IfM Centre for Strategy and Performance

YNOT research institute

INCOSE

UNIVERSITY OF CAMBRIDGE Department of Engineering

YNOT Institute
Queens' College Cambridge

Working Paper

The Business Form Factor

Getting from Zero to One the Right Way

Erik Simanis, PhD
IVE Ventures

Tom Manuel
IVE Ventures

Mohamed Khater, PhD
Institute for Manufacturing, YNOT Institute
University of Cambridge

Erika Palmer, PhD
Cornell University
Deputy Technical Director,
International Council on Systems Engineering (INCOSE)

Jeroen Bergmann, PhD
Associate Professor of Engineering Science, Senior Research Fellow in Engineering & Entrepreneurship,
University of Oxford

Manuscript Version: April 25, 2023

"The beginning is the most important part of the work."
(Plato, 375 BCE)

Startups today reflexively follow an agile, customer-driven innovation playbook that solves first for the customer and their needs. As the thinking goes, if you can't get the customer to buy a product on a repeat basis, there's little point in worrying about profits. It's a sensible-sounding approach that was originally developed to keep corporate product development managers from wasting time and money perfecting innovations inside the four walls of the company that customers didn't actually want.

In the startup world, the approach has settled into a three-step process. First, use human-centered design to get insight into customers' needs and problems, and then home in on a general product concept by rapidly testing basic prototypes with target customers. Next, build and launch a stripped down, "minimum viable product" (MVP) into the market so that it can be tested and adapted with real customers spending real money. Lastly, once product-market fit is demonstrated—i.e., you've built up a rapidly-growing base of repeat customers—solve for profitability by experimenting with ways to optimize the business model.

But while engaging customers to come up with and test new product ideas has worked well for established companies in established markets,[1] things have turned out quite differently for startups trying to create new markets—that is, to profitably commercialize disruptive innovations that bring new functionalities into people's lives. Think peer to peer home renting and ride sharing; runway clothing rental for the average woman; meal subscription kits for busy families; space travel.

The most telling statistic is that today's startups are the least profitable at any time over the past 50 years.[2] At the end of 2020, only 6 of 73 "unicorns"— startups achieving a valuation above $1 billion—were profitable. Twenty-one of the 73 had annual losses greater than 50% of revenues and another 13—including Uber, Lyft, Pinterest, and Snapchat—had annual losses greater than 30%.

And it's not a case of companies deliberately accepting losses early-on to capture first-mover advantages. These companies start off and remain unprofitable. Most are more than ten years old, leading to exorbitant cumulative losses. Forty of the 74 amassed losses in excess of half a billion dollars, with Uber taking the top spot with $29-billion of accumulated deficits. Today's economic climate and the disappearance of easy money are, at last, spot-lighting this issue.

The pattern, however, is widespread across the startup landscape, not just unicorns. Most of the companies profiled in The Lean Startup—one of the first books to adapt the agile, customer-driven innovation approach to entrepreneurship—have been shuttered (including Aardvark, Food on the Table, and Votizen), acquired for lack of profitability (Grockit), or are still struggling to reach profitability (Dropbox). It's why Director of the MIT Innovations Teams Program Luis Perez-Breva

---

[1] Change, W. and Taylor, S. (2015) The Effectiveness of Customer Participation in New Product Development: A Meta-Analysis. *Journal of Marketing*, 80(1), pp. 1-22.

[2] For excellent overviews and analyses of the declining profitability of startups over the past decade, please see:

i. Mackintosh, P. (2021) *What Happens to IPOs Over the Long Run?* NASDAQ Economic Research. Available at: https://www.nasdaq.com/articles/what-happens-to-ipos-over-the-long-run-2021-04-15 (Accessed 15 Nov 2021).

ii. Funk, J. L. (2021) *Most Unicorn Startups Will Not Overcome their Losses*, Personal Blog. Available at: https://jeffreyleefunk.medium.com/most-unicorn-startups-will-not-overcome-their-cumulative-losses-ecbe7133cf26 (Accessed 15 Nov 2021).

iii. Funk, J.L. (2021) *Only 6 of 73 Unicorn Startups are Profitable, and None Did Recent IPOs*, Personal Blog. Available at: https://jeffreyleefunk.medium.com/only-6-of-73-unicorn-startups-are-profitable-and-none-did-recent-ipos-287d5c7ac8d0 (Accessed 15 Nov 2021).

recently commented that ten years' experience has shown that the various incarnations of the agile, customer-drive approaches to innovating new ventures "…are an expensive recipe for remaining an entrepreneur longer, not really for innovating."[3]

Having worked at the intersection of disruptive innovation research and practice for two decades, and collectively led or guided over two dozen new corporate ventures and startups, we've uncovered why the agile, customer-driven innovation approach works for established companies in established markets, but generates the opposite effect for entrepreneurs trying to create disruptive markets: it requires the presence of a commercially-viable business form factor. In established markets, you have one; in the startup world, you don't.

A business form factor is the high-level business idea an innovator has in mind. It's a mash-up of the product form factor and the operational model into a single concept. It describes the basic "shape" of a business—i.e., what it does for the customer and how it does it, like "block-chain enabled credential verification for expatriates" or "online short-term rental of designer clothing for women." To borrow from serial entrepreneur Peter Thiel, the business form factor is what gets you from zero to one and establishes a new venture's starting point.

