Shipping is NOT Success, Let it Sail

by Naresh Jain

All organisations, from the largest enterprise to the newest startup, face the same challenge: how to solve their users' problems by bringing a superior product (or service) to market faster than their competitors while reducing effort spent on overhead activities or, worse, building the wrong product. I hope no one wants to build "stuff" for the sake of being busy.

This is especially critical in organisations that produce consumer products. My case study describes how one such organisation, building products used by millions of diverse users every day, used a #noprojects and #noestimates approach to meet this challenge.

PROBLEMS

In any organisation, understanding your user is always difficult. And this is compounded for consumer products with millions of diverse users, each with their own reasons for using the product. In our case, we had three distinct problems:

1. How do we build a novel product that appeals to our entire user base without making it too complex or expensive (both to build and to maintain)?

2. How do we find the right user needs to focus on and make the right feature decisions (with the best ROI)?

3. How do we innovate rapidly and build a superior product faster than our competitors (improve the time to market and reduce friction)?

While we had a good start with millions of happy users and big investors, we were also exposed to stiff competition from some of the most innovative global tech giants. To beat them, we had to up our game.

CAUSES

The company founder hired a new executive from the Valley. The new hire started to question some of the decisions that had been made: "How many features did we build last year? What was the traction of those features?"

That is not to say that the company wasn't already asking those questions, but a fresh perspective was a strong reminder that they needed to critically examine the product. When they looked at the data, they saw that they had shipped hundreds of features in the previous year, but many of them weren't being used as widely as they had wished. The teams were super productive in shipping but not in generating engagement. They were too eager to celebrate success as soon as they shipped. They produced a lot of waste in the process.

> *"When they looked at the data, they saw that they had shipped hundreds of features in the previous year, but many of them weren't being used"*

There had to be a better way. This was the state when I joined the organisation.
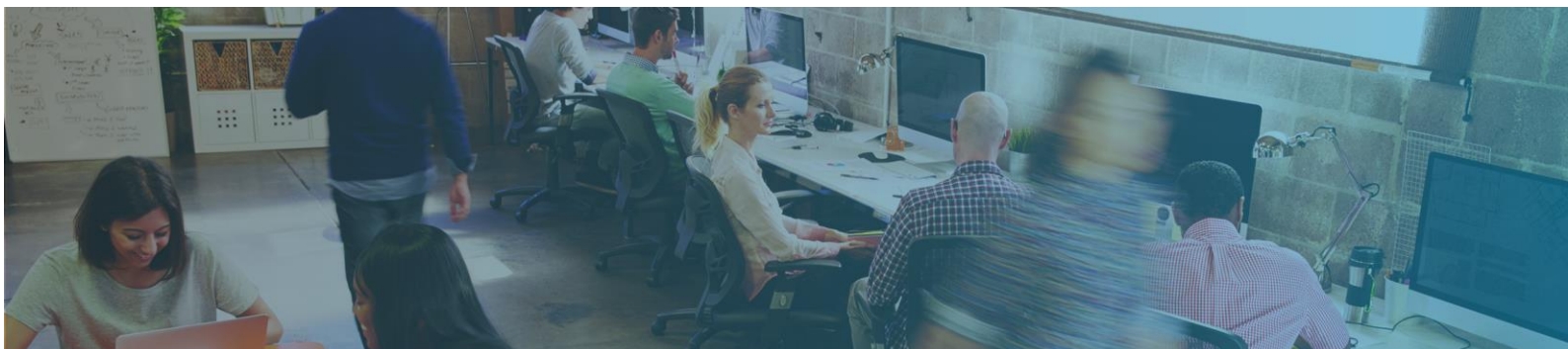
SOLUTION

Our solution had five key elements:

1. We structured teams around user-first product thinking rather than platforms or functions. We called these "value teams" — each responsible for a specific end-to-end user experience (or theme) and consisting of people with the skills necessary to deliver the result. Because user experience is common across platforms, we needed everyone, from mobile to web, to work together. This also meant completely changing how we managed the product roadmap, planned team capacity, and measured effectiveness.

