1 Identify Key Business Outcomes

Modern product development is witnessing a drastic shift. Disruptive ideas and ambiguous businesses conditions have changed the way that products are developed. Product development is no longer guided by existing processes or predefined frameworks. Delivering on time is a baseline metric, as is software quality. Today, businesses are competing to innovate. They are willing to invest in groundbreaking products with cutting-edge technology. Cost is no longer the constraint—execution is. Can product managers then continue to rely upon processes and practices aimed at traditional ways of product building? How do we ensure that software product builders look at the bigger picture and do not tie themselves to engineering practices and technology viability alone? Understanding business and customer context is essential for creating valuable products.

This chapter addresses the following topics:

- Defining our business model and unique value proposition
- Deriving inputs for product development
- Understanding key business outcomes
- Defining our impact-driven product

We are driven by purpose

Honeycombs are an engineering marvel. The hexagonal shape of honeycombs is optimized to reduce the amount of wax needed to construct the hive, while maximizing the storage capacity. However, building wonderfully-crafted honeycombs isn't the raison d'être of the bees. The goal of their existence is to maximize their chances of survival, to keep their lineage alive. Every bee activity is centered around that.

The need to maximize chances of survival is true for nearly every living species, but that does not mean that the queen of the ants should ask her ant army to construct ant hills with wax in hexagonal tubes. It doesn't work that way. Every species has an integral

DNA that defines what it eats, how it socializes, how it defends or attacks, how it adapts and how it survives.

The engineering that went into the honeycomb contributes to a specific way of life that is suited to the bees. It is important to realize that the very first bees didn't initially build wax hexagonal tubes. They iterated over many generations and responded to feedback from nature (which is often harsh and can result in extinction). Bees have pivoted to this model and it seems to have worked rather well for them. Heck, they've even managed to find channel partners in humans!

In the product world, understanding outcomes and user goals is essential to building successful products. A product's success always relates to the end goals it meets. Engineering in isolation adds no business value. The same can be said about marketing or sales. Business functions acting in isolation, without the context of business outcomes or creating value to the customer, are not sustainable. The sooner we define end goals, the better for the business as a whole. After all, we don't want to end up building honeycombs for ants.

Business context defines everything we do

"Universal law is for lackeys. Context is for kings."

– *Captain Lorca, Star Trek: Discovery*.

When the Agile manifesto was created in 2001, software practitioners were calling for a change in the way that software was delivered. The waterfall model of software development, which was predominant, required that software specifications and business requirements were frozen upfront. There was very little room for change. Change had to be avoided because changing anything midway would increase costs and timelines. Waterfall had evolved as a way to handle software development processes based on the limitations of the software technology of that time. Much of software development then was about automating manual processes or building bespoke technology solutions. Technology has since evolved a lot. It has become increasingly flexible and inexpensive to build technology solutions with direct consumer impact. The focus for software builders has expanded to include customer satisfaction and is no longer about just meeting budgets and timelines.

There are many frameworks (Scrum, XP, and so on.) that offer a methodology for software teams to focus on user experience and customer value. They foster team collaboration and enable teams to respond to change. The guiding principles behind these frameworks are noble, but failures always happen in how these principles are put into practice.

Without context, intent, and outcomes being defined, even well-meant advice can go to waste. This is because we're humans and not honeybees. We are driven by purpose, we are lazy and we are imaginative. People can conjure up images of success (or failure) where there is none. They plan, speculate and adapt. They try to optimize for imagined future situations. We know that a sense of purpose is crucial to rallying a group of diverse, independent and creative people to come together and effect change (or even build great products). The problem is that's exactly where we seem to fail time and again.

We fail because we focus on the wrong stuff. We try to predictably measure productivity in a creative profession. We over-engineer solutions to non-existent use cases. Software engineering is an interesting field of work. It's like an artist and a mathematician coming together to generate business value! Yet, we treat software engineering like automobile manufacturing. I know amazing software engineers who can churn out beautiful and working software, but their pace of delivery is not predictable. Every person has their own pace, and style of working, which is a problem for project managers. When we have strict timelines and budgets to meet, how do we commit if there is so much subjectivity? Software teams have spent a lot of time inventing processes to accurately predict engineering efforts. Estimations, planning activities, functional scoping, and progress reporting have gained a lot of visibility in this context. We focus too much on engineering practices.

Software product development is missing out on a great opportunity here. By carrying over software delivery frameworks of meeting timelines, budgets and customer satisfaction, we are in a way restricting ourselves to looking at outputs instead of outcomes. It is time that we changed our perspective on this. Product management shouldn't be about measuring effort or productivity. Product management shouldn't be about measuring output. Instead we should focus on product outcomes. What value are we creating for the customer? What outcomes is the product helping the business to meet? How can we execute a plan to achieve this in the smartest way possible? So how do we measure these outcomes?