In this article, we explain, why the business form factor is so consequential for startups, and how today's agile, customer-driven innovation playbook sets new ventures up to fail. Drawing on key insights and practices from model-based systems engineering, we introduce a three-step process for innovating what we call "robust business form factors"—ones that create a wide path to profitability and significantly boost the potential for ending up with a profitable venture.

The Business Form Factor

When a profitable market already exists, innovators are working—knowingly or not—with key product specifications given by the business form factor. These specifications establish design guardrails within which the product must be innovated in order to be made, sold, and delivered profitably.

So, when GE's Appliance's FastWorks team applied agile, customer-driven innovation to create a refrigerator with French doors for their premium "Monogram" line, there was a lot about the product that was already established by the proven "compressor-technology refrigeration appliances for domestic use" business form factor.

The refrigerator's basic dimensions, weight, and positioning of key components were set by the size of residential home's entry doors, the standard size of kitchen counters, and what installers could handle. There was a baseline in terms of interior capacity, shelving, and cooling performance set by the kinds of foods typically stored by households.

The price range acceptable to the consumer for in-home food refrigeration was already socially ingrained, and constrained further by competitors' offerings. And the allowable cost of the refrigerator's components was largely given, as the unit-level manufacturing overheads, required above-the-line marketing and promotional material, shipping costs, and retailer margins for an established white goods appliance were largely set.

As long as the FastWorks team worked within these parameters, all insights gleaned through deep customer needs analysis and rapid customer prototyping could be translated into a refrigerator that customers loved and that GE Appliances could profitably make and sell.

---

[3] Perez-Breva, L. (2018) *"The Lean Startup" is an unproductive legend*. Quartz at Work. Available at: https://qz.com/work/1349238/the-lean-startup-is-an-unproductive-legend (Accessed: 15 Nov 2021).

The same conditions were at play in the famous milkshake case study detailed by the late disruptive innovation guru, Clay Christensen.[4] Christensen explains how an innovation team used human-centered design to understand the specific "job" that customers of a fast-food chain implicitly "hired" a milkshake to do. The team discovered that a large group of customers coming for a quick meal en route to work wanted the milkshake to reduce the monotony of the morning commute, leading to the recommendation that the milkshake be made thicker with chunks of fruit to further slow its consumption and add surprise and delight.

Once again, though, the team wasn't working from a clean slate. The proven "fast-food restaurant" business form factor set design guardrails within which the new-and-improved milkshake solution had to fit, including its basic format (ready-to-consume, single-use portion dispensed into a cup), it's required price range, and the allowable cost of all ingredients and packaging, as the margin needed to cover the restaurants overheads was set.

With startups, there is no commercially-proven business form factor to provide any of these critical design guardrails. But you can't build an MVP without having one. After all, what are you going to create an MVP of? There has to be a starting point. And there is a right way and wrong way to do it.

Startups using an agile, customer-driven approach unwittingly do it the wrong way—by defining it based on how best to meet the customer's wants, needs, and jobs. Setting the business form factor in this way harbors a fatal flaw that every systems engineer would spot immediately: designing one part of a system to meet a single requirement makes the overall system underperform and be significantly more costly, over 50% higher according to research[5]. To use a phrase popularized by physicist and energy efficiency pioneer, Amory Lovins, "optimizing a part 'pessimizes' the larger system."[6]

It's a critical point: businesses are complex systems—a collection of interacting parts that individually perform simple functions, but which give rise to new, complex functions when arranged in a particular way. A business brings together a dizzying number of otherwise lifeless "business parts"—product features, packaging, branding strategies, marketing mixes, pricing tiers, payment terms, credit risk teams, raw material suppliers, warehouses, sales people, and so on. It "comes to life" when that pattern of parts solves a customer problem and gets them to pay a price that generates competitive returns on capital.

Trying to build up a new, complex system part by part hampers its effectiveness and creates high inefficiencies because, once the first part is designed, it constrains how a system's complex, functions can be met, while also limiting the options available for the next part. As more parts are added to solve issues created by the previous parts, the costs and constraints keep adding up.

---

[4] Christensen, C., Hall, T., Dillon, K., and Duncan, D. D. Know Your Customers' 'Jobs to Be Done'. *Harvard Business Review*, 94(9), p. 14.

[5] E. C. Honour, Systems engineering return on investment (SE-ROI), Results from 15 years of research, with data from 94 programs, www.hcode.com/seroi/

[6] "Lovins, A. B., Lovins H. L., and Hawken, P. A Road Map for Natural Capitalism. *Harvard Business Review*. July-August, 2007.

An extreme example of the clumsiness and inefficiencies created by building a system up part by part is the "Rube Goldberg Machine"—humorous contraptions consisting of a series of parts where the action of one triggers the next one to do something, with the final part fulfilling the system's ultimate function.

Take, for example, Rent-the-Runway, an online renter of designer clothing to women that tipped into unicorn status in 2019 but has accumulated over $800 million of losses in 11-years of operation with reported losses of $212 million in its most recent fiscal year. Starting off with "online, short-term rental of designer clothing" as a business form factor created very challenging cost, customer adoption, and competitive barrier problems, while also limiting the options available to solve them.

