



JOBS-TO-BE-DONE

Intercom on Jobs-to-be-Done

Intercom on Jobs-to-be-Done

Drawing together the most valuable lessons we've learned so far, Intercom on Jobs-to-be-Done helps you understand the real job customers are using your product for.

...

If you'd like to read more, we regularly share our thoughts on Jobs-to-be-Done, product management, design and startups on our blog, [Inside Intercom](#).

Thanks to John Collins, Adam Risman, Ruairí Galavan,
and Zara Burke for their help creating the book.

Design and illustrations: Matt Yow

© 2017 Intercom Inc.

ISBN 978-0-9861392-3-9

Long paragraphs of impenetrable legal warnings aren't really our style so we'll keep it simple. Please don't share this book, rip off any content or imagery, or otherwise try to use it for your own gain. If you do share or write about it somewhere (which we actively encourage) please be sure to give Intercom appropriate credit and a link.

Table of Contents

3	Foreword
4	Introduction
5	Focus on the job
10	Understanding your real competitors
13	More than mattresses: using Jobs-to-be-Done research for software
19	Getting customers to switch
23	Where one job stops and another starts
28	A worse product can do a better job
32	Abandoning personas: the story behind Job Stories
36	Designing features using Job Stories
44	Asking why the right way
48	Conclusion

Foreword

As one of the earliest architects of its method and theory, I have been using Jobs-to-be-Done for more than 25 years. It's helped me create and launch more than 3,200 products and services to date, with many more to come. So it's a great honor to write the foreword to this book.

The team at Intercom started as a student and client of mine a few years back. Seeing how they have applied Jobs-to-be-Done at Intercom since then, they are peers at this point. Their focus on their market, the depth of application of the framework (across marketing, product, research, engineering, etc), and their confidence to make it all public for customers to see, is so impressive.

This book should help product managers, marketers and designers understand the causality of product design, sales and continued usage. Jobs-to-be-Done has never been more important, especially for web and software businesses. The low-hanging fruit of correlation and large sample sizes is fast running out. Focusing on the job, understanding true causality, is going to be the only way to get people to switch and use your product.

The most intimidating point of building a product is when you stand on the abyss of the unknown. You look forward and see nothing and everything at the same time. Jobs-to-be-Done is the light guiding you and your organization through the abyss.

On the surface, the job your product is being hired for appears to be deceptively simple. As one of the fellow architects of Jobs-to-be-Done theory, Clay Christensen, has said:

“In hindsight the job to be done is usually as obvious as the air we breathe. Once they are known, what to improve (and not to improve) is just as obvious.”

This book will make it clear the rigor of using Jobs-to-be-Done has a payoff of simplicity. But it takes hard work and collaboration to apply it right. This book is an excellent practitioner's guide on how to use the theory and methods of Jobs-to-be-Done successfully.

Bob Moesta, CEO, The ReWired Group

Bob Moesta is President and CEO of The ReWired Group. Along with Clayton Christensen, Professor at the Harvard Business School, Bob was among the principal architects in the mid-1990s of the Jobs-to-be-Done theory.

Introduction

It's very easy in software to convince yourself some app you just thought of is actually a brilliant idea. Technologists are drawn to new trends like moths to flames. Artificial intelligence, augmented reality, machine learning, wearables; it's very easy to join a couple of dots and believe you have a killer product on your hands. So what's really important is a lens that shows you the real needs of the customer. It lets you appraise new product ideas by simply asking: "Does this advance our ability to get the customer to their desired state?"

When we were first introduced to Jobs-to-be-Done, it quickly resonated with something we already intuitively knew – that great products were built around solving problems. What Jobs-to-be-Done gave us was a better way of framing what we felt – a vocabulary and framework to unite the team behind a product strategy. Over time, it turns out it's not just a great way for thinking about product. It's become a marketing strategy at Intercom, as well as informing research, sales and support.

Despite being almost 30 years old, the application of Jobs-to-be-Done to the software industry is still pretty new. Most of the writing out there caters to building physical products. You might have read why people buy milkshakes or mattresses. But how does this translate to buying software?

