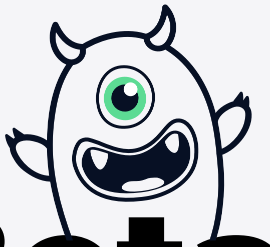
stomio

2023



Beta Testing Handbook

Our guide to launching new products
your customers will love

www.stomio.io

stomio

Welcome to the beta testing handbook

Many product professionals find themselves in shock while beta testing their products. It’s very much a process on its own, with multiple stages that each have their own requirements and challenges. **Let’s break it down.**

stomio

Table of Contents

1.

Introduction

Introduction

"Where are the beta units? At the warehouse? I can't wait to see them!"

I was feeling like a kid in a candy store. It had been almost a year since we started working on this product, and today was the day we got to see the full experience: our beta units had arrived. I rushed to the warehouse to open the first carton and almost teared up looking at the 12 products neatly packaged inside. I thoroughly examined each detail we designed in the development stages, eager to get these products into the hands of customers.

Six months later, I was getting yelled at by the same customers I thought would be ecstatic.

"What kind of clown show are you running?"
"How could you release this product with this defect?"
"I'll never buy another product from your company!"

I spent the next three months working to analyze, triage, document, fix, and communicate the particular defect. And it was a pretty bad one. We had an incompatibility with a growing player in our space. It was completely our fault and required a board-level change. There was no way around it–we had to recall tens of thousands of units from the field.

Another three months went by cleaning up the mess we created before it was time to reflect on what went wrong. Exhausted and depleted, I sat down in my boss' office where he asked an important question.

"How did we miss this in our testing?"
I replied that because we didn't have this brand in our lab, we

relied on beta testers who had it in their environment.

A few days later, I combed through threads and found an offhand comment from a beta user who said

“The product can’t seem to work with this brand. No big deal.”

Well, it turned out to be a huge deal.

This small, overlooked comment cost us hundreds of thousands of dollars in product cost, overhead, communications, and most importantly, customer trust. It took us two years to regain that trust due to a recall. A recall that could have been avoided entirely had we dug into that comment buried in a survey.

There are countless stories showcasing the impact of things missed during beta testing. If you’re reading this far, you probably have similar stories of your own.

- Mass-scale failures.
- Software bugs requiring engineers to work overtime.
- Angry customers.
- Social media blowing up.
- Executive team grilling you.

It’s easy to feel that you failed at doing your job as a product manager. But you shouldn’t. The reality is, being a product manager is a demanding job. And beta testing, as important as it is for success, can be one of the most frustrating phases of your product development journey.

But we’re here to help make it better. Bring order to the chaos. Help you effectively manage your beta programs at scale, so you never miss any critical feedback from your customers again.

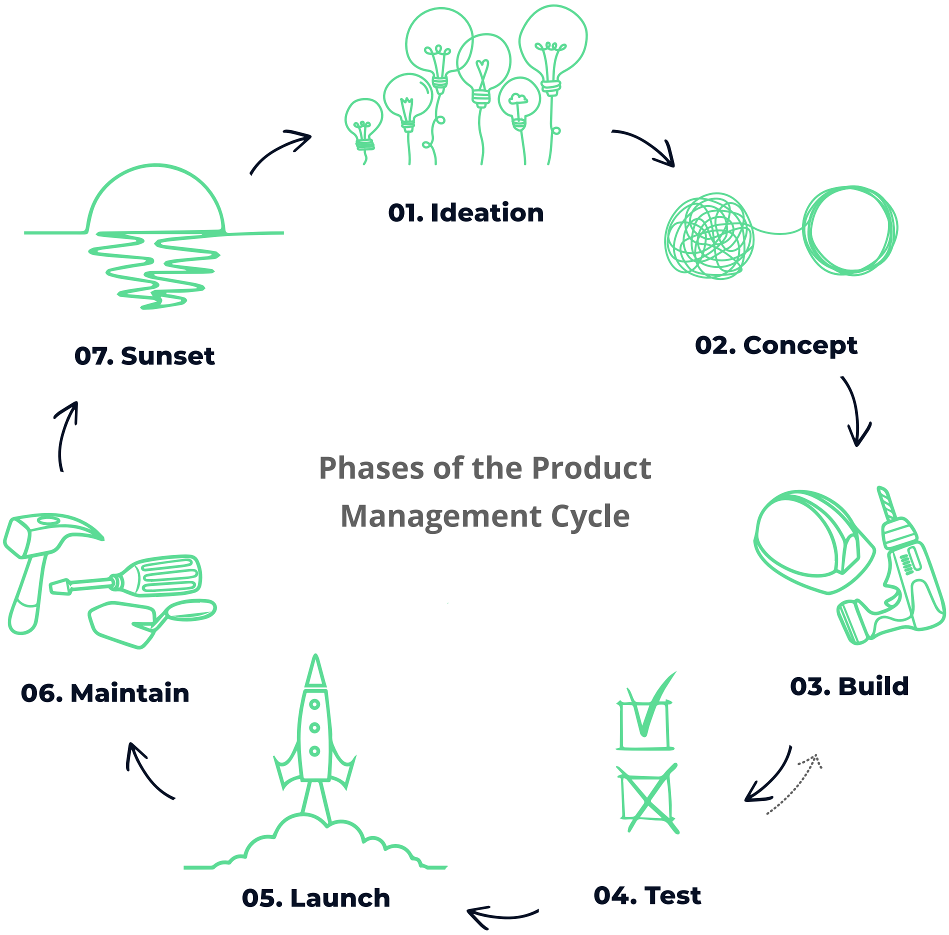
2.

The Product Management Cycle

The Product Management Cycle

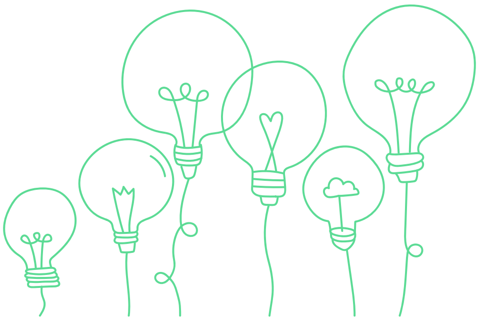
Before diving into the intricacies of beta testing, it's important to understand where it fits in the overall product development process. There are countless resources that explain product development in full detail, but I want to just give a quick overview to help set the stage for the rest of this book.