2. We changed our organisational metrics to focus on nine key OKRs, which we gradually reduced to just three, centred on user engagement, user retention, and how deeply the user is invested in our product. Even that wasn't going far enough about five months into the journey, we realized that even three were too many. We gave each value team one primary OKR that they had to actively improve and two secondary OKRs that they had to maintain at least at the same level.

3. We focused on building a learning organisation and building skills and talent (through both acquisition and improvement). We focused on craftsmanship and developing mastery in all disciplines. We also tried to create cross-functional teams by embedding dedicated designers, testers, data analysts, and user-insight people in each value team.

4. We decentralised portfolio management and used a simple data-driven governance model to decide how much to invest in which value team. We killed all discussion of prioritisation, estimation, and resource allocation while shifting the focus to value and impact. Each team could statistically show their impact and decide things appropriately.

5. We moved all teams towards a continuous-discovery and continuous-delivery culture. We started with a problem hypothesis and validating the same with user data. We'd come up with three solution hypotheses and determine the best fit of these by testing them on users. We'd take the winner and run a slice of it in an A/B test with 1% of our users. We used data to decide if we'd refine, scale, pivot, or scrap the idea.

Let me explain how this worked. We had a team that was responsible for the onboarding experience — everything (on every platform) that a new user experiences in the first 24 hours. Another team was responsible for the payments experience. And yet another team was responsible for the third-party-partner experience. We created 12 value teams — although, over a year, we killed six of them and added two new teams as we learned what worked and what did not.

An important consideration was how to measure the impact of each team and how to identify a team that may need more leadership support. This needed us to shift away from output-centric measurements toward measuring outcomes and impact — and to get our teams to understand impact and outcomes. We spent the better part of a month designing what we thought would be a good set of universal measures and defining what good outcomes mean.

It was important to build the right culture. And because we believe that culture emerges from structure, it is important that these teams truly owned the end-to-end experience and could go deep into it to connect with the user experience and the user psychology.

When I describe it like this, it might sound like we started with a clear idea about how everything would work. In reality, it was a lot messier. It took a year of experimentation and adaptation to come up with this solution.

IMPLEMENTATION

The first thing that we did was to challenge how product decisions were being made. We moved from gut feelings or pseudo data-driven (biased) decision making to using statistically significant user data to make informed decisions. We made a significant investment in democratising data and running experiments at scale, which could help everyone in the company to gain insight from data and make more informed decisions. Anyone could challenge the product manager if the data did not support the hypothesis.

The other big challenge was the number of long-term bets that the company had been focusing on. In order to keep the board and the investors happy, we had to have grand things to talk about. Many times, these big, new, shiny things would take over. Breaking those big ideas into tangible, quantifiable experiments was really important. Putting a WIP limit

on these long-term bets was equally important as we were seeing a significant impact from refining and polishing what we already had. The balance is extremely critical.

Once we started down this path, two things emerged:

1. Our annual plan focused on fewer big bets and featured a lot of emergent short-term bets (refinements). This meant that the teams could be lot more responsive to users, as they learned about user behaviour and measured usage patterns without having to worry about the big bets.

2. This also meant that we could move out of marketing-driven-development mode — i.e., marketing was no longer calling the shots on the timeline. This led to teams feeling less pressure to deliver. They could really experiment and iterate ideas in a safe-to-fail environment. We would prove with data that a hypothesis worked on a statistically significant chunk of the user base before we rolled it out to the entire user base. Marketing also become an integral part of our day-to-day activities, giving teams more confidence in this approach. Now, our teams were able to focus on the core user need and give users a "wow" experience.

This directly led to thinking that we could structure work around self-contained themes, thus allowing teams to own the full end-user experience. We called these "value teams".

Unlike earlier project teams, these value teams were permanent. We brought together cross-functional skills and encouraged the teams to deeply examine user psychology. We needed them to understand user needs so that they could design a product that would glue our users to their screens for 12 hours a day.