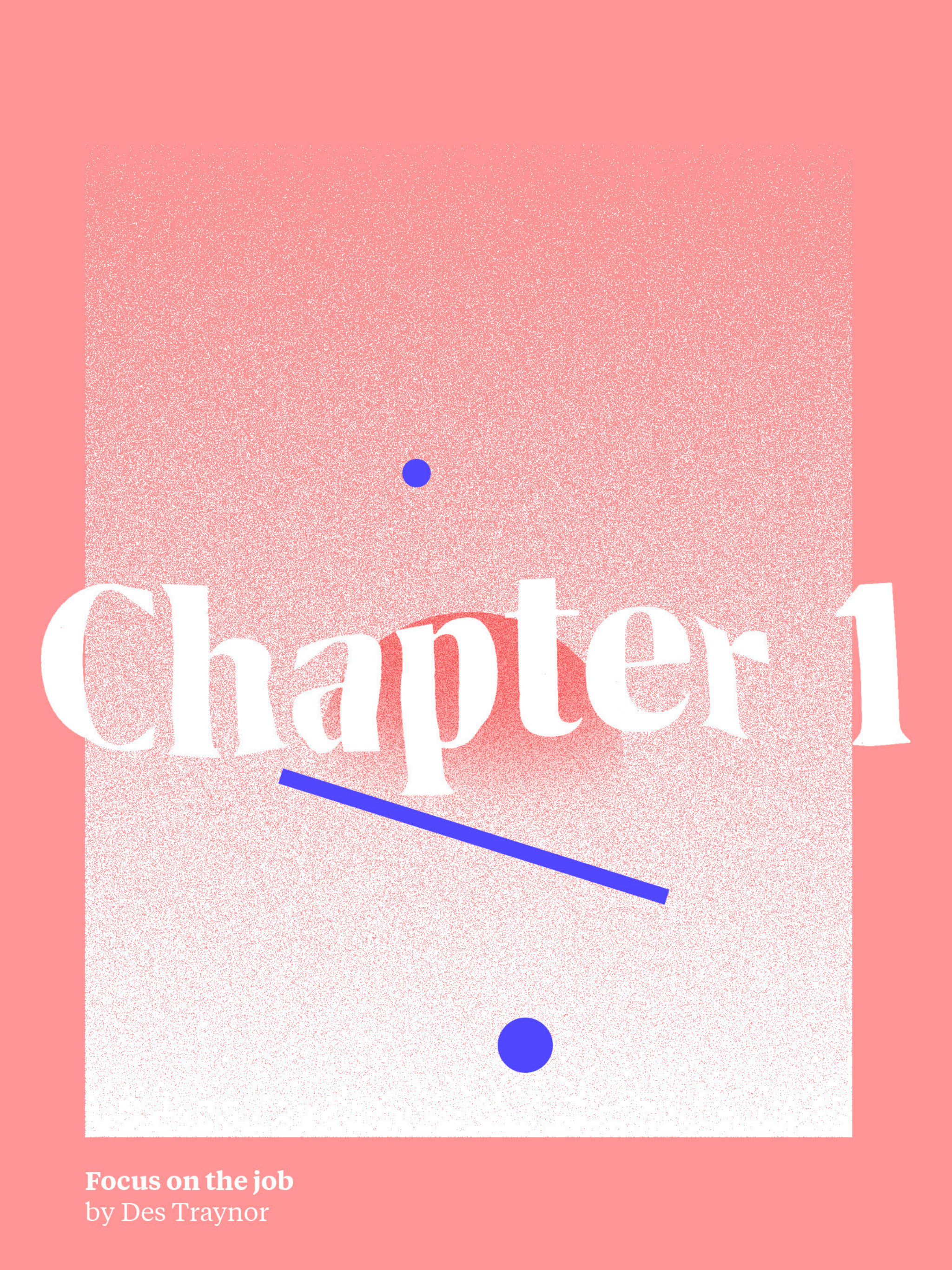
Compared to physical products, it's even easier for software to go off the rails. Before you know it, your simple time tracking tool has become enterprise version 4.3 of a project management app. Nobody sets out to build bloated software; it can actually come about through a series of tiny decisions from your founders, engineers, and product managers.

Jobs-to-be-Done helps you creatively limit the imagination of your product. It lets you focus on making things people actually want. When you're solving needs that already exist, you don't need to convince people they need your product. It's easier to make things people want than it is to make people want things. The challenge for any company is to understand what products are currently serving those needs, and improve upon that.

This book isn't a step-by-step guide to applying Jobs-to-be-Done to your business. It will give you some of the most valuable lessons we've learned over the last five years as we've applied the theory to our product. It'll show you how Jobs-to-be-Done can be applied to a modern software company.