It created the problem of holding numerous sizes of a rapidly depreciating good, as runway-styles change each year. It created the problem of allowing customers to order multiple sizes and make temporary alterations, as customers would otherwise doubt whether the gown seen on the runway would fit. It created the problem of paying large slices of revenue to prized designers that drew in customers, as they could otherwise sell through prestigious retailers.

To solve these problems, Rent-the-Runway maintains huge inventories, climate-controlled warehouses, large numbers of employees managing fulfillment, a large reverse logistics team, shipping partnerships, royalty fees, and more. And each of these parts invariably trigger their own "end of pipe" problems, requiring additional parts—flagship bricks and mortar stores to woo prestigious designers who want to associate their brand with an equally prestigious brand; drop-boxes in areas where a shipping partner doesn't have a retail presence; and security measures like RFD tags to prevent theft of inventory. The parts and costs keep piling up.

An ineffective, high-cost starting point may sound undesirable, but not a deal killer. After all, the agile, customer-driven startup approach begins with an MVP precisely to preserve flexibility, with the expectation that you can optimize the business model later.

While that sounds plausible, it harbors a second fatal flaw. The other key property that shapes how systems engineers innovate new systems is that approximately 70% of a system's life-cycle costs are fixed once a core concept is defined, with an additional 15% locked-in once the core design is specified.[7] It's yet another unique consequence of a system's functions arising from the interaction of its parts.

In other words, the moment Blue Apron founders Matt Salzberg, Mathew Wadiak, and Ilia Papas defined their business concept as "a meal subscription kit company that mails recipes and pre-measured and pre-cut fresh ingredients direct to people's homes," the company had already locked-in 70% of its cost structure. Once they pulled the trigger on an initial offering of three

[7] Sheldon, D. F., Perks, R., Jackson, M., Miles, B. L., and Holland, J. (1990) Designing for Whole Life Costs at the Concept Stage, *Journal of Engineering Design*, 1:2, 131-145, DOI: 10.1080/09544829008901649

recipes packed and shipped out of a rented commercial kitchen in Queens, NY, 85% of Blue Apron's cost structure was unavoidable.

An ineffective, high-cost business form factor that simultaneously locks-in 70% to 85% of a venture's cost structure is a precarious foundation on which to build a startup, as it significantly narrows the path to profitability and the probability of success.

Worst case scenario, it might completely close it off from the very beginning, trapping the startup on what Michael Porter calls a "productivity frontier"[8] where even the most optimally-configured product and operational model cost more than the value created for customers. In economics terms, the business can't create surplus value—i.e., make and sell a product that generates more value for customers than the whole cost of the product, including cost of capital. It's the most basic requirement for commercial viability.

Consider Blue Apron's experience. Launched in August 2012, the company has toiled to uncover a profitable business model for the direct-to-consumer subscription meal kit business form factor. To acquire customers, they've expanded to 50 weekly meal options catering to all kinds of dietary needs; partnered with well-known chefs; re-designed recipes to make them easier and quicker to prepare; launched ready-to eat options; tried a low-cost option that requires customers to supply their own protein; created Weight Watchers approved meal plans; kicked-off a Wellness360 campaign providing benefits to customers of Blue Cross Blue Shield, Planet Fitness and Calm; and created third-party sales partnerships with Jet.com, Walmart Marketplace and Amazon.

On the operational side, they've opened an automated distribution center that cut their East coast fulfillment workforce from 2,000 to 800; rolled out an order management system throughout all of its warehouses; added a shipping fee onto customer orders; sourced more ingredients directly from farmers and ranchers; and even tried retailing meal kits inside Costco stores.

Despite all their efforts, the company remains deep in the red, generating a net loss of $26 million in the third quarter of 2022. The company's accumulated losses since launch are close to $700 million.

To be clear, this isn't a Blue Apron problem. Aside from a temporary Covid lockdown blip, almost every subscription meal kit company is struggling to achieve sustained profitability. Even Hello Fresh, the far-and-away market leader, which has almost 60% market share in the US and unrivaled economies of scale, saw its lockdown-boosted profit margins of 6% drop precipitously below 2% in 2022. Free cash flows for Hello Fresh's first quarter of 2023 are forecast to be back into negative territory.[9]

In other words, when scores of talented entrepreneurs, investors, and managers can't make a business model work after a decade of trying, it's very likely that a flawed business form factor is to blame— i.e., one that's imported a baseline cost structure that prevents a business from ever creating surplus value.

It explains why profitability remains elusive for so many "disruptive industries" today, despite years of experimentation to optimize the business model: from app-based food delivery (e.g, Deliveroo, DoorDash, Uber Eats), peer-to-peer car sharing (e.g., Turo, Drive Not, car 2 go), online women's designer clothing rental and personal shopping (e.g., Rent the Runway, Le Tote, Stitch Fix), to buy-now-pay-later consumer financing (e.g., Klarna, Affirm, GreenSky, and Afterpay).

---

[8] Porter, M. E. (1996) What is a Strategy, *Harvard Business Review* (1996): 61-78.

[9] *HelloFresh: Weak Q1 2023 Guide, Huge Acceleration Needed To Hit FY23 Guide*, Seeking Alpha, March 8, 2023. Accessed at: https://seekingalpha.com/article/4585628-hellofresh-weak-q1-2023-guide-acceleration-needed-hit-fy23-guide (Accessed 30 May 2023).

And when this problem is married up with today's focus on first rapidly growing a company and demonstrating product market fit before worrying about profitability, the eye-watering, billion-dollar accumulated losses typified by Uber, Snapchat, and Pinterest will follow.[10]

So, how do you innovate a novel business form factor that has a high chance of profitably making and selling a product that customers buy? The short answer is that it has to be designed holistically, taking all of a business's functions into consideration—so, not only what satisfies customers wants and needs, but also what's needed to profitably make, sell, deliver, and get paid for the product.

While that sounds pollyannish, given the dizzying number of parts that come together in creating a new venture, it's realistic and doable. But it requires solving for a business's core functions at a broad, first-principles level where complexity and uncertainty are far lower, and only then adding in detail once key design parameters are set. It's a very different way of thinking about and addressing the seemingly-intractable uncertainty that entrepreneurs face.

But the approach isn't new. Systems engineers have been applying it for over 70 years to systems far more complex than a business and under conditions of even greater uncertainty.

Taking Inspiration from Systems Engineering

Systems engineering is a branch of engineering uniquely focused on innovating new, complex systems. It was first taught as a discipline in 1950 at MIT, and is used today in applications that range from aerospace, robotics, and software design to the construction of bridges, buildings, and transport systems. System engineers work from a blank sheet of paper to create new systems—new configurations of parts that perform functions that current systems are unable to do, like the International Space Station.[11]

The end-goal of systems engineering is an "elegant system." Simply put, an elegant system is the simplest one—i.e., it gets the system to work using the fewest parts. Systems engineers strive for an elegant design because they are more reliable, more robust, and less costly. They're more reliable because there's less that can go wrong, and they are more robust because there are fewer "weak links" that can fail. Elegant designs have lower costs, as there is less stuff to prototype and validate and get working together properly, and then less stuff to buy, replace, and manage once operational.

Systems engineers know that the only way to end up with an elegant system rather than an ineffective, high-cost Rube Goldberg machine is to start with an elegant design. By design, they don't mean its aesthetics, but how the thing fundamentally works—it's core strategy. And the key to an elegant design is solving holistically for a system's functions by taking into account all of the things a system has to do when coming up with the design.

Why? Solving holistically creates the opportunity for synergies, or "multi-purpose" parts that perform or support multiple system-level functions. In other words, design synergies let you "delete parts," as Elon Musk famously describes in an interview on the Everyday Astronaut YouTube channel.[12]

---

[10] Simanis, E., Manuel, T., Khater, M., Palmer, E., and Bergmann, J. (n.d.) *When Cash is King: A Systems Engineering Approach to New Venture Creation*. YNOT Institute Working Paper, University of Cambridge.

[11] For an introduction to systems engineering, please see the following handbook prepared by the International Council on Systems Engineering (INCOSE): Walden, David D., et al., editors. Systems Engineering Handbook: A. Guide for System Life Cycle Processes and Activities, 4th Edition. John Wiley, Hoboken, NJ: 2015.

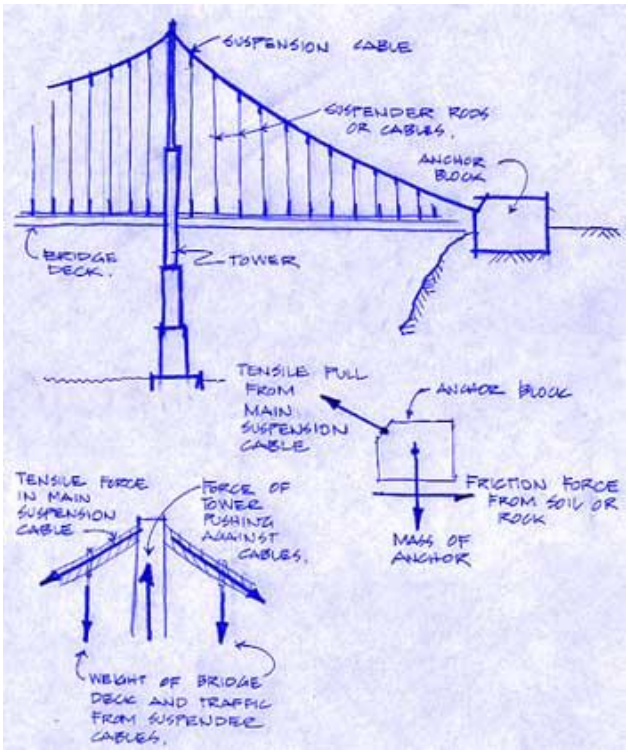
[12] *Starbase Tour with Elon Musk [PART 1 // Summer 2021]*, August 3, 2021: Accessed at https://www.youtube.com/watch?v=t705r8ICkRw (Accessed 30 May 2023).

A basic, but evocative, example of holistic design synergy is the "spork"—the cutlery innovation that performs the job of both a spoon and fork. By solving for the functions of a fork and a spoon simultaneously, 50% of the parts are eliminated.

Solving holistically for a spork that performs two very basic functions is one thing. Doing so for a complex system like the International Space Station that performs thousands of micro functions and consists of tens of millions of interacting parts, is another. It doesn't seem humanly possible—and it isn't.

System engineering's unique insight is that you can eliminate complexity and still solve holistically by "levelling up"—that is, by first solving for the system at the architectural level where there are relatively few, "high-level functions" to address. Because, while every one of a system's detailed parts does a very specific, focused job (e.g., a bolt secures a steel plate to a concrete base), they all ultimately ladder-up in support of a handful of "first-principles" functions that govern how something works at its most basic level.

So, if you're building a complex bridge, you first figure out a basic, but still complete, bridge design that most elegantly solves for the main physical forces it will have to endure—the compression and tension forces created by the weight of what it carries, and the torsional and shear forces caused by winds, water, seismic activity and the distance it has to span—in addition to key human-imposed requirements, like budget. The high-level solution would look something along the lines of the suspension bridge design below. Once you're confident it works at that level, you break it down into smaller sections or subsystems, and solve for more detailed functions—i.e., the functions of the functions.



The key thing is that the high-level design establishes basic design guardrails for the subsystems, as subsystems are doing more detailed jobs that ladder-up in support of first-principle functions.

So, for our suspension bridge, the next step would involve solving for the basic design of its subsystems: the bridge deck, the towers, the anchor blocks, the suspension cables and the suspender rods. The systems engineer would aim to answer the question, "given the functions that the bridge deck plays in the high-level suspension bridge architecture, what are the key requirements that it has to meet in order to fulfill these roles?"

Once the sub-system level is solved, you break it down into its sub-systems and follow the same approach, ultimately getting down to detailed functions and requirements played by individual nuts and bolts. Through these nested levels, every detailed part will have design guardrails that ultimately connects them to back to the system-level architecture.

By designing top-down in this way, you're able to solve a complex system holistically and ensure that individual components work optimally and synergistically as part of the system.

It's the very process behind famed architect Frank Gehry, the architect behind iconic, never-before-attempted structures like the Guggenheim Museum Bilbao and the Walt Disney Concert Hall, Los Angeles, California. In designing his buildings, Gehry works in stages, moving "smoothly from big ideas at the grand scale (what about a skyscraper twisted like licorice?) to increasingly fine-grained details (How do we design this window?)."[13]

But there's another huge benefit from the systems engineering approach. Apart from ensuring individual parts work synergistically, levelling complexity allows design synergies to be baked into the system architecture where they exert an oversized impact on its effectiveness and efficiency, as they eliminate big chunks of parts all at once.

Amory Lovins, for example, demonstrated how solving holistically for a building's "system architecture" could generate enormous energy savings, reduce costs, and improve comfort simultaneously. The company's legendary headquarters in Snowmass, Colorado eliminated all heating systems by combining 16-inch-thick walls (compared with conventional six-inch walls), krypton-filled super windows, walls and floors that function as solar heat sinks (walls and floors that absorb solar energy and release it after sunset), and a southwestern orientation. The facility houses a greenhouse warm enough to grow bananas.

When systems engineers innovate new, complex systems, the beginning is truly the most important part of the work.

Applying Systems Engineering to Venture Creation

Over the past decade, we've worked closely with approximately two dozen new venture teams at global corporations including Pearson, Barclays, and Procter & Gamble, as well as several startups, to apply model-based systems engineering to new venture creation. As part of the methodology, we've refined a three-step process to architect new, robust business form factors. The architecting approach increases the monetizable opportunity targeted, while driving down the cost structure. Or, to use a golfing analogy, it expands the size of the cup while making the ball smaller. It's how robustness is engineered.

Step one involves stripping an idea back down to its essential, "core functionality" to avoid prematurely importing unnecessary design constraints. Step two uses the core functionality as a lens to isolate the "prime use case/s"—i.e., the broad application/s of the core functionality that hold/s the greatest urgency for customers and has a large membership. Step three architects the business form factor by solving for 10 first-principles business requirements, and then productizing the resulting 10 "theories of change" into an elegant business form factor.

To be clear, once a business form factor has been created, it isn't simply assumed to work. The complete methodology follows the model-based systems engineering technique of modeling, simulating and stress-testing a design at each level of detail before proceeding on to the next. The methodology then uses a bottom-up validation process to make sure each "business part" does its job before the parts and subsystems are brought together into a live pilot test.[14] A detailed description of the methodology, along with the modeling and simulation tools, are available at www.seivc.org.

---

[13] Flyvberg, B. and Gardner, D. (2023) How Frank Gehry Delivers On Time and On Budget. *Harvard Business Review*, January-February.

[14] For an overview of the methodology, please see: Simanis, E., Manuel, T., Khater, M., Palmer, E., and Bergmann, J. (n.d.) *When Cash is King: A Systems Engineering Approach to New Venture Creation*. YNOT Institute Working Paper, University of Cambridge.