This was a fundamental shift in how the organisation operated — shifting from project-based thinking, with all its associated overhead and short-term perspective, to product-based thinking from a user's point of view.

Even the makeup of the teams changed. We had:

- a product manager,
- a data analyst,
- a designer,
- a tech lead,
- iOS developers,

- Android developers,
- Windows developers,
- back-end microservices developers,
- a user-insight (CI) person (folks that reach out to users for feedback and user intelligence), and
- theme testers (integration testers would essentially look at the entire product across all the teams).

You'll notice that we didn't have an ops person in the team - we taught the developers to take on this responsibility.

CHALLENGES

There was a lot of experimentation and volatility throughout our journey. We could categorise what we learned as team design, politics, and measurement.

We spent a lot of time experimenting with team design. We started by creating 12 teams, which we then dropped to eight, then to six. We ended up going back to eight. This volatility was natural as we were clearly measuring the impact each team/theme had on the broader business outcomes. Themes that didn't have a material impact on the OKRs we either dismantled or restructured and reassigned those people so there was no fear of job loss, which allowed the teams to be honest and not game the metrics.

We had the usual challenge of managers who felt that they were losing control. Historically, they'd build up political capital based on team size, and we took that into account when forming (and re-forming) teams.

Our biggest challenge was measurements and metrics. Historically, we tended to try to micro-measure things. We generally thought that the more we measure something and the more precisely we measure it, the better we got at understanding it. But the opposite was true. At the time, we were measuring NPS and several other misleading KPIs, but all we were really doing was feeding our own confirmation biases — the more data we had, the more we could make that data say what we wanted it to.

> *"We spent a lot of time experimenting with team design. We started by creating 12 teams, which we then dropped to eight, then to six. We ended up going back to eight."*

Once we decided that increasing engagement and retention was our target OKR (and I'd go so far as to say that that target is relevant for every company), we needed to decide how to measure it. We commissioned our data-science team to look at groups of highly engaged users to find their "aha!" moment and distil that into something we could measure. We discovered some interesting correlations between various behaviours and usage patterns that allowed us to refine our impact measurements.

We ended up using this information proactively to improve user engagement and retention. It also helped everyone in the company focus on the same impact and talk the same language. These are important ingredients for creating a user-first thinking culture.

OUTCOME

We achieved some pretty fantastic outcomes. User retention rose by over 20% and engagement rose by over 30%. While it took us longer than before to release features, when we did release, we had much better conversions. Without spending a single penny on marketing campaigns, we had a steady flow of new users coming in. And, most importantly, the culture changed as well. We are now more likely to experiment and kill new features than ship and fail.

> *"User retention rose by over 20% and engagement rose by over 30%"*

The journey continues. While I'm not involved with this company anymore, they continue to evolve their business structures and models to meet the changing demands of their users.

- Naresh Jain

Developer... consultant... conference producer... startup founder… struggling to stay up to date with technology innovation. Null-process evangelist Naresh Jain is an internationally recognized technology and product-development expert and the founder of ConfEngine. Over the last decade, he has helped streamline the product-development culture at many Fortune 500 companies like Google, Amazon.com, HP, Siemens Medical, GE Energy, Schlumberger, EMC, and CA Technologies. His hands-on approach to product innovation by focusing on product discovery and engineering excellence is a key differentiator.

Naresh founded the Agile Software Community of India and organises the Agile India conference. He is also responsible for organising 50+ international conferences including Functional Conf, Simple Design and Testing Conference, Agile Coach Camp, Selenium Conference India, Open Web & jQuery Conference, Open Data Science Conference India, and Eclipse Summit India. He has started many agile user groups including the Agile Philly User Group and groups in India. In recognition of his accomplishments, the Agile Alliance in 2007 awarded Naresh with the Gordon Pask Award for contributions to the agile community.

Naresh Jain, Developer, Consultant, Conference Producer, Startup Founder