But the lessons, like some of the jobs we discuss, are timeless for any business. You'll get a new way to think about your competitors, how to get customers to switch, how to define the scope of your product, and how to identify the job your product does. Our hope is you'll find your own inspiration in the lessons Jobs-to-be-Done has taught us so far.

Des Traynor, Co-Founder, Intercom

An abstract graphic design featuring a solid red background. A large, white, textured rectangle is positioned in the center. Overlaid on this rectangle are several blue geometric elements: a small circle near the top center, a larger circle near the bottom center, and a thick diagonal line crossing the rectangle from the lower left towards the upper right. The text 'Chapter 1' is written in a large, white, serif font across the middle of the white rectangle.

Chapter 1

Focus on the job
by Des Traynor

Personas will always have their place. If you want to create an advertisement that attracts 21-year-old men or 35-year-old professionals, personas will help create realistic representations of your target audience.

But if you want to build a great software product, making crucial decisions based around a series of personality traits won't get you there. That's because products don't match people; they match problems.

We learned early the outcome a person wants is much more important than the person themselves. Knowing it's a 37-year-old's hands on the keyboard rarely changes how you design your product to deliver their outcome.

By focusing on the job and the context of customers, you can develop and market products well-tailored to what customers are already trying to do. That's something a composite sketch of six different people just can't achieve.

...

Personas are a tool for sharing a common vision of a target user with everyone on a project. When everyone knows the sort of end users being targeted, it helps cut out some unnecessary debates.

A persona depicts what you need to know about a typical end user of your product to make informed design decisions. There are a few guidelines about how best to create, present and use them. Here are two important ones:

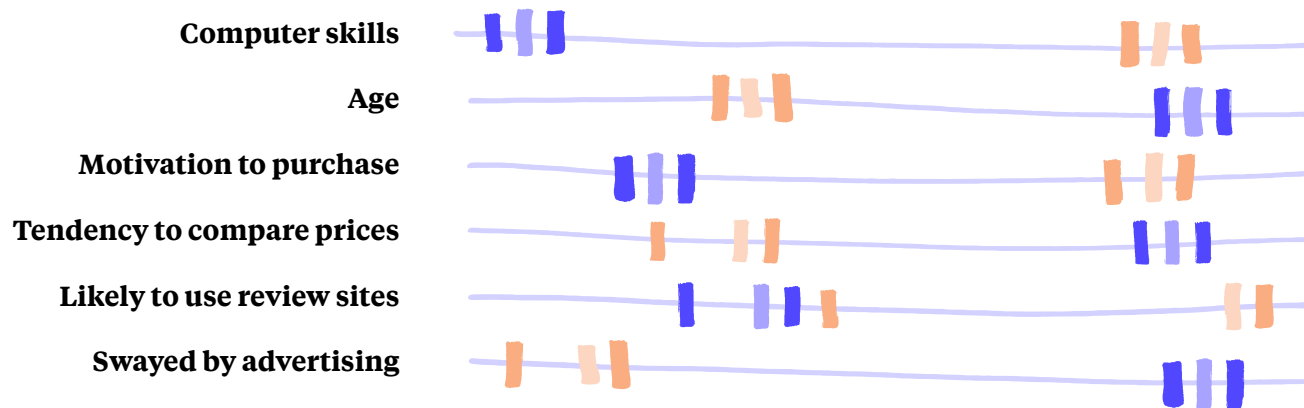
No nonsense. Every sentence in a persona should have a design implication. For example, saying the user is 72 and often texts their nephews and nieces could imply you need to cater for diminished eyesight, low computer skills and consider outbound SMS messages.

A shared creation. The project's stakeholders have to be involved in both the research and analysis involved in creating them.

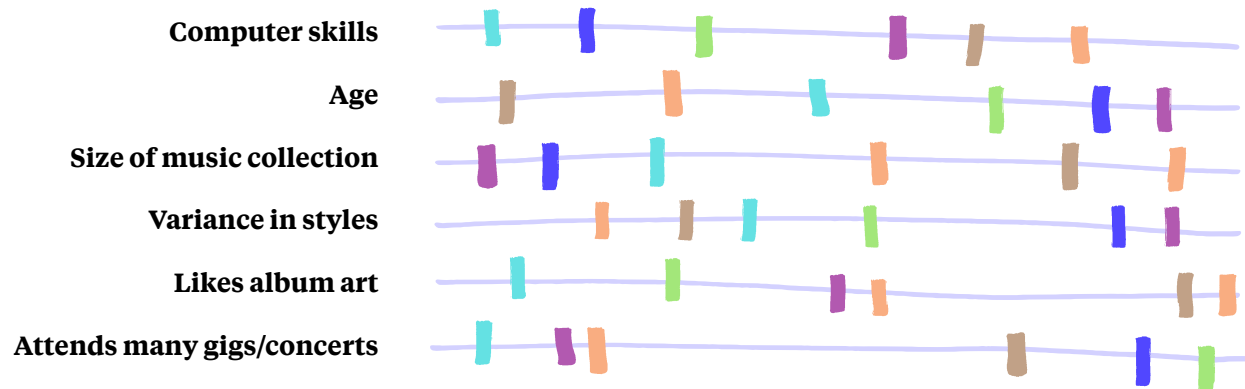
Personas are the end result of a chunk of work. As Jared Spool says, they are similar to holiday postcards; they're evidence a journey took place, but you can't buy postcards and think you've been on holiday. Personas work well when the user base can be broken down into different types of users with different needs.

But when you're building a product, that's not always the case.

Creating a persona



One way to create personas is to plot a variance of user attributes along scales, and see how users cluster. In the above diagram, each shade represents a different user; purple and orange highlight the clusters. This implies two user types you would explore further. Note there is lots more going on behind the scenes here; this is only a whistlestop tour.



If there are no key differentiators between all your users, you'll end up with useless, vague personas. 16-55, college-educated, good sense of humor, and other useless criteria. If a persona could be used to describe everyone you know, you can rip it up. If it's identical to the last ten personas your design team has produced, then you've a problem. Good intuition beats bad data.

So bad personas happen because:

- Insufficient or poor research has been done.
- Personas are the wrong tool for the job.

When personas yield nothing

Some products are better defined by the job they do than the customers they serve. For some products, customers come in all shapes and sizes, from all countries, all backgrounds, all salaries, and all levels of computer skills. The only thing in common is the job they need to get done. In these cases, it's best to get an intimate understanding of the job itself, what creates demand for it, and what you'd hire to do the job.

Clay Christensen, Professor at the Harvard Business School, describes this as job based marketing. He offers a masked example of a fast food chain looking to sell more milkshakes. Their initial approach was to study the users and make changes based on demographic analysis, customer analysis, and psychographic variables. This failed, and they sold no extra milkshakes. There was no meaningful insight to be found in analyzing the users themselves.

Instead, Clay suggested they focus on the job customers hire a milkshake to do. It certainly sounds weird (no one thinks of “hiring a milkshake”) but switching perspective offers new insights.

It turns out more than 40% of milkshakes were hired first thing in the morning – to provide sustenance for a long commute. So milkshakes were actually competing with bagels, bananas, and energy bars, rather than the rest of the chain’s breakfast menu. Milkshakes had advantages over energy bars and bananas – they’re tidier and easier to consume in a car. They beat bagels – they’re too dry and leave you thirsty. And they beat coffee – they’re more filling and don’t leave you desperate for a bathroom in the middle of a 40 minute drive. Once the chain realized this, they were able to make changes that made milkshakes the best tool for the job – a convenient, easy to consume snack to stave off hunger till lunch.

Focusing on the job

Here are some common jobs that have been around for a long time:

- Get a package from A to B with confidence, certainty and speed.
- Keep everyone up to date on a project they’re involved with.
- Get me face to face with my colleague in San Francisco.

Jobs can offer a long term view. Julius Caesar had to do the first job often, and he hired men and horses to solve it. Today we have FedEx. The job hasn’t changed.

Using this perspective can expose your real product’s competitors. Email is the biggest competitor to project management software. People hire email many times a day to keep their colleagues in the loop.

Economy travel and business travel are both capable candidates applying for the third job, though they’re looking for significantly different salaries. Video conferencing isn’t as capable, but is willing to work for a far smaller salary. I’ve a hiring choice to make.

When do you hire an app?

There are a several different jobs you might like to do once you’ve taken a photograph. Here are six:

1. Capture this moment privately between two people, so we can look back on it fondly in years to come.
2. Embarrass a friend in front of another friend.
3. Get this photo backed up online, so I can point others to it.

4. Get a copy of this photo to my grandmother who doesn't use computers.
5. Make this look cool and interesting. Like me. And then share it.
6. Get this edited and into my portfolio so people consider hiring me for future engagements.

In this case the products you could hire are Facebook, iPhoto, Instagram, