left at the customer interface, and *only* what is left, is useful.

The combined processes are also seen in the Tao, where Lao Tzu notes that simplicity *follows* complexity:

If you want to shrink something,

you must first allow it to expand.

If you want to get rid of something,

you must first allow it to flourish.

It's worth noting that Lao Tzu didn't have to champion complexity, or "more." He knew that it would arise organically. A part of his minimalist ethos was to avoid saying that which didn't need to be said, and he didn't need to encourage complexity – it does fine on its own.

Of course, this is all easier said than done, even for the individuals and companies most committed to and adept at achieving the proper balance between complexity and simplicity. For example, in 1985, Steve Wozniak, the co-founder of Apple, had grown frustrated at how difficult it was for him to manage his sophisticated home theater system. So, he left the company to create a universal remote control. Although Wozniak succeeded in creating a high-functioning product, he was unable to enforce simplicity. Released in 1987, the CORE UC-100 remote had 20 buttons and a 90-page user manual. Not surprisingly, this design proved too complicated for users and ended up a commercial failure.

As this example illustrates, it's hard to say no and it's hard to find places where more is *not* better, even for the best product designers. This is why simplicity must be enforced so that all of the utility does not get lost at the surface. The Tao addresses this, too:

Let your workings remain a mystery.

Just show people the results.

Takeaways

That Taoist wisdom can be put into practice in the service of creating great tech-based products:

1. *Don't be (just) a tech company.* Instead, be a company that creates customer solutions. While such a focus may seem to fly in the face of a culture (such as ours) that privileges technology above all other things, remember that no matter how technologically sophisticated your product is, it can't solve your customers' problems if they can't figure out how to use it.
2. *Guide emergent functionality with strategy*. Products need innovation and change, and good engineering teams come up with new ideas regularly. Firms should encourage that while ensuring that such new ideas are in line with strategic priorities.
3. *Dare to delete*. Enforcing simplicity falls to designers and product managers, who can't be shy about saying "no" and cutting things out if they harm the user experience or obscure the key value generators. Be fearless, even in the face of substantial development efforts and strong advocates from engineering teams.
4. *See and sell the whole*. Excellent products bring sophisticated technology to users in a pleasing, straightforward, effective way. A good user experience delivers both: outstanding utility that is easily accessed. If that can't be readily seen, it can't be readily sold, and the product needs reworking.

---

[i] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref1) Wired staff. 2007. 1956: Zenith Space Commander Remote Control. Wired, October 23. Accessed October 29, 2020 from https://www.wired.com/2007/10/vg-greatestgadget/

[ii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref2) Maeda, J. "The Laws of Simplicity" (Cambridge, MA: MIT Press, 2006).

[iii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref3) Hawlin, S. "The Complete Critical Guide to Robert Browning" (New York: Routledge, 2002, 86).

[iv] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref4) Whitehead, A.N. "Process and Reality: Corrected Edition" (New York: The Free Press, 1978, 64).

[v] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref5) Simon, "The Architecture of Complexity", 468.

[vi] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref6) Simon, H. "The Architecture of Complexity," Proceedings of the American Philosophical Society, no. 106 (1962): 467-482.

[vii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref7) Gell-Mann, M. “The Quark and the Jaguar: Adventures in the Simple and the Complex” (New York: Henry Holt, 1994).

[viii] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref8) Attributed to a Gropius speech in Heyer, P. “Architects on Architecture: New Directions in America” (New York: Walker and Company, 1966).

[ix] (denied:applewebdata://79DE10FE-1D9D-40AD-9B3B-738B6DFDC57F#_ednref9) Obendorf, H. “Minimalism: Introduction and Synopsis” (London: Springer-Verlag), 5.