Step 1, Define the Core Functionality

"Core functionality" is what a product or new technology does (its utility) and how it does it (its form) at its most basic level. It's what ultimately performs the work that customers use to achieve an outcome. A car's core functionality, for example, can be defined as "mobility via an engine-powered enclosed cabin." A motorcycle's core functionality, by comparison, would be "mobility via an engine-powered bicycle."

Core functionality is the best starting point for designing a business form factor, as it provides the minimally necessary information to identify customer applications for which it is uniquely suited—and, therefore, where it can generate the biggest bang for the buck and have a leg up on competition. It also strips away premature design elements that an entrepreneur may have unwittingly imposed.

The goal is to articulate the core functionality with the least amount of resolution needed to define use cases effectively—the focus of Step 2. Because of that, it's best to approach Steps 1 and 2 iteratively, "testing-out" different definitions of core functionality and the different use cases they surface.

In the case of an energy-related startup we advised that generates energy through high altitude kites. we found the best definition for its core functionality to be "70 kilowatts of energy generated through a truck-sized airborne kite flown at high altitude." In the case of a food company that was keen to target nutritional deficiencies using a core extrusion technology, we settled on a core functionality definition of "delivers precise micronutrient dosages through extruded food." And for a new venture team at a bank that was trying to disrupt the credit card industry, we defined its core functionality as "access to credit at secured-loan interest rates through unsecured lending."

Step 2, Isolate the Prime Use Case

The prime use case (PUC) is the broad application for the core functionality shared by a group of people or organizations that holds the greatest value potential. Defining the PUC ensures the venture focuses from the outset on the target opportunity with the greatest chance of success. It also ensures customer research proceeds strategically, rather than opportunistically.

The key to listing out "good" use cases is to define them based on the core functionality's unique attributes. Take the kite-generated energy example above. One of the things that makes the core functionality unique is the quantity of energy generated—70 kilowatts of energy can power about 70 modern homes. Power generated by capturing high-altitude wind is also very consistent. Air currents don't fluctuate like they do at the ground level where windmills operate. Third, the fact that it is a truck-size unit means it's quite mobile.

So, what are different applications where these three dimensions come together? A business-to-government use case could be humanitarian and military operations that otherwise would be using diesel-powered generators. A business-to-business application could be temporary or mobile commercial events, like concerts, festivals, fairs, and circuses. A consumer application could rural villages in emerging and developing economies that lack grid access.

Use cases are then evaluated according to the size of their membership, and the degree of importance or urgency it holds for the user. These two parameters broadly establish the value potential of a use case, as the size of the use case impacts the size of the "total addressable market" (TAM) and potential unit sales, while the degree of importance impacts price and willingness to pay.

Finding estimates for the total addressable market is a matter of desk research. The urgency assessment is qualitative, and requires reflecting on the customer outcome supported by the core functionality and the routines they currently have. We rate urgency as either green, yellow, or red,

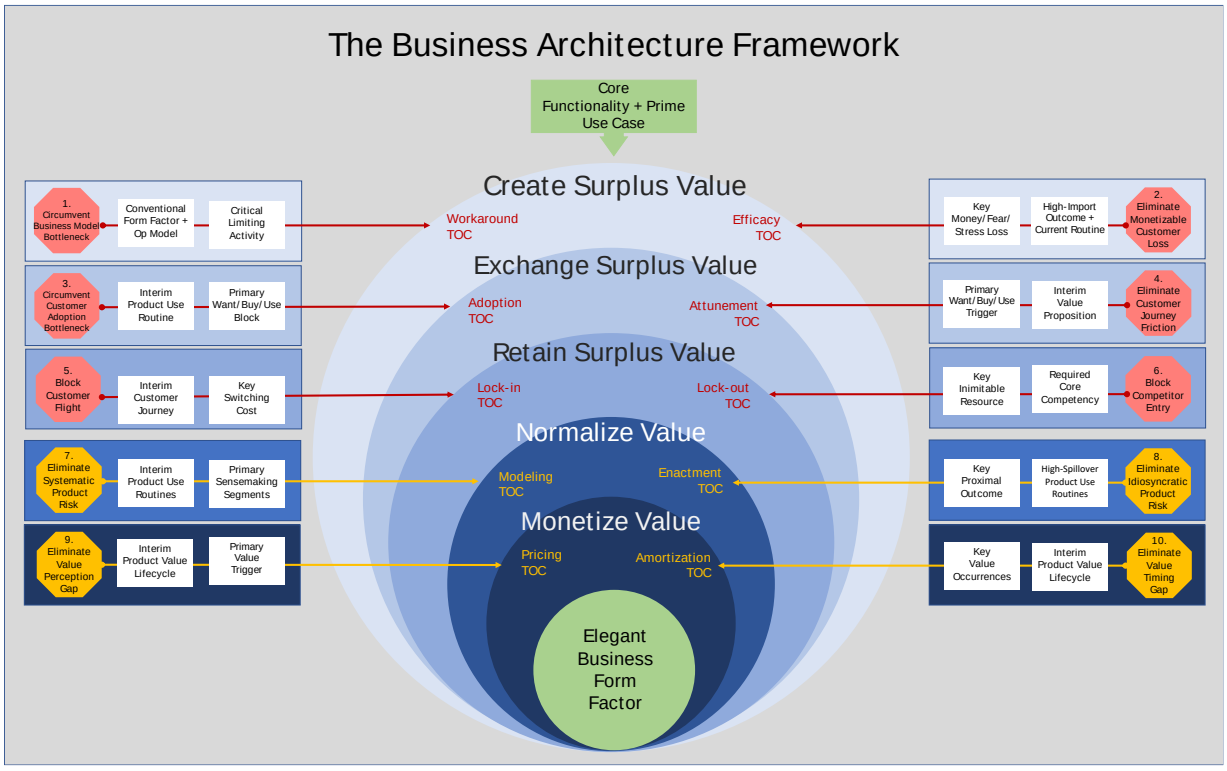
where green is “nice to have” and red is “mission critical.” For example, if a military or humanitarian relief effort were to lose power, the results would be life threatening; consequently, a huge amount of effort goes into securing supply chains. For villagers using kerosene, battery-powered lamps, and wood or propane to cook, the presence of consistent electricity would certainly improve life, but it wouldn’t be a life-or-death matter.

Remember—the goal is to figure out where to focus attention, so the required level of precision needed at this stage is relatively low.

Step 3, Architect a Robust Business Form Factor

Step three is where systems engineering’s levelling up strategy comes into play. In order to level-up, you need a "logic framework"—a map of the cause-effect relationships that explain how a complex system works at its most basic, first-principles level. Through our research and in-field work, we’ve developed a logic framework for business that we call the Business Architecture Framework, or BAF (see below).

The BAF maps out 10 requirements that govern the basic commercial potential of a new venture, and the key information or inputs needed to solve those requirements. Below we’ll overview the basic structure of the BAF and how to use it to create a robust business form factor.



The BAF is made of five nested circles representing five high-level functions critical to every new-to-the-world venture: the creation, exchange, and retention of surplus value, and the normalization and monetization of the new-to-world customer value created. Solving the first three functions—the creation, exchange, and retention of surplus value—establishes the potential for commercial viability. Solving the last two—normalizing and monetizing customer value created—expands that commercialization potential.

Each function is defined precisely:

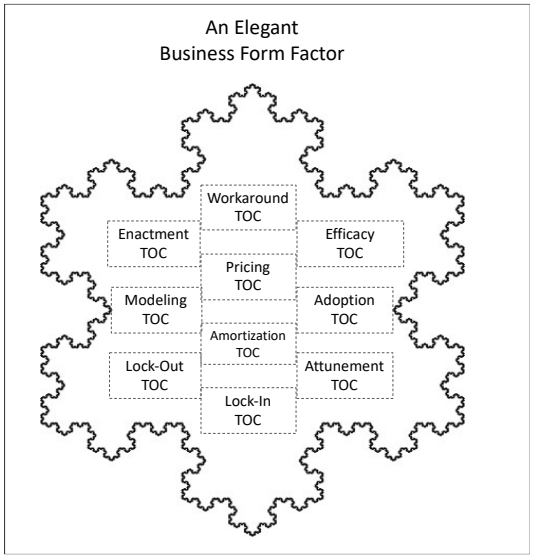
- Create surplus value means making a solution that generates more value for customers than the whole cost of the solution, including return on capital.
- Exchange surplus value means convincing a customer to pay a price for the solution that is equal to or greater than the product's whole cost.
- Retain surplus value means holding onto a portion of what customers pay that is equal to or greater than greater than the whole cost of the solution.
- Normalize customer value means getting customers to marry the new core product routines with their idiosyncratic life routines such that net product value is greater than the price.
- Monetize customer value means getting customers to marry the new product spend with current spend such that total expenditures are equal to or less than the customer's total budget.

As the definitions reveal, the five functions are nested. You can't normalize and monetize customer value, until you first create, exchange and retain surplus value. At the same time, you don't actually know if you've created surplus value until you know the full cost of the "parts" needed to exchange and retain surplus value, and to normalize and monetize customer value.

The two boxes flanking the nested functions are the work that needs to be done to in order to solve the functions. They contain two things: the two most essential requirements that underpin each of the functions—i.e., the biggest problems to be solved—and two key inputs that govern how each requirement should be solved. It's the logic part of the logic model.

The solution for each requirement takes the form of a "theory of change," or TOC (e.g., 1. workaround TOC, 2. efficacy TOC). Think of "theory of change" as the reasoning for how a requirement is best fulfilled, given what's been learned about the precise nature of the problem.

The goal is to "productize" the ten TOCs into an elegant business form factor. Productizing these three functions brings a product form factor and a business operational model together into a single unified entity. In so doing, it simultaneously molds the high-level shape of the product and operational model such that they fit together seamlessly. It's like designing the key hole, which then drives the shape of both the lock and the key.



A solution is elegant when the ten TOCs come together into a single, integrated concept. So, it shouldn't look like a Swiss army knife containing ten "single-purpose" tools.

A powerful example of an elegant business form factor comes from microfinance—an industry that today provides loans and financial products to people who lack access to conventional banking. Prior to microfinance, small holder farmers and petty traders across the developing world could only access small working capital loans via local loan sharks at often-times usurious rates

Pioneered by Noble Prize winner Muhammad Yunus and his Grameen Bank in Bangladesh, the business form factor that underpinned the early micro-finance business models could be described as "a small business peer group loan issued in tranches to a self-formed team of community members that share mutual loan responsibility and meet regularly as a group to make repayments and discuss each other's business status." So, the loans aren't given to individuals,

but to groups of five to ten borrowers. Group members only receive their next working capital tranche if all other members have paid back their individual loan.

The “small business peer-group loan with mutual loan responsibility” was a truly elegant innovation.

- It eliminated the significant costs of conventional due diligence that prevented the conventional banking model from profitably serving these customers (Create Surplus Value).
- It harnessed peers to recruit other customers, thereby overcoming the stiff marketing challenge of selling financial services to people who are illiterate and innumerate (Exchange Surplus Value).
- The interdependence and solidarity forged through the group made it difficult for competitors to siphon off customers (Retain Surplus Value).
- Regularly meeting as a group with the loan collection officer, typically in a public location where other non-customers could watch, provided a way for the otherwise very novel product routine to become relatable (Normalize Customer Value)
- Selling loans in a “group tranche” and repaying on a weekly basis as a group enabled members to jump in and learn the new budgeting routine needed to become a customer (Monetize Customer Value).

Because of the robustness of this business form factor, the micro-finance industry has grown explosively. The global market is today valued upwards of $200 billion and growing at a rate of more than 10% annually. And yes, they are profitable. Mexican microlender Compartamos, the largest microlender in Latin America, posted a return on equity in 2019 and 2021 in excess of 20%.

Conclusion

When start-ups fail, more is lost than simply the prospect of new, life-enhancing products. A failed start-up means lay-offs, stranded customers, tax-payer money diverted from other pressing needs, and environmental resources consumed for naught. So, while the prospect of failure can never be eliminated, we owe it to society to invest in building an entrepreneurship practice with the greatest possibility for success—the goal should be to obsolete pivots and failures, rather than treat them as an entrepreneurial rite of passage. We hope this introduction to the unique power of systems engineering to solve the challenge venture creation helps serve this cause, and bring about a new, profitable generation of disruptive innovations.

References

Change, W. and Taylor, S. (2015) The Effectiveness of Customer Participation in New Product Development: A Meta-Analysis. *Journal of Marketing*, 80(1), pp. 1-22.

Christensen, C., Hall, T., Dillon, K., and Duncan, D. D. (2016) Know Your Customers' 'Jobs to Be Done.' *Harvard Business Review*, 94(9), p. 14.

Honour, E. C. (n.d.) Systems engineering return on investment (SE-ROI), Results from 15 years of research, with data from 94 programs. Available at: www.hcode.com/seroi/

Flyvberg, B. and Gardner, D. (2023) How Frank Gehry Delivers On Time and On Budget. *Harvard Business Review*, January-February 2023.

Funk, J. L. (2021) *Most Unicorn Startups Will Not Overcome their Losses*, Personal Blog. Available at: https://jeffreyleefunk.medium.com/most-unicorn-startups-will-not-overcome-their-cumulative-losses-ecbe7133cf26 (Accessed 15 Nov 2021).

Funk, J. L. (2021) *Only 6 of 73 Unicorn Startups are Profitable, and None Did Recent IPOs*, Personal Blog. Available at: https://jeffreyleefunk.medium.com/only-6-of-73-unicorn-startups-are-profitable-and-none-did-recent-ipos-287d5c7ac8d0 (Accessed 15 Nov 2021).

HelloFresh: Weak Q1 2023 Guide, Huge Acceleration Needed To Hit FY23 Guide (2023) Seeking Alpha, March 8. Available at: https://seekingalpha.com/article/4585628-hellofresh-weak-q1-2023-guide-acceleration-needed-hit-fy23-guide (Accessed 30 May 2023).

Mackintosh, P. (2021) *What Happens to IPOs Over the Long Run?* NASDAQ Economic Research. Available at: https://www.nasdaq.com/articles/what-happens-to-ipos-over-the-long-run-2021-04-15 (Accessed 15 Nov 2021).

Perez-Breva, L. (2018) *'The Lean Startup' is an unproductive legend*. Quartz at Work. Available at: https://qz.com/work/1349238/the-lean-startup-is-an-unproductive-legend (Accessed 15 Nov 2021).

Porter, M. E. (1996) What is a Strategy, *Harvard Business Review*, November-December 1996, pp. 61-78.

Sheldon, D. F., Perks, R., Jackson, M., Miles, B. L., and Holland, J. (1990) Designing for Whole Life Costs at the Concept Stage, *Journal of Engineering Design,* 1(2), pp. 131-145.

Simanis, E., Manuel, T., Khater, M., Palmer, E., and Bergmann, J. (n.d.) *When Cash is King: A Systems Engineering Approach to New Venture Creation*. YNOT Institute Working Paper, University of Cambridge.

Starbase Tour with Elon Musk [PART 1 // Summer 2021] (2021) [Video] Available at: https://www.youtube.com/watch?v=t705r8ICkRw (Accessed 30 May 2023).

Walden, D. D., et al. (2015) *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th Edition. John Wiley, Hoboken, NJ.

YNOT Institute

Queens' College Cambridge
ynot@queens.cam.ac.uk | www.ynotinstitute.org