What influences product success?

In the good old days of business process automation (until late 1990s), measuring success seemed much simpler. Software teams could say, "There is an existing way of doing things. Let's automate that, but stick to budgets, timelines and quality metrics." Goals were simple and measurable. Software design was never discussed. User experience was non-existent. Business impact was purely profitability based and no behavior changes were hinted at. Even as we moved into the era of Agile, business folks would give us weird looks when we asked to speak to end users. A client once told me, rather dismissively, "We're building this (app) for accountants. They don't have an eye for color. I don't know why you're worried about user interface or design." He may have made total sense a decade ago.

Today, you'll easily hear a CEO say things like, "We need to create stickiness, and to ingrain in our users a desire to engage with our apps." Many a product manager will roll their eyes and think, "Whatever that means." Success for a product is no longer about meeting budgets or timelines. The narrative of software development has changed. Can product management afford to lag behind?

Understanding business goals, and evaluating risks and costs, has always been essential for driving trade-off decisions. Business goals today are multi-dimensional and so are risks. How we make trade-off decisions depends on what aspects of product success we want to consider. Today, a product manager has to handle many facets of goals and risks, before defining the right product to build and driving trade-off decisions. According to the American consulting firm McKinsey & Company, "Product managers are the glue that bind the many functions that touch a product — engineering, design, customer success, sales, marketing, operations, finance, legal, and more. They not only own the decisions about what gets built but also influence every aspect of how it gets built and launched."

Traditional *project planning* isn't going to help anymore. Product managers are in the driver's seat. We have to steer the product to meet ambiguous business goals. When building disruptive products in ambiguous business conditions, there is no "existing way" to refer to. All we have is a bunch of glorious ideas. So, the first step to demystifying goals is to articulate intent.

Impact-driven product development needs to consider four key pieces of input that influence and guide product success:

1. Business outcomes that a product can deliver. Why is this product important for the business?
2. Value that a product can create for a customer. What should we build in order to deliver the maximum value to users?
3. Execution plan. When there are many ways to implement the same functionality, how do we decide upon the best way forward?
4. Internal business constraints and external market influences.

A business as a whole has its own value creation goals outlined. A product may address some or all of a business' goals.

Business model

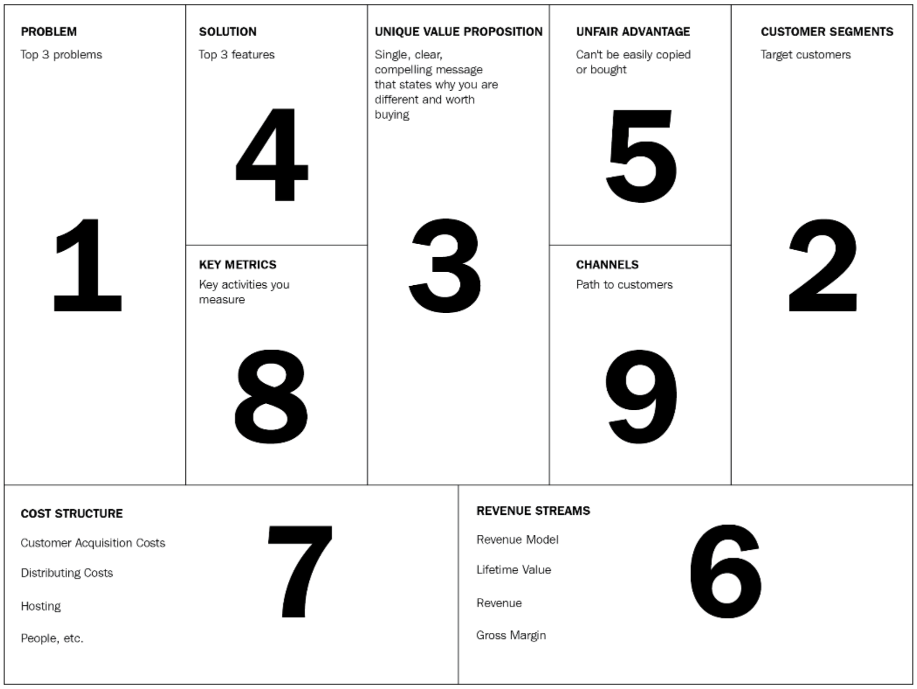
A business model is a starting point for identifying a product idea and drawing up a plan. The desired outcome of this plan is to identify the unique value proposition that our business intends to deliver to the users/customers. In order to arrive at this, we need to define who our users are, what problems we're trying to solve, what channels we can leverage, who our competition is, how we plan to make our revenues, and what costs we might incur.

For startups, or businesses in the early stages of development, the Lean Canvas, or any other framework that can help to visualize the market, business ecosystem, unique value proposition and target segments, is a great place to start. Even for a product manager joining a team midway through product development, understanding the vision, market and target groups and so on is key. After all, context is king. If the business model hasn't yet been laid out, then I recommend that you do that first.

