

virtusa

WHITE PAPER

Experience-led Engineering

It's not about being design-led. As Steve Jobs espoused, "You've got to start with the customer experience and work backward to the technology."

Too often, teams rush to start building before they know that they're building the right thing.

"Experience-led engineering combines a passion for designing useful, usable, and desirable experiences with rigorous engineering discipline. Useful apps contain the proper functionality necessary for users to achieve their goals. Usable apps have a short learning curve and are appropriately easy to use. And desirable apps contain carefully crafted experiences that elevate them above the norm."

Have you ever noticed that once a software project commences, the team is always eager to begin coding right away? Too often, this is the case, and teams begin implementing before there's clarity about what success means. Few teams take the time to define a "north star," a visual representation of product concept ideas that the team can rally around. It's a given that people have different pictures in their heads and expectations about what the future looks like, especially early in a project. By exploring conceptual ideas early, we can pull ideas out of people's heads and into a tangible form that people can react to.

A north star, which most often takes the form of one or more prototypes, provides a few important benefits:

Prevents the "illusion of agreement," where people think they're aligned and understand what the project team will deliver, only to find out down the road that they had different mental models and weren't aligned at all

Preserve the integrity of the concept from the inevitable "death by a thousand cuts" that over time water down the original concept

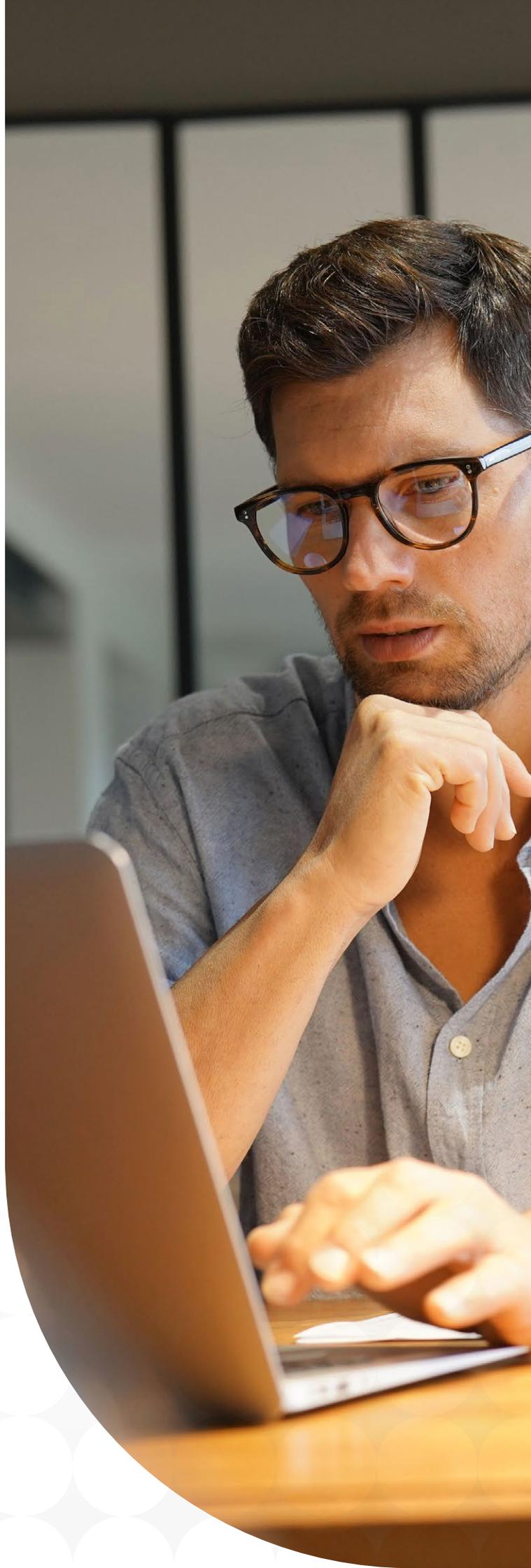
Help "sell" the idea internally to obtain funding and resources

Get customers and users excited and interested in participating

But isn't a sprint zero intended to help with this? After all, during sprint zero, teams try to get their arms around the requirements. A sprint zero typically focuses on functionality and requirements, not the experience the app is intended to support. You might think that this suggests that "engineering-led design" should be replaced with "design-led engineering," but the **design is an approach to achieving the goal, not the goal itself. Rather, a more appropriate term is "experience-led engineering,"** which puts the emphasis where it belongs — on the outcome we're trying to achieve and the experience we're trying to create. Engineering should be in support of the experience. When done well, business benefits will follow.

Designing such a system requires that engineering, design, and product management collaborate. Remember, technical choices impact user experience and vice-versa. Technical choices— such as the structure of the database, the performance of the servers, and speed of the networks — enable or constrain the experience, while design decisions influence technical choices. By collaborating, we can balance user needs, technical/operational feasibility, and business viability.

Experience-led engineering isn't about creating beautiful apps. As Ed Catmull, a Pixar co-founder notes, "Visual polish doesn't matter if you aren't getting the story right. In our world, the "story" is all about how effectively we are enabling users — and often the businesses they are part of — to achieve their goals.



But how can teams live an experience-led life? Here are the key steps:

Get out in the field to understand needs, frustrations, and opportunities

The old practice of relying on SMEs and/or their managers doesn't work anymore. Instead, get out in the field and spend time with the people who will use the application that is going to be built. Try to avoid pulling users into a conference room and conducting a focus group-type session to learn about their needs. Though people can typically describe what's frustrating about their current situation, they often can only identify incremental improvements, not fundamental changes.

"When we designed the mobile and web applications for the network operations users of a large cable client, we spent days riding along in the trucks with line technicians to gain insights that the line techs and their in-office counterparts might never think to articulate on their own." Our UX team is trained to thoughtfully observe situations that users often take for granted or that they don't think can change.

Sometimes it's impossible to spend time directly with customers. In situations like these, we do the next best thing: we spend time in the call center listening in on calls and asking probing questions. When we designed an app for a major travel insurance client, we spent hours in the call center. We learned some important things: travel insurance customers tend towards the older, most people have never submitted a travel claim before, and, given the amount of documentation required to file a claim, it might take a few sessions to gather all the information needed to file a claim. As a result, we crafted an experience using plain, easy-to-understand language, a prescriptive process, and a forgiving interface that enables the customer to put the claim aside and pick it back up where s/he left off. (Side note: a product with poor experience often impacts people who never use the app. For example, complicated products can significantly increase call center volume.)



Write narratives that illustrate the desired experience

Traditional user stories are often clinical and describe requirements instead of the experience. We've had success writing more expansive narrative stories that describe the experience the team is trying to create. Depending on the application, it may take multiple narratives to address various types of users and situations that need to be explored.

Here's an example for a first-time user of an application that supports continuing professional education for CPAs:

- Jim Nazium recently completes his BS in Accounting and passes the CPA exam in his home state of NJ. He joins DC&H and, as an eager young accountant, wants to get a head start on earning the CPE credits he'll need to renew his CPA license in a few years. As part of his onboarding with the firm, Jim receives credentials for the CPE site/app. With some time on his hands, while he awaits his first assignment, Jim decides to explore his CPE options. He downloads the CPE app from the App store and logs in.
- The Home screen welcomes Jim and gives him a few suggestions to get started. He is encouraged to take a moment to complete his profile, which he does. He's able to enter his CPA license and renewal info, areas of interest, work location, etc. The app also suggests a variety of ways that Jim can select interesting courses, including lists of new, popular, trending and highly rated courses, and a custom "New CPA" playlist created by his firm and containing courses the firm believes will get Jim's career off on the right foot. Jim reviews the courses in the New CPA learning path and bookmarks each one so that he can quickly find them again.
- The next morning, Jim decides to take the first course, "Auditing 101," while commuting on the train to his office in New York. He's able to launch the course on his phone, and he's impressed with the readability of the slides. He completes about half the course on his commute and puts the app aside while walking to work. Later that morning, Jim has an open slot on his calendar and decides to finish the course on his laptop. He logs into the CPE site, sees the course on the Home page, and is able to resume the course where he left off. Jim finishes the course in short order.
- During his commute home, Jim opens the CPE app and notices that the app displays a chart indicating that Jim has earned two credits toward his CPE renewal.

These stories are crafted in a specific style: they are written to describe the experience we're trying to create without including specific user interface decisions. We avoid phrases such as "Click the Submit button" and other content that is too literal. Done well, these stories enable everyone — from customer service reps to C-level execs — to debate and get aligned on the goals and enable the designers to create a user interface that matches the spirit of the experience.

Create prototypes that bring the narratives to life – early and often

Narrative stories are great, but they're often not enough to eliminate the "illusion of agreement" issues that we see all too often. When the experience is considered late in the game rather than at the beginning, teams often hear "that's not what I was expecting" feedback the first time people outside the core team get to see and explore the application. The result: teams scramble, and improvements are rushed into development at high cost and stress.

The next step is to create prototypes that bring the narratives to light. Created with tools including Sketch/InVision, Adobe XD, and/or Figma, we can quickly create interactive mockups that help us get the experience right.

Define an MVP that contains the smallest amount of functionality that delivers value

As the team defines the experience, it's possible to determine the timing and sequencing of capabilities into a series of releases. Few great apps are created using a big-bang approach. In fact, the more functionality scoped into a release, the greater the risk that it will be late, over budget, and/or bear little resemblance to what was originally conceived. **"The smallest amount of functionality..." by itself isn't enough; "...that delivers value"** is necessary, too. In other words, keep the scope as tight as possible, but not at the expense of enabling users to succeed. (Pro tip: Given the choice of partially supporting the journeys of a variety of users or more deeply supporting one or two, the latter is often the best strategy.)

Use the experience to guide engineering

Experience-led engineering is not a silver bullet, but the activities described above take the significant risk out of application development programs and help avoid one of the most insidious forms of scope and cost creep: speculative features that never get used. And by aggressively seeking to produce artifacts that clarify needs and design intent, development teams can move forward with a deeper understanding of what to do and mitigate the all-too-common problem of discovering important information late in the game that introduces significant rework, cost, confusion, and stress.

Summary

Designing effective apps has always been challenging. As teams continue to become more and more geographically distributed, they need to modify the processes and tools used to explore, design, define, and engineer effective applications. Conditions like the Covid pandemic accelerate the need for creative new ways to collaborate and mitigate the “illusion of agreement” challenges.



David Katz

Senior Vice President
Virtusa



virtusa

To know more on how we can help,
write to marketing@virtusa.com

About Virtusa

Virtusa Corporation (NASDAQ GS: VRTU) is a global provider of digital business strategy, digital engineering, and information technology (IT) services and solutions that help clients change, disrupt, and unlock new value through innovation engineering. Virtusa serves Global 2000 companies in Banking, Financial Services, Insurance, Healthcare, Communications, Media, Entertainment, Travel, Manufacturing, and Technology industries.

Virtusa helps clients grow their business with innovative products and services that create operational efficiency using digital labor, future proof operational and IT platforms, and rationalize and modernize IT applications infrastructure. This is achieved through a unique approach blending deep contextual expertise, empowered agile teams, and measurably better engineering to create holistic solutions that drive business forward at unparalleled velocity enabled by a culture of cooperative disruption.

Copyright © 2020 Virtusa Corporation. All Rights Reserved.

virtusa.com