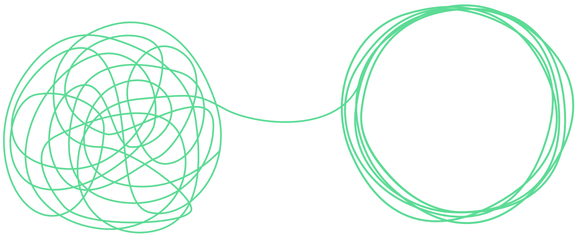
Whether you're building hardware, software, or an app, you go through multiple development phases either in a sequential (waterfall) or iterative (agile) fashion. The general phases are: **ideation, concept/business case, build, test, launch, maintain, sunset**.



01. Ideation

Your ideation phase can simply be a customer request or competitive pressure. It could be an innovative solution based on deep understanding of your customers' pain. It could also be an executive mandate to align the product roadmap with the company's mission. Or simply a technological evolution your space is going through.



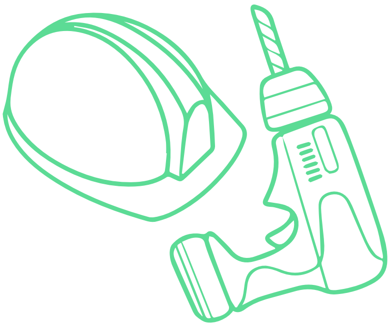


02. Concept

Building a concept and a business case to support your hypothesis could be a web mockup, 3D design renderings, or just words on a slide explaining the concept and main features. You want to validate two important pieces of information here: 1. Is this a needed product? And 2. Will our customers use/buy this product if we build it? If the answers to both are yes, you get executive buy-in to fund it, then it's off to the races with your development team!

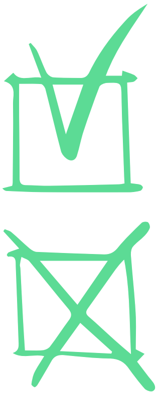
03. Build

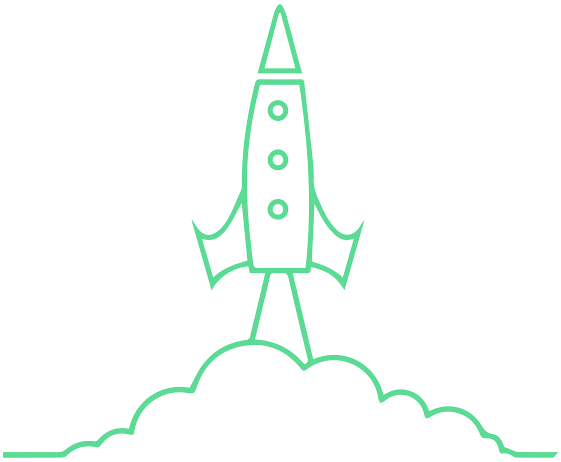
Building your product requires a more detailed level of specification, and by now you're thinking about a customer's experience and their interaction with the product. Typically, the building cycle is broken down into milestones: prototype, rough first version of the product, internal beta version, public beta version, then final version. In each milestone, the product is refined and improved until it is ready for production.



04. Test

The testing phase typically starts with internal testing. Your test scope and number of testers increases with each product iteration. Each milestone uncovers corner cases and bugs that require attention and resolution. Beta testing is incredibly important because it allows you to collect both product- and business-level feedback.



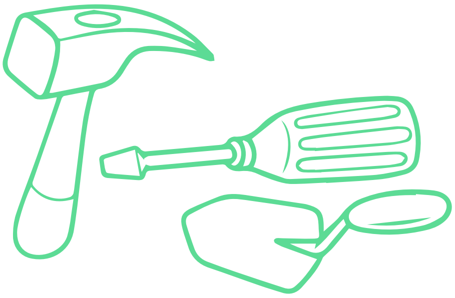


05. Launch

Launch day is always exciting, where the team feels like children on Christmas morning! The level of energy is unmatched. Launches are usually a company-wide event, and depending on the importance of the product, may align with a conference, company all-hands, or online event. No matter the format, it's an event the entire company rallies around to help deliver maximum impact.

06. Maintain

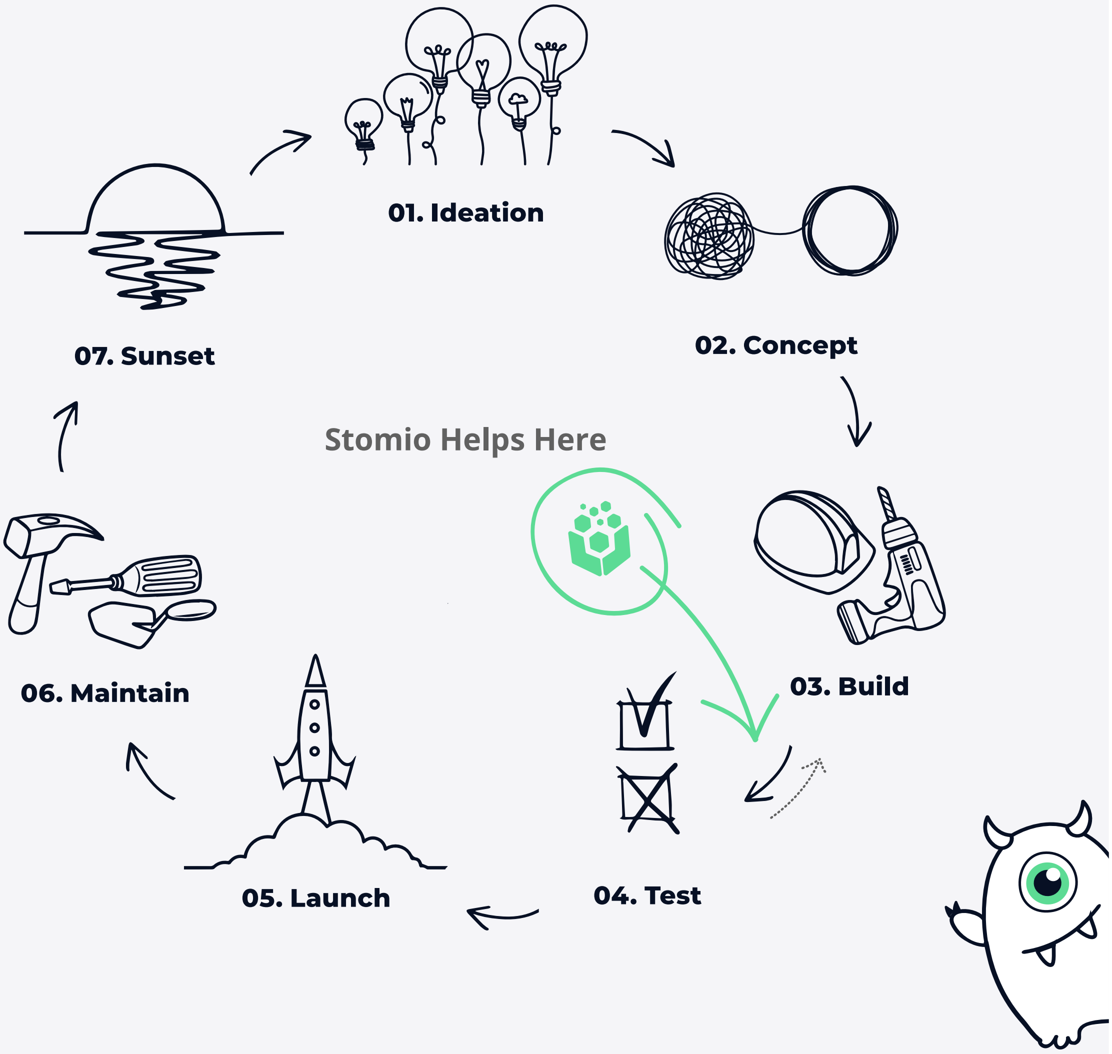
After the adrenaline rush of a launch wanes, you go into maintenance mode, which is usually broken into two segments: evangelism and management. Ideally, you want to spend the first 3-12 months post-launch evangelizing your new product. Attend conferences, support your sales team, and close big wins as you build excitement promoting your product and generating revenue. Eventually, you'll get feedback from all channels. You'll fix bugs, deliver new features, and deal with escalations until the product reaches its end of life.



07. Sunset

All good things must come to an end, including products. When sunsetting a product, it's important to provide a graceful exit and move your customers and company to the next best thing: either a replacement product or a new product that solves a newly developed problem.

How Can Stomio Help Me?



3.

Voice of Customer

Voice of Customer

Throughout your product lifecycle, your customer is the most important stakeholder. Unfortunately, your customers won't say early on if your product will be adopted by them or not unless they are experiencing numerous problems. What they will do is share use cases, pain points, points of view on how to fix a problem, and other parallel market solutions. They'll be your evangelists, your social proof, and valuable references when you launch. This is why talking to your customers (both current and prospective) is an essential part of being a product manager.

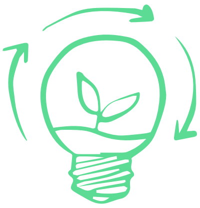
When you start building, you need to get into the nuts and bolts of what your customer experience will be.

The type of feedback you collect is different in each phase of your product development cycle. You'll also see differences in the way feedback shapes your product and the type of feedback you want to collect due to varying personas.

Early Stages	Building	Alpha Test	Beta Phase	Launch
Understand current pain points	Design choices require customer inputs	Focus on the core technology and use cases	Focus on validation and viability	Customers test the product and viability
5-10 customer interviews	3-5 customers (design partners)	10-20 customers (internal and external)	20-200 customers	First product buyers
Subjective feedback	Experience feedback	Actionable feedback	Insightful feedback	Use-case feedback

Early Stages

In the early stages of your product journey, look to get more "exploratory" feedback. Utilize open-ended questions to better understand current pain points about the problem your product can solve and what differentiates your future product in the current landscape. You will likely get a lot of subjective feedback, but luckily you don't need many customer interviews to find trends to build your business case on. Typically 5-10 customer interviews will give you 80% of the information you need. To achieve this, target decision makers and/or visionaries within your customer base who are comfortable with ambiguity and envisioning a future that doesn't exist today.



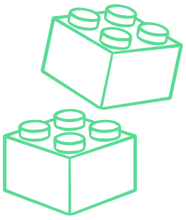
Understand current pain points.

5-10 interviews.

Subjective feedback.

Building

When you start building, you need to get into the nuts and bolts of what your customer experience will be. Inevitably, you'll run into multiple design choices that require customer input (because they'll be the ones using it). It's critical that you have a handful of customers available to give feedback within 1-2 days, so you can keep your design/engineering teams working. Those customers (ideally 3-5) can be your design partners, who will typically provide free-form feedback via email or video calls.



Design choices require customer input.

3-5 customers design partners.

Experience feedback.

Alpha Test

Once you get to your alpha test stage, your product is more-or-less formed. It has enough structure that it feels more real than imaginary. At this point, you need to expand your feedback group to include both internal and external customers. Internal customers (typically sales engineers or customer success managers) are critically important, since they constantly speak with tens of customers, and can be a good proxy for their collective voice. They are also needed to help sell your product when it launches. During the alpha stage, you're more focused on the core technology and use case your product is solving than perfection. Utilize a select group of 10-20 customers (internal and external) with hands-on experience/knowledge of the problem, so they can give you actionable feedback. You'll also want to start

Focus on technology and use cases.

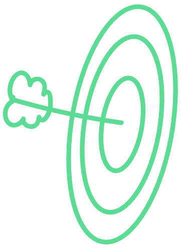
10-20 customers (internal and external).

Actionable feedback.

introducing a level of feedback structure to better manage your notes and ad hoc feedback channels.

Beta Phase

When you get to your beta phase, your product is as close to final as it gets. Your focus now is on validation. Does your product actually solve the problem? Do your target customers see it as a viable solution to purchase? You want to find corner cases and bugs in real life so you can avoid disasters at launch. You also want to validate product-market fit so you don't end up with an underwhelming product launch. At this stage, target 20-200 customers to get a statistically significant population covering as many of the corner cases as possible. You also want to create a good amount of content for your launch, and only a small portion of beta testers will be able to help provide such content (e.g. testimonials). Product users should be involved in your beta to give you insightful feedback as well as be your champion when the product launches.



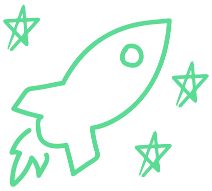
Focus on validation and viability.

20-200 customers.

Insightful feedback.

Launch

When you launch your product, it's showtime! While your sellers will start bringing in some customers, it is typical to have major customers wait or do their version of beta testing once the product launches. The perception among this group is to wait for roughly six months for your team to "iron out the kinks." Those customers will then seriously test your product and evaluate its viability to be adopted. Look to funnel their feedback, which is typically verbatim and use-case driven, through sales or customer success managers.



Customers test the product and viability.

First product buyers.

Use-case feedback.

Once your product starts gaining momentum, you'll switch back from validation feedback to exploratory mode: What features can you add to your product to unlock more use cases? What new pain points or bugs are discovered that your team needs to fix? What would a future iteration of this product look like?

Product management is not a sprint, it's a marathon that resets every time a product is launched.

4.

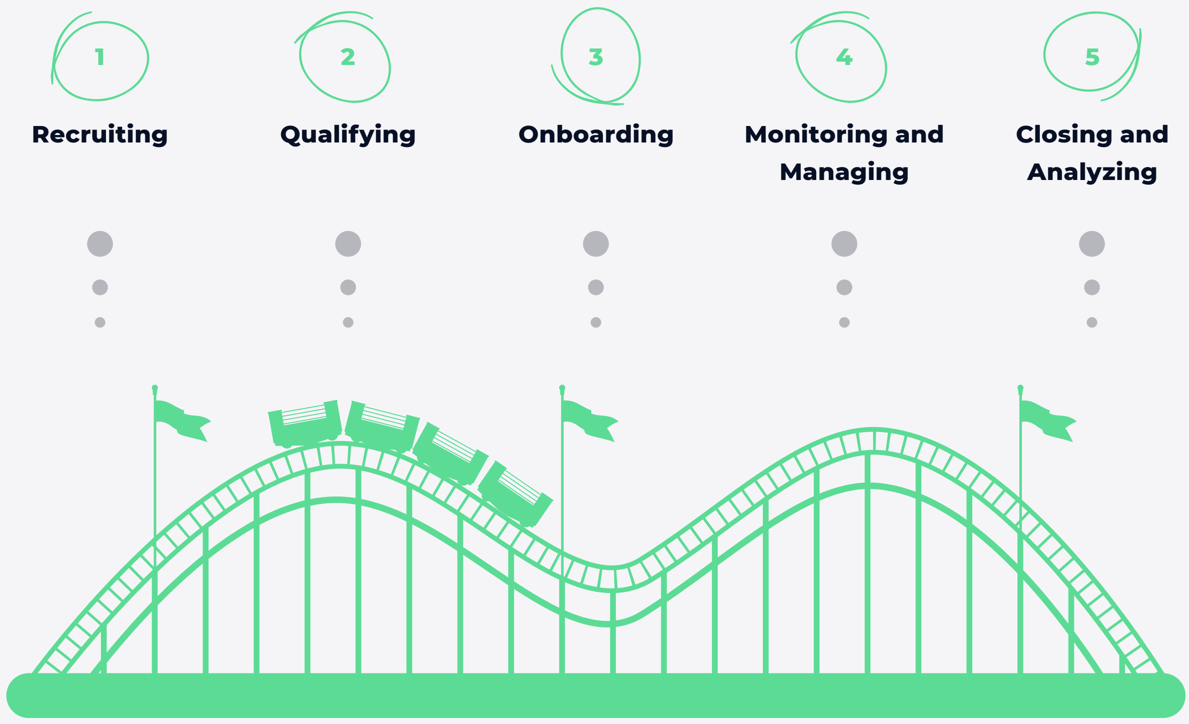
Beta Testing: The Process Within the Process

Beta Testing:
The Process Within the Process

Now that you understand the importance of beta testing in the grand scheme of things, let's dive deeper into this specific stage of your product journey.

A lot of product professionals find themselves in shock while beta testing their products. It's very much a process on its own, with multiple stages that each have their own requirements and challenges. Let's break it down.

Stages of Beta Testing



1 Recruiting

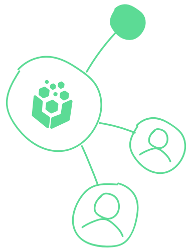
2 Qualifying

3 Onboarding

4 Monitoring and Managing

5 Closing and Analyzing

Stage 1 - Recruiting

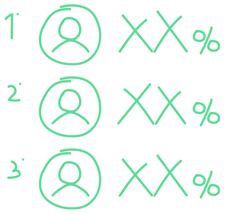


Recruiting your beta testers requires you to tap into your customer pool and look for those who are passionate about your product and willing to give you time and honest feedback. The challenge in recruiting beta testers is filtering out noisy requests coming from your sales team, who may just be using this opportunity for customer acquisition.

Best Practices

1. Reach out to your customer-facing teams to recruit beta testers.
2. Make sure there is a single, permanent place for prospective testers to sign up.
3. To avoid any friction, make sure you set the expectations with your testers and customer-facing teams that signing up does not guarantee a seat in the program.

Stage 2 - Qualifying



Your main focus should be finding good beta testers, regardless of the size of their business. You should certainly include your biggest customers to help promote exclusivity, but revenue shouldn't be the only criteria. You also need a diverse set of testers with different markets, use cases, install types and test environments. This increases the chances of covering more use cases and corner cases. Finally, you want testers who can be evangelists or good case studies for the marketing team to use before or at launch.

Ideal Beta Tester Profile

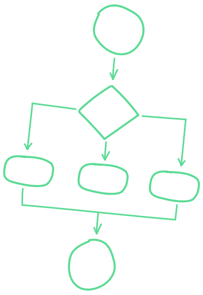
- Genuinely excited or skeptical about your product.
- Someone tech-savvy or a hobbyist who loves playing with products and breaking them down.
- Ability to add a unique use case or install environment.

Qualify your beta testers using custom fields

Read more →

Best Practices

1. Have a simple sign-up form to qualify your beta requests.
2. Remember that internal customers ARE customers, so make sure to include your top sales engineers and support team members.
3. Maintain your beta list for future products.



Stage 3 - Onboarding

As you begin receiving sign ups for your beta, you should start thinking about the tester experience. Here are some things to consider:

- Where to find relevant information for your test.
- Establish a structure through a minimum set of tests to follow.
- How testers can provide feedback to you.
- Who on your team is responsible for follow ups with inevitable questions and requests?
- The rules of engagement when it comes to privacy and security.
- What testers can expect from you at the end of the program.
- Where testers can go for support if they run into issues.

Thinking through all of these items ahead of time will guarantee a smooth experience for you and your testers.

Best Practices

1. Provide a level of structure or minimum test you want your testers to follow. Nothing kills momentum faster than just saying “Here you go! Let me know what you think!”
2. Provide your testers with a welcome message to generate excitement.
3. Establish a communication channel for your testers to provide feedback and ask questions.
4. Set expectations on follow ups and target time to finish your beta testing. Make sure those expectations are realistic, keeping in mind most of your testers are doing it out of good will.

Small steps, big results... driving engagement through micro-instructions

Read more →

Now that you have the infrastructure established for your testers, send them your welcome message and make sure to include all important information when it comes to rules of engagement. Typically, you want to have an NDA in place with each tester before giving them access to any details surrounding your beta program. Also, be sure to store those agreements in a secure place for future reference.

As you receive NDA sign ups, start providing access to your beta testers (software products) or shipping beta units (hardware products). Make sure to include instructions on where to find relevant information and how they can access your team. Strive to make their experience as smooth as possible.

Stage 4 - Monitoring and Managing

Your beta testers will come online gradually. Just like anything else in life, there is an initial moment of inertia where everyone is excited, but that can quickly fade if left without care and maintenance. It is your responsibility to keep this newly formed community engaged and motivated, so you can get the most out of your beta test.

Your goal in this initial stage is to simply keep your testers engaged. However, this is easier said than done. If you don’t provide clear expectations to your testers, or if they are not the right fit for your product, they will disengage. If you don’t provide a base level of instructions, they won’t care enough to provide feedback. If you don’t stay on top of their messages, they‘ll stop communicating. This may sound like a lot of work, but it is the reality of collaboration. And beta testing is essentially a collaborative sprint between you as a product team, and your customers. The more effort you put into it, the more you get out of it. More importantly, the more engagement you have from your beta testers, the more use cases, bugs, insights, quotes, and validation you’ll receive.

Use tickets to capture qualitative tester feedback

Read more →

Best Practices

1. Stay on top of the feedback you receive from testers. Acknowledge every message, good or bad. Ask follow-up questions and provide any specific updates to keep testers engaged.
2. Be proactive and give periodic updates to all testers. You will have releases and capabilities added throughout your beta, so be sure to announce them.
3. Gamify the experience to drive higher engagement. Create a leaderboard, announce a competition, or give group shoutouts. FOMO is real, and it works.
4. Make sure you flag and prioritize bugs reported to your engineering team, so they can be resolved quickly. Quick bug fixes lead to higher engagement. Make sure to update the tester who reported the bug so they can see the impact of their feedback.
5. Probe your testers for pictures, reactions, and delightful moments. Ask their permission to share those either internally or externally at launch.
6. Have fun! It's truly one of the most exciting stages when you get to share your hard work with a large audience. Your energy will resonate with your beta testers.

Stage 5 - Closing and Analyzing

Your beta program will conclude when the product is stable enough and the feedback from your beta testers is positive. However, sometimes you're running against a launch window deadline to meet KPIs (not a great reason to launch, but a reality).

While it's tempting to just let the program die naturally from lack of activity, it's important to keep everyone in the loop and send an announcement to your beta community with a summary of what happened and next steps.

Make sure you reward your active testers in more than one way. A public acknowledgement always goes a long way, and a symbolic token of appreciation usually sticks. We've heard from our interviewees that monetary rewards aren't always impactful, but limited-edition swag does wonders!

Give your testers directions on what to do next. If they are part of a software beta that will be launched behind a paywall, can they enjoy access to the feature for a period of time as a reward? If they have a hardware beta, can they keep using it? Should they return it? Providing instructions at the end of the beta period is equally important to the start of it.

Give your testers directions on what to do next.

Best Practices

1. Announce the end of the beta program to your community and thank them for their time and effort.
2. Send a summary of the activity and impact the beta test had on your product (number of bugs reported, number of use cases validated, etc.)
3. Reward your top testers with public acknowledgement and a token of appreciation.
4. Give instructions to unwind the program and keep the lines of communication open.

If you ran a successful program, you're now sitting on a mountain of data, feedback, messages, and pictures that you need to dig into.

You have two levels of feedback to analyze: urgent feedback from during the test (e.g. bugs reported) and insightful feedback after the test program is closed (e.g. roadmap ideas). It's imperative that you keep your engineering team informed when it comes to urgent bugs that are launch blockers. It's equally important to categorize and report on the learnings found during the beta program at its conclusion. The learnings and insights can be broken into segments that serve different internal audiences: non-critical bugs and issues to help the engineering team prioritize their work in the first three months of your launch, quotes and rave to feed your product marketing efforts, internal learnings to help your product teams improve in the next beta cycle. Doing this analysis might seem overwhelming, but the payback is 10x the effort in terms of professional and personal development.

Create a Beta in Stomio

Read more →

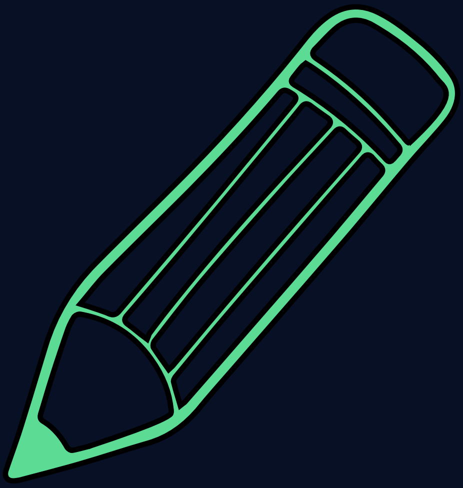
Best Practices

1. Categorize the feedback to serve various internal audiences (product marketing, engineering, product leadership, etc.)
2. Document your learnings from the beta program and keep it internally for future reference.
3. Share your findings from the beta program with other team members during an all-hands meeting or a lunch-and-learn to enable others in your company to be equally effective.

Multiple Disconnected Tools · High Overhead · Poor Engagement · Less Data

Faster Results · Rich Engagement · Meaningful Insights

5.



Your DIY Beta Stack

Your DIY Beta Stack

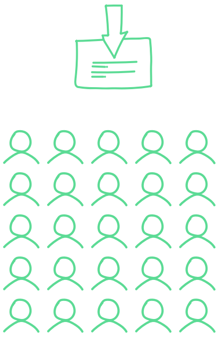
We learned in the previous chapter that beta testing is essentially a process on its own. Just like any process, you need to manage various stages using tools that serve various functions. Below is a beta testing stack example if you decide to do it yourself.

Stages of Beta Testing

1. **Recruiting**
2. **Qualifying**
3. **Onboarding**
4. **Monitoring and Managing**
5. **Closing and Analyzing**

Tools That Could be Used

Stage 1 - Recruiting



Before you start planning your beta program, create a folder for it in your company's shared drive so you can organize your files. It's always helpful to create subfolders based on the type of documentation you'll be archiving. Google Drive, Microsoft SharePoint, Box, and Dropbox are all popular options. To create a sign-up form for customers, look at using Google Forms or a survey tool like Typeform.

Evangelize and spread the word about your beta program to sales leaders using your company's messaging platform of choice: Slack, Microsoft Teams, and Cisco WebEx come to mind.

Also, don't underestimate the power of joining your customer-facing team's all-hands meeting. Look to use a couple of minutes to announce the program and give them clear action.

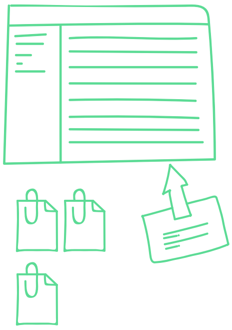
Stage 2 - Qualifying

Hopefully you'll receive significant interest in your beta program, which means you'll need to ensure those who signed up fit the needs of your product.

Be sure to ask qualifying questions in your sign-up form so it's easier to make a decision on a potential beta tester.

Look at results in your survey tool of choice and start creating a list of qualified beta testers using a spreadsheet.

Stage 3 - Onboarding



Once you start communicating with your beta testers, you'll need to establish a communication channel of choice. Email is ubiquitous and still the most used, but understand that messages can be missed in cluttered or busy customer inboxes.

Create a unique, dedicated email address to funnel all communication between you and your testers. It's also a good idea to establish a group channel (e.g. Slack group or WebEx room) for group messages and announcements.

It's always good practice to have your beta testers sign an NDA to protect you from leaks, either to the competition or simply via social media sharing. Your product is still in development, so you don't want any external pressure to launch simply because a beta tester leaked information. Get a simple 1-2 page NDA document in a PDF format and have your qualified testers counter-sign them. Keep a copy of each signed NDA in your beta shared folder for future reference. We recommend using an electronic signature tool such as DocuSign for capturing signatures, or simply embed the NDA in your sign-up form.

Before sending out communications to your qualified beta testers, you need to give them instructions on how to interact with you, when to expect the beta program to start, how long it will take, and how frequently you will communicate with them. Keep these instructions handy so you can easily copy and paste into your email.

Some product teams create hidden forum pages for each beta program, so beta testers have a repository of all documents and a place to interact with the team and each other.

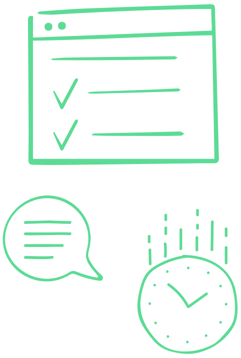
Write down the minimum test instructions that all testers need to follow in a word document or a spreadsheet and share it with you testers after they are onboarded.

How micro-instructions help you become a better researcher

Read more →

Now start reaching out to your qualified testers to officially welcome them to the beta program and set the expectations and rules of engagement.

Stage 4 - Monitoring and Managing



You will use the tools and instructions you set up during onboarding to monitor and manage your beta program. Stay on top of all email threads, keep an eye on the conversations happening in hidden forum pages, and be aware of conversations happening in communication channels.

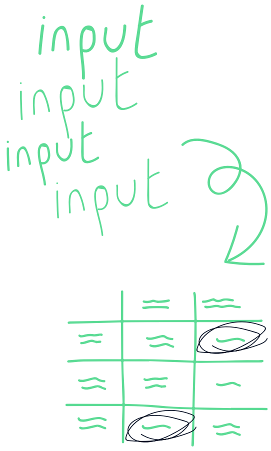
Not all testers will reach out to give you feedback, so you need to extract that feedback yourself. A few weeks after onboarding, be sure to send a feedback survey using your survey tool of choice.

Make sure to save those survey responses in a spreadsheet for future reference and analysis. Report any issues to your engineering team using your company's ticketing system (e.g. JIRA, GitHub, GitLab).

Track testers who are not engaged or responding to messages and keep reaching out for feedback. It's a numbers game, and the more testers you have, the sooner you can launch your product with confidence.

If you have a low engagement rate, consider other methods to apply urgency, such as phone calls or feedback meetings.

Stage 5 - Closing and Analysis



Send your closing announcement in all established communication channels: email, forum page, and collaboration rooms to make sure you cover all bases.

Use spreadsheets to dig into the feedback and identify any trends.

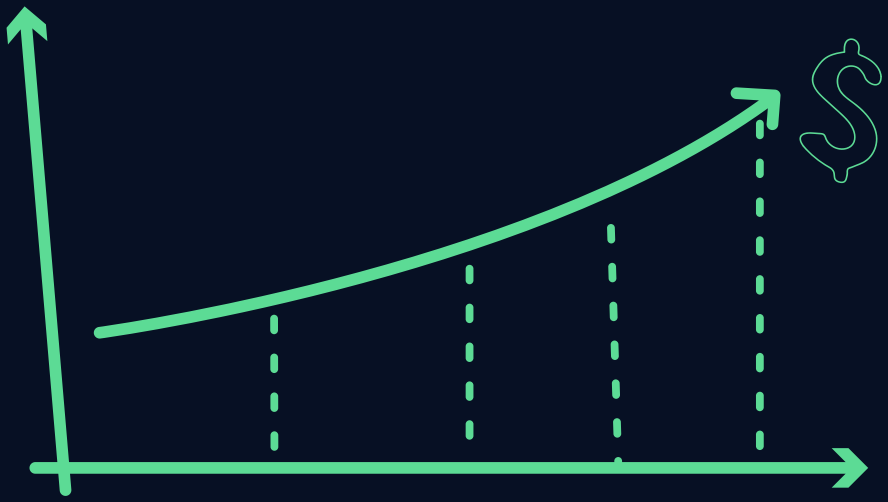
Organize your findings in a presentation that can be easily consumed by fellow product team members and can be shared in the next all-hands meeting.

No one needs another tool! Why Stomio?

Read more →

6.

Transform Your Beta Test from a Chore to a Sales Tool



Transform Your Beta Test from a Chore to a Sales Tool

The beta testing process can be overwhelming, and as a result, some product teams may simply treat the phase as a box to check. They may go through the motions with minimal preparation and low tester sample size. There is a fallacy that 5-10 customers can satisfy the need for any kind of research. This may be true when you are validating an idea or a concept early in the product life cycle, but when you have your product built, you need a statistically significant population of beta testers to satisfy the needs for that phase.

Your goals during beta testing are:

1. Make sure the product works in the customer's environment and for their use case.
2. Validate that customers will actually adopt this product at launch.

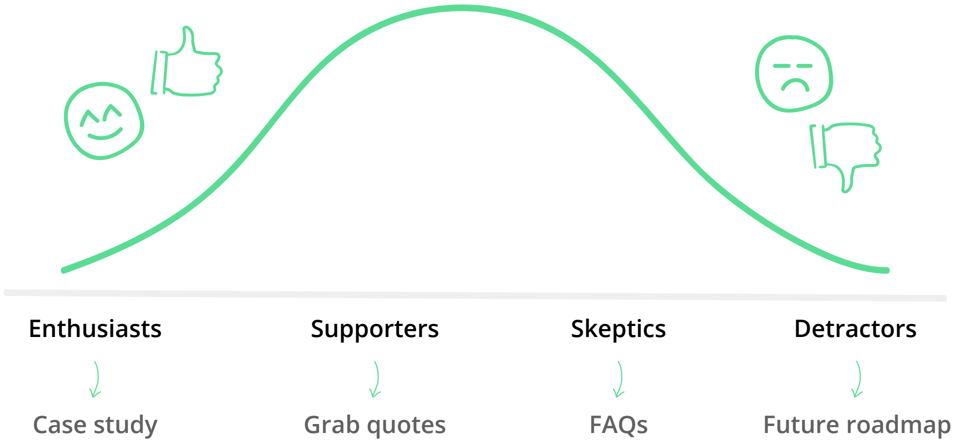
When you approach your beta phase with these objectives in mind, you will quickly realize that going through the motions or having a small beta testing pool is not going to get you to your target objectives.

The cost of (not having) the right tool

Read more →

Despite your best efforts, there will be breakage and customers who go dark. People are busy, so having a small pool of testers will only provide single-digit use cases. Any data obtained from such a small population can't be extrapolated to the larger customer base. Using DIY tools, you should expect an 80-90% drop rate from your testers. That means to have a minimum of 20 active testers in your beta program, you should set up the program for 150-200 customers.

The great news is that if you plan for a significant beta pool and get a higher engagement rate, you can leverage the additional data to aid your launch. Just like anything in life, you will have a normal distribution of beta testers: 10% enthusiasts, 40% supporters, 40% skeptics, 10% detractors. Here is how you can leverage the feedback from each group to your advantage.



Enthusiasts

Try to get them to participate in case studies. Dig deeper into their use case and how your product is solving their pain points. This group is a great asset to your sales team at launch.

Supporters

Grab quotes from them to use in your product launch. Ask for permission to reference them.

Skeptics

Use their feedback for an objection-handling sales book for your sales team and FAQs for your customers.

Detractors

Usually a great source of information for future innovation. Your product simply didn't address their pain point, which means it's an opportunity to fish for future opportunities from them.

All feedback is great feedback.

In other words, the only bad feedback is no feedback at all. The power of an effective beta program extends beyond your product launch. Your beta program is part of your company persona. If you run a thoughtful and engaging beta program, your customers will be more confident in your product and team. This results in higher brand loyalty and satisfaction, which leads to more available customers in the next beta program.

Your beta program is part of your company persona. If you run a thoughtful and engaging beta program, your customers will be more confident in your product and team.

By simply keeping track of testers and including them in future beta programs, you'll create a flywheel effect that will increase the quality of your beta testing pool over time. The ultimate goal should be to have roughly 100 high-quality, engaged beta customers who are motivated to provide quality feedback at each program.

Your beta testing phase is not simply a box to check or a way to cover your bases when the executive team investigates. It's truly a sales and customer relationship tool to create high-quality relationships with your customers and lifetime value for your business.

7.

To Beta Test or to Launch

To Beta Test or to Launch

With the amount of overhead and pain product teams go through during a beta test, there is a natural internal conflict whether to "just ship it" and deal with the fallout later, or do a robust beta test cycle. As a product manager, you find yourself in between your QA/Engineering team trying to perfect the product before launch, and your sales and marketing team trying to have more products to sell and talk to.

As a product manager, it's your responsibility to bring alignment between those groups, and your beta-testing phase is the ultimate battleground for this internal tension. So let's run through both extreme scenarios to see the pros and cons of each.

As a disclaimer, there is a lot of value to move faster with small, less complex products such as accessories and minor tweaks. My argument below applies to complex products and those that carry material impact to the business and/or customers.

Just Ship It	**Robust Beta Testing**	**Just Perfect It**
Minimal beta testing	Effective beta testing	Long beta testing
Fast-launching, but at an extremely high cost of your customers' trust	Launching when you get clear success metrics	A lot of time is spent on testing. Competitors are capturing market share
Product failures and bad customer experience	Product fits in customer needs	Growth rate suffers

Scenario 1 - **Just Ship It**

So you decide to skip over or do very minimal beta testing. This naturally means you're launching products much faster. It's very exciting to launch a product. The energy is unmatched and everyone is feeling great (except for your engineering team). Your marketing team will be busy with a new product to talk about. Your sales team will be happy in the short term because new products mean new opportunities and ammo to close their quota and get their bonuses. Your product team will also be excited because they're fulfilling their function (ship products) and can now focus on the next product to build.

That sounds like a great scenario, except it isn't. Optimizing for velocity to this level creates a big challenge for your engineering and product teams. Your fast-launch mantra means you're launching half-baked products that can be riddled with bugs, design flaws, performance issues, and a never-ending MVP.

Your customers will quickly realize the major gap between what your sales and marketing teams are saying and the reality of the product. They'll see a great potential but never a real delightful solution to their problems. In the case of hardware products, your risk envelope is much higher: product failures in the field that require a recall (super expensive), compensating customers for time spent replacing your product with another (or worse, a competitor), massive disruption to your supply chain because of inventory depletion due to warranty replacements, and extreme overhead on your product and engineering teams to fix the problem (I've been there, it's not a fun experience).

- In this scenario, you're launching new products quickly and constantly (good), but at an extremely high cost to your customers' trust and brand loyalty (bad).
- Quickly enough, your brand will be associated with terrible products that are buggy and never great.
- This means fewer sales and a recipe for overhauling the product development process/organization.

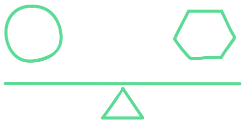
Scenario 2 - Just Perfect It

You decide to hand the keys of product launches to your engineering team. You want to make sure that your product is iron clad before any creative brief is written. All use cases must be documented, measured, tested, and passed. Your engineering team is happy (less issues to deal with post launch), your QA team is very happy (zero issues = success), and your program managers are happy (we stuck to the letter of the law for each phase gate). Your customers will experience a robust, thoroughly- tested product they can trust. And your sales team is confident that what they're selling actually works.

It sounds like another great scenario, except it isn't either. In this scenario, your product velocity suffers the most. Your product team feels chained with endless red tape and approvals. A lot of time is spent on testing and perfecting your product, and by the time you launch, you're already late to the party. Your competitors are capturing market share while you're seeking perfection. Your sales team is facing pressure to grow the business with existing products, while the market is evolving more rapidly than your product velocity. Your customers are frustrated with the lack of innovation and new use cases that can't be solved until you launch your product. Your product team loses control of their roadmap because they don't own or have any reliable way to predict when a product is launched.

- In this scenario, you are optimizing for quality and robust products at the cost of stagnation and growing the business through new product launches.
- Quickly enough, your growth rate suffers, your brand image is associated with being late to market, and the only way to get back to velocity is by over-investing in your QA team, which results in bad unit economics for your company.
- This is also a recipe for overhauling your product development process and organization, so there is a balance between quality and velocity.

Scenario 3 - Robust Beta Testing



So you decide to test during development, but your QA team can't feasibly cover all use cases (especially true in smart products). Your only great option is to run a meaningful beta program with your customers and partners. You start building your beta testing muscle to fit your needs. You get intimate with a handful of customers early in the process and then get a statistically significant population during your beta program. You drive engagement from your beta program and run it long enough to make sure your product is aging well in the field.

This is the optimal scenario. You're launching when you get clear signs that the product fits your customer needs and is reliable enough for them to use (your engineering team is relatively happy). You have confidence that the product is going to hit 80-90% of the use cases without an issue and have control of your roadmap with minimal variance (your product team is happy). And you launch with enough velocity that you're never late to the party and your growth is not stagnating (sales team is happy).

- Your product will never be perfect, but it will never be unusable either. You're almost never first to the market, but you're among the first wave. Your brand is associated with great products that are of good quality and improved reliability.
- This scenario is optimized for customer experience, which is the ultimate growth driver for any business.
- The only downside to this scenario is the need to run an effective beta testing program at a meaningful scale, which can be overwhelming if you're not prepared or your team is not investing in the proper tooling for your product organization to run effectively.

8.

PD

Product Manager vs Project Manager

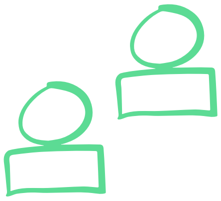
PM

Product Manager vs Project Manager

I want to end this book with an important topic. I have had the privilege of working with numerous product teams and there is a clear distinction between a product manager and project manager (or scrum master), even though they work on the same team in the same organization managing the same product line.

Product Manager

A product manager exhibits the highest level of ownership. Their products are their babies. The product is an extension of their personal brand, identity, and sense of fulfillment. They take extreme pride in their products and feel a deep level of empathy with their target audience. Their passion is undeniable and they can talk about the smallest details with their customers as well as paint the rosiest pictures of what the future can look like, with their products at the center. Product managers are fierce advocates of their customers in boardrooms and engineering meetings.



Project Manager

Project managers are executioners. They perceive the job as simply relaying information. Information from customers to engineering. Information from engineering to marketing. Information from sales to customers. The act of managing a product is more important than the goal of building the product to begin with. There is little passion and purpose to the product and how it can transform their customers' lives.

Product Manager vs Project Managers

Exceptional product organizations realize the importance of project managers and heavily invest in the success of their product managers by enabling them to solve customer problems through delightful products. They optimize for customer experience instead of managing internal conflict. Exceptional product organizations are winning, and will continue to win, market share and trust in the era of product.

About the Author

Ayham Ereksousi

I'm a seasoned product leader with a long track record of successful products in the networking industry. I truly believe that beta testing today is a broken experience for both product teams and their customers. That's why I started Stomio, to empower product teams with unmatched customer insights throughout the product life cycle.

I've experienced first hand the difference between a successful team and a failed one. At the core of a successful team is data-driven decision making. I learned the importance of understanding the voice of the customer as early and as often as possible. I've seen product professionals fall into the false idea that customers "don't know what they really want." Top-down innovation definitely works when paired with strong voice of customer discipline throughout the development process. But getting meaningful feedback during development is really hard. I believe it shouldn't be.

I started my product career at Snap One (Nasdaq: SNPO), where I built a networking brand for the high-end smart home industry: Araknis Networks. After five and a half years at Snap One and more than 60 products launched, Araknis Networks is the dominant market share leader, trusted by more than 70% of smart-home integrators worldwide.

After my successful tenure at Snap One, I moved to San Francisco to join the Secure SD-WAN team at Cisco Meraki (Nasdaq: CSCO), where I had the privilege to lead a team of nine product professionals to drive the SD-WAN portfolio. My team managed MX, MG, vMX, and Z product lines and was responsible for launching high-impact product refreshes such as MX75, MX85, MX95, and MX105. We expanded Cisco Meraki's virtualization footprint to include public and private cloud offerings, paving the way for a compelling Secure Access Service Edge (SASE) offering. My team also introduced Meraki MG, which enabled Cisco Meraki to enter the growing wireless WAN market, a critical component of a modern Secure SD-WAN architecture.

I spent 10 years in product leadership, launching more than 70 products with hundreds of millions of dollars of impact. And throughout this time, beta testing was one of the hardest things to manage. That's why I started Stomio alongside co-founders Ibrahim Menem and Simon Martin–to help others through these pain points.

Stomio is the first product-led beta testing platform designed for modern product teams who want a better way to manage their beta testing cycles and drive more engagement from their customers.

stomio

Beta Testing Handbook

www.stomio.io