The Lean Canvas is a fantastic tool for capturing the business model around a product. Product Vision Board is also another tool to help us to visualize product vision. Defining the business model around the product is a great way to define the target customers, the problems we're trying to solve for them, and the unique value proposition that we can offer. This can form the basis of how we want to run our business. Not every aspect of our business needs to be productized. Also, not every user segment may value every product feature with equal merit.

The business model can help us to visualize these target segments, and capture which segments we want to go after in the current phase of business. However, the Lean Canvas stops at the business model level. When investing in technology, we need to be clear about the outcomes we seek and the trade-offs we need to make. Building a product requires that we understand the larger business context, but we iterate on a product strategy that helps us to deliver a solution that meets business outcomes and provides value to customers. Software technology is extremely flexible. We can build quick and dirty solutions, or we can build robust, long-lasting solutions. However, every product must be based on business and customer motivations, and not based on technology preferences.

Below is the Lean Canvas model proposed by *Ash Maurya*:



PROBLEM
Top 3 problems
1

SOLUTION
Top 3 features
4

KEY METRICS
Key activities you measure
8

UNIQUE VALUE PROPOSITION
Single, clear, compelling message that states why you are different and worth buying
3

UNFAIR ADVANTAGE
Can't be easily copied or bought
5

CHANNELS
Path to customers
9

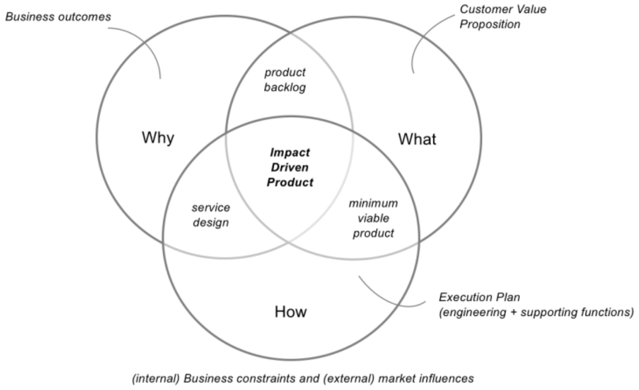
CUSTOMER SEGMENTS
Target customers
2

COST STRUCTURE
Customer Acquisition Costs
Distributing Costs
Hosting
People, etc.
7

REVENUE STREAMS
Revenue Model
Lifetime Value
Revenue
Gross Margin
6

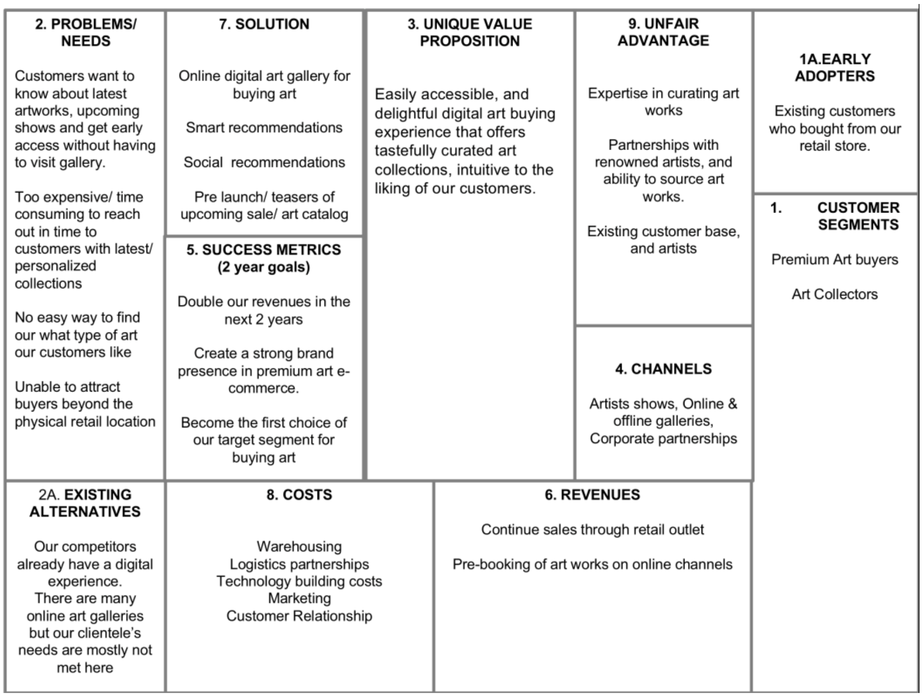
The overall business plan may also vary depending on the stage of the business. Business context and constraints have a bearing on what is achievable. For instance, at early stages, a business may be interested in creating value for early adopters. Some businesses may not attach importance to revenue generation until there is sufficient traction. As a business scales, other goals may become important. Value creation becomes more comprehensive.

This relationship between business outcomes, a customer value proposition and an execution plan can be represented as below:


Business outcomes
Customer Value Proposition
product backlog
Why
What
Impact Driven Product
service design
minimum viable product
How
Execution Plan (engineering + supporting functions)
(internal) Business constraints and (external) market influences

Case study

Throughout this book, we will use a sample business case to illustrate concepts. We will use an imaginary business called ArtGalore, which is an art gallery specializing in curating premium works of art. The company now wants to venture into creating a digital art viewing and purchasing experience for premium art buyers. Let's look at the Lean Canvas created for ArtGalore (this also refers to the case study chapter). Below is a sample representation:



2. PROBLEMS/ NEEDS

Customers want to know about latest artworks, upcoming shows and get early access without having to visit gallery.

Too expensive/ time consuming to reach out in time to customers with latest/ personalized collections

No easy way to find our what type of art our customers like

Unable to attract buyers beyond the physical retail location

7. SOLUTION

Online digital art gallery for buying art

Smart recommendations

Social recommendations

Pre launch/ teasers of upcoming sale/ art catalog

5. SUCCESS METRICS (2 year goals)

Double our revenues in the next 2 years

Create a strong brand presence in premium art e-commerce.

Become the first choice of our target segment for buying art

3. UNIQUE VALUE PROPOSITION

Easily accessible, and delightful digital art buying experience that offers tastefully curated art collections, intuitive to the liking of our customers.

9. UNFAIR ADVANTAGE

Expertise in curating art works

Partnerships with renowned artists, and ability to source art works.

Existing customer base, and artists

4. CHANNELS

Artists shows, Online & offline galleries, Corporate partnerships

1A.EARLY ADOPTERS

Existing customers who bought from our retail store.

1. CUSTOMER SEGMENTS

Premium Art buyers

Art Collectors

2A. **EXISTING ALTERNATIVES**

Our competitors already have a digital experience.
There are many online art galleries but our clientele's needs are mostly not met here

8. COSTS

Warehousing
Logistics partnerships
Technology building costs
Marketing
Customer Relationship

6. REVENUES

Continue sales through retail outlet

Pre-booking of art works on online channels

In the above Lean Canvas, the unique value proposition describes the concept of a digital art buying experience, the success metrics define the business goals, and the solution describes how the business intends to deliver the value proposition to its clientele. At the stage of problem/solution fit, the actual execution plan of how this digital experience will be created is not yet finalized. Even the value proposition, target segment and solution are yet to be validated. The Lean Canvas represents the big hypothesis that the business is making about its growth in the next two years. The Lean Canvas, in a way, also represents a slightly longer-term view for the business. Along the way, as the business discovers more about the customer's perception of value and actual growth indicators, the canvas could change and ArtGalore could pivot to a different business model if the hypotheses are proven wrong.

From a product development perspective, we need to identify the three aspects of product success discussed earlier in this chapter:

Defining business outcomes that the product can deliver (why?):

What does the business hope to achieve by investing in a digital art gallery? The success metrics shown in the Lean Canvas are a broad indicator of long-term goals. Building a brand and doubling revenue are long-term goals. In order to work towards those goals, the business needs to make some product investments in the short term.

Defining the value proposition that the product will create for the customer (what?):

The unique value proposition, and the solution concept, coupled with the customer segments, as described in the Lean Canvas, addresses this. It is natural for us to jump into ideas about product features, but before we even arrive at features, we need to think about user personas in our customer segment and define how the solution will help to deliver the unique value proposition to them. Eventually the features get refined into product functionality.

Defining the execution plan (how?):

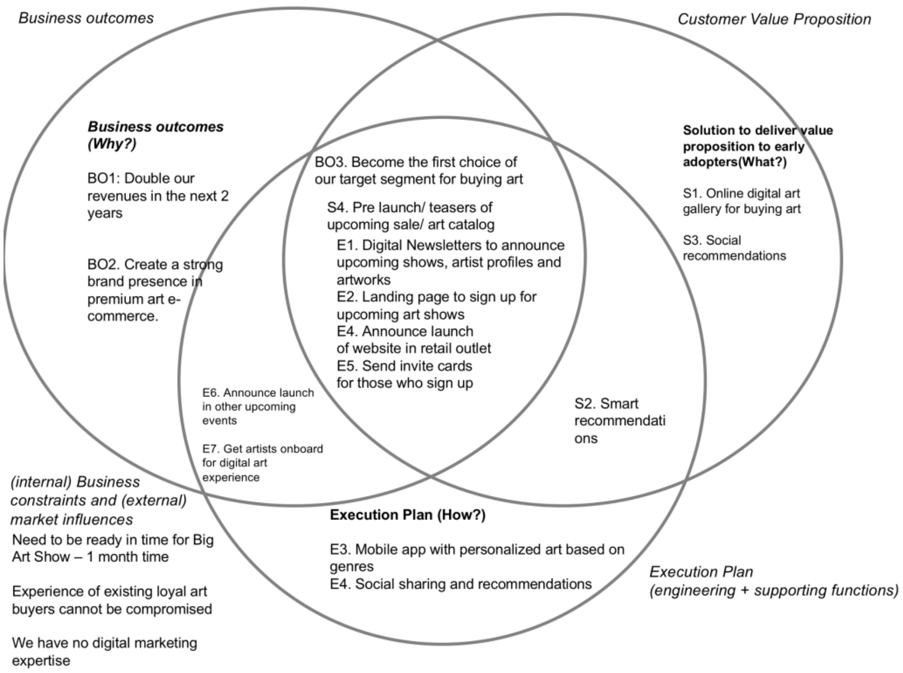
There can be many ways to deliver the value proposition to the customer and to meet business outcomes. The execution plan for delivering the value proposition and business outcomes can depend on ease of execution, cost of execution and the potential to create a positive impact on business outcomes. While the preferred solution for ArtGalore is a digital art gallery, not all aspects may need to be built in upfront. The execution plan must factor in the strengths of all the business functions that exist in ArtGalore. The focus must not be just about how technology can solve all customer problems/meet business outcomes. The focus must be about how to work seamlessly with business functions and ensure that the end-to-end product experience is satisfying.

Business constraints and market influence:

This includes factors such as time available to go to market and the potential to raise investments. Governing policies, regulations and compliances (if any), technology potential, capabilities, skills available and so on. can also influence business goals and the product experience. These factors will determine the key goals to focus on, the product features to offer, and the best way to implement them. In addition to constraints, the intrinsic business intent is important. What are our uncompromising principles? What do we value more? What values drive our decisions? What is our appetite for risk? These factors will decide how we deliver value.

A product can impact the business or the organization that is creating the product. It can also impact the customers who will adopt the product. An impact-driven product therefore is one that finds the best way to deliver value to customers and meet business outcomes under internal and external constraints.

Based on the unique value proposition, and business context and business outcomes prioritized for a specific time-bound milestone, following is a sample illustration of the inputs for product development for ArtGalore.


Business outcomes
Customer Value Proposition
Business outcomes (Why?)
BO1: Double our revenues in the next 2 years
BO2. Create a strong brand presence in premium art e-commerce.
BO3. Become the first choice of our target segment for buying art
S4. Pre launch/ teasers of upcoming sale/ art catalog
E1. Digital Newsletters to announce upcoming shows, artist profiles and artworks
E2. Landing page to sign up for upcoming art shows
E4. Announce launch of website in retail outlet
E5. Send invite cards for those who sign up
Solution to deliver value proposition to early adopters(What?)
S1. Online digital art gallery for buying art
S3. Social recommendations
E6. Announce launch in other upcoming events
E7. Get artists onboard for digital art experience
S2. Smart recommendati ons
(internal) Business constraints and (external) market influences
Need to be ready in time for Big Art Show – 1 month time
Experience of existing loyal art buyers cannot be compromised
We have no digital marketing expertise
Execution Plan (How?)
E3. Mobile app with personalized art based on genres
E4. Social sharing and recommendations
Execution Plan (engineering + supporting functions)

This is quite a simplified representation of the product development inputs at a given point in time. The diagram illustrates how business constraints may influence what the impact-driven product may deliver to the user.

The above illustration indicates that the most important outcome for ArtGalore is to become the first choice for art amongst its target segment. The most important aspect of the value proposition is offering tastefully-curated art collections for existing buyers who don't have this information today. The most effective plan to execute this solution is to offer digital newsletters, provide an online sign-up for upcoming art shows, follow up on the sign-up with invites and announce the website. This has been identified based on the business constraints of being ready for the upcoming art show, and not compromising on the existing buyer experience and so on. The other solutions, value propositions and execution plans are also great, but given the business constraints, the impact-driven product is where we want to steer all our resources to.

Minimum viable product versus Impact driven product

Minimum viable product (**mvp**) is a concept that is quite prevalent in the software development scene. There are different interpretations of the concept of an mvp and *Chapter 6*, Chapter name dives into the interpretations of minimum viability.

At a basic level, an mvp intends to test the riskiest proposition of a business idea or a technology, without running out of resources. This is very helpful when testing an unproven technology or when building under tight business constraints. However, an mvp falls short in terms of delivering an end-to-end product experience to the consumers. The problem with how we define an mvp is that we tend to view technology viability as a value driver operating in isolation. An mvp is great in the problem/solution fit stage, but in the product/market fit stage, there are many other aspects to consider. When launching a new product, there are umpteen activities that need to be orchestrated to ensure success. Marketing, sales, support and other operations need to come together to ensure a product's success. Testing viability of a product in isolation isn't going to ensure success for the business.

In my startup, we had built up a platform for businesses to design and build mobile apps for client engagement. We had started out in the events and conferences domain, where app creation lead times were pretty low, audience engagement was critical, and last-minute changes to a schedule were quite common. Our platform was geared towards addressing those needs, but we soon started seeing interest from other business verticals, who saw the potential that our platform had in driving customer engagement. One such business asked us to enhance our platform with bespoke functionality.

We were in early stages in our startup and naturally we got quite excited about the opportunity. The possibility of opening up our platform to another domain was quite lucrative. While we didn't see much value in creating a custom solution for a single client, we realized that we could enhance our platform to cater to the entire business vertical. This would also give us a great opportunity to launch our product into the B2C space. So, we carved out an mvp for this, and our engineers got busy with re-architecting the platform to make it flexible across business verticals. After spending more than a month on this, we were more or less ready with the revamped software.

Our revamped product was technologically viable. Yet was the revamped software impact-driven? No. Why? We hadn't looked at the end-to-end value of the product. Our business model was different. Our target audience was new. A B2C market needed a very different approach. We hadn't planned for marketing, content creation and promotions. We realized we had neither the right skills nor the investment to drive this product to success. We hadn't really defined to ourselves what the new product would deliver. After all, we had just one customer who was interested. We hadn't looked at an end-to-end product experience.

Our rookie mistake cost us a month. We ended up shelving the product and didn't even close our sale with the one customer who showed interest. From the software perspective, we had it all covered, but we had not given any thought to the whole product experience or to the outcomes it could generate for our business. We are much better off today having learnt and recovered from that mistake quickly, instead of having gone down that path purely riding on the viability of our platform. We had looked at one customer's request and created a hypothesis about the problems our new platform could solve and the needs it could satisfy for that customer segment. We were confident that the product would address their needs and it could have. However, we hadn't given any thought about what it meant for our business, the stage we were in, our internal capabilities, and our weaknesses.

I know many developers who take great pride in how their software is built. The prevalent thinking is that if you build your software right, it will take care of itself. I've seen very smart engineers express deep disregard for supporting functions. They look down upon marketing, promotions and sales saying, "People will adopt our product because my software is great." However, success isn't in and of the technology process itself. This is a core difference between task-oriented thinking and goal-oriented thinking.

Businesses make the same rookie mistake when going after early adopters. We want to get our business model validated quickly and cost-effectively. So, we build an mvp with a lot of loose ends on experience, revenue generation, and operations. Then we identify a safe pilot group from our family, friends and well-wishers, or we offer our product for free to anyone who agrees to try it. Early adopters need to be potential (paying) customers who will risk adopting a product with partial experience, because it solves their most pressing needs. They will risk an incomplete product as long as their needs are met.

However, if we don't choose our early adopters right, we may end up validating the wrong product, which then tanks in the product/market fit stage. For instance, when we offer our product for free, how do we know if the customers are adopting our product because they see real value in it or because it's free? We feel happy with the validation we get for our software, but as we move out of that pilot group, we realize that our product doesn't have many takers. This is because we may have chosen our pilot group safely and they told us what we wanted to hear. The real test for a product is in testing out the riskiest propositions. The riskiest propositions are not just about how our product can create customer value, or about the technology viability in delivering that value, but also about the business capabilities.

If we're building a business, then we need to have a plan for how to make it grow, create impact and be sustainable. Good software is never a replacement for a sound business model, and vice versa. A product entering the product/market fit stage should have validated the viability of the riskiest business/technology propositions. Product/market fit should focus on offering an end-to-end product experience that offers the *most* value to the consumer and meets the business outcomes. The learnings that we have had right from product ideation and beyond have to continuously feed into the product experience. We need to think about a holistic product experience, rather than feature set or technology.

If we don't realize that a product experience is a result of many technical and operational parts coming together, to meet a larger goal, then we're just missing the point.

After all, we're not just laying the bricks; we're building a cathedral, a hospital, or a school.

An impact-driven product is the one that combines the technical and operational aspects of a product to create a holistic product experience. It comprises the entire product experience, across all touchpoints, for the customer. The impact-driven product ensures that we carve out a product knowing full well what business outcomes to target. We factor in the support and co-ordination we need from business functions. We identify the smartest way to deliver value to customers under given business constraints. Knowing key business outcomes is also a critical input to product development, and this should not be a one-time activity.

Key business outcomes

In early 2000, I worked as a business analyst. We were following the waterfall methodology, and I used to maintain huge 500-page requirements documents. Waterfall methodology assumes that software development happens in stages. The first phase is to understand business requirements. It is followed by designing and architecting the solution. Solution development starts only after architecture is signed off. Testing begins only after development is complete and so on.

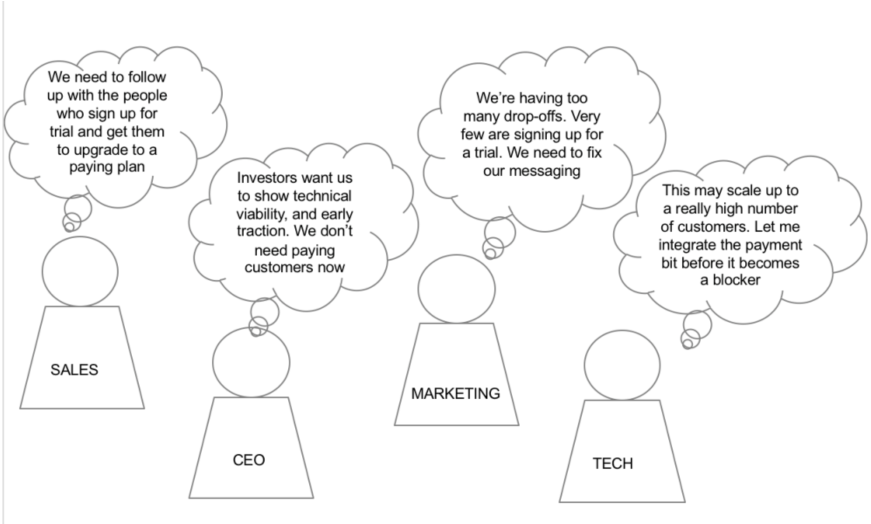
Interactions with business stakeholders happened primarily at the requirements gathering phase. When building a large solution, that phase could run into months. There wasn't much interaction with businesses, or customers, after this phase. In my case, most of my interactions used to be confined to the analysis team. We used to be isolated from developers and testers. I mean, we actually sat on different floors, and in one instance testers were in a different building altogether!

For the entire solution to be delivered, it could take many months, or even years. By then, a lot could have changed on the business scene. My first experience of understanding the impact of this came by chance. I was asked to fill in for my manager on a year-end prioritization call. I had to read out a one-line status update about a project we had been working on for six months at that point. I had no inkling about what the call was about. As a rather naïve junior analyst, I had no idea that all the regional business heads were going to be on the call. Towards the end of the two-hour long call, the facilitator read out a list of projects that were ongoing and in the pipeline for the next year. Every regional head made a go/no go decision, indicating whether they were willing to sponsor that project or not. To my shock, an application that one of the teams had been building for two years got rejected. Poof! Just like that, two years of software development turned to ash. I was intrigued. How could this be? Why would they waste so much money building something and then scrap it on a whim?! Was this only a problem with waterfall? How could we have anticipated business priorities?

I have witnessed many Agile projects shutting down. Not because we weren't building things right, but because the business context had changed. In some cases, the business ran out of money. Product teams were not closely aligned with business and while the business context had been shifting for a while, the previously identified backlog was still being built. In other cases, the product wasn't rolled out to users until much later, and adoption went downhill. Agility helps in responding to changing business context, but what if we could predict the trends of where the business is heading? The problem also occurs when business functions don't collaborate and work towards the same business outcomes. Even when the situation isn't calling for something as drastic as a shutdown, product teams are often taken by surprise about how a feature is perceived by a business. Business priorities need to align with every aspect of product building.

I know many a software engineer who is confounded by the idea of a quick-and-dirty working code, even when the code meets customer needs and business outcomes. I have seen business stakeholders fuss over the exact hue and shade of the brand colors in the user interface, and not really care so much about how performant the system is. Sometimes, this is due to an incorrect frame of reference. Some stakeholders don't really understand the effort that goes into optimizing a page load for an image-heavy website that is used by a million users. So, they may just brush it off and say, "Of course, that's to be expected. Why would it be otherwise?" This is because we make some assumptions about product outcomes. We treat a business outcome as a one-time requirement that is frozen for the lifetime of a product. We then iterate only on a product plan to deliver product features. We make assumptions about which customer value proposition is important. The business (as a whole) should determine trade-off decisions, the value proposition and what customer feedback to act upon. Understanding business outcomes should also be iterative.

What value proposition do we need to deliver in our product? What business outcomes are we trying to meet? Isn't it wasteful for developers to build a highly performant system, when product differentiation is what we're going after? Isn't it just as wasteful for marketing teams to strategize about reach and engagement, when what we're really looking for are a few early adopters? These questions require collaboration and collective decision-making.


We need to follow up with the people who sign up for trial and get them to upgrade to a paying plan
SALES
Investors want us to show technical viability, and early traction. We don't need paying customers now
CEO
We're having too many drop-offs. Very few are signing up for a trial. We need to fix our messaging
MARKETING
This may scale up to a really high number of customers. Let me integrate the payment bit before it becomes a blocker
TECH

As shown in the illustration above, each of us holds a certain view of the business potential. Each of us is right in our own way. Brand messaging, sales follow-up, integrations to ease operational activities, and planning for scale are all good, but is now the right time to address them? What determines that?

Where we direct our efforts depends on the present context of the business. This should not be a choice that business functions make in isolation. Business functions (product, marketing, sales operations and so on.) should collaboratively make this decision. Why we choose one way to execute over another, or what trade-offs we make, depends on the intrinsic organizational makeup. The combination of context and this choice determines business intent.

At every stage in a product's development (and especially at the start) we need to identify what drives the business intent. How do we plan on delivering the unique value proposition of our product to the customers/market? What would tell us that we're doing the right thing for the business? What aspect of our value delivery is most important for us? What trade-offs are we making?

Growth, sustainability and impact are pretty much the goals for every business (non-profits and socio-political organizations may align better with impact than profitability). Yet if the business is not making money sustainably, or influencing sustainable change, can it exist? Should it exist? With this business goal, we can come up with a list of outcomes which are pertinent to the present business context. These outcomes can help us to align our product value propositions. They are the key business outcomes we want to track. Each business outcome would have its own success metrics, which we can measure.

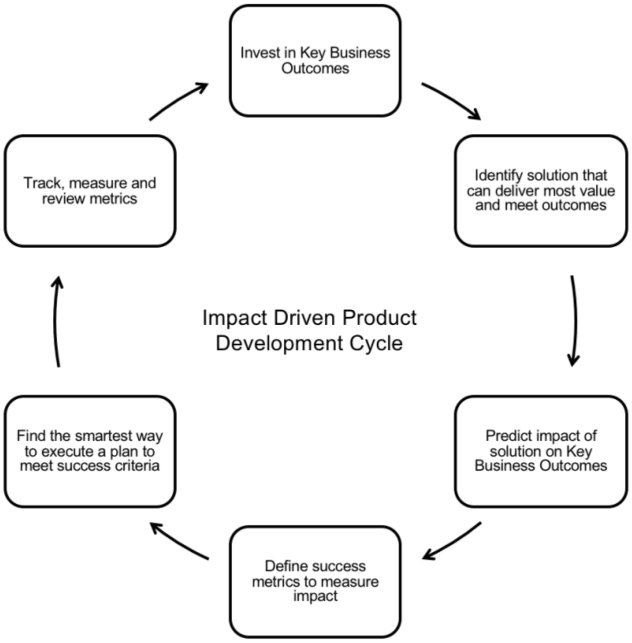
Some of these key business outcomes are listed below. This list is indicative and generic. We may be able to identify other outcomes based on the context of our business. Typically, aim for not more than one to three outcomes:

- Growth
    - Acquisitions (new registrations)
    - Marketing (branding, reach, visibility, virality)
    - Scale (gearing up exponential growth)
- Sustainability
    - Revenues (sales)
    - Investments (fundraising)
    - Costs (reduction/control)
    - Streamlined operations (improved SLAs)
- Influence
    - Retention (churn)
    - Engagement (active users, support, content)
    - Adoption (training, access, user experience)

As you may note, some of these outcomes are tangible and easily measurable, for instance acquisitions. Others are quite abstract, such as customer satisfaction or marketing reach. In the next few chapters, we will try to define the criteria for measuring every business outcome.

Every business is unique and goes through different stages of maturity. Therefore, some business outcomes are likely to be more important than others, depending on which stage of business we are at, and what type of organization we are. Also, it's nearly impossible to go after everything listed above. We may spread ourselves too thinly trying to achieve success in every aspect. We will need to make trade-offs, and those trade-offs need to translate into the product.

The unique value proposition from our business model, and the key business outcomes, form the basis for defining the impact-driven product. The representation below indicates the various steps in the impact-driven product development cycle.


Invest in Key Business Outcomes
Identify solution that can deliver most value and meet outcomes
Predict impact of solution on Key Business Outcomes
Define success metrics to measure impact
Find the smartest way to execute a plan to meet success criteria
Track, measure and review metrics
Impact Driven Product Development Cycle

We will review each step in the impact-driven product cycle in the following chapters.

Summary

Practice trumps theory. While software methodologies and frameworks can guide us, our interpretation and adoption of these frameworks in our unique context matters a lot. The focus needs to remain on measuring outcomes and not output. Product development needs to factor in business outcomes, customer value, an execution plan, and internal and external constraints to ensure that a holistic end-to-end product experience can be created. Identifying the key business outcomes is an important input to product development, and this should not be a one-time activity. Being cognizant of the business constraints, and staying aligned on key business outcomes, can ensure that we don't lose focus of the metrics that will make our business succeed.

EXPERT INSIGHT
Mangalam Nandakumar
Lean Product Management
Successful products from fuzzy business ideas
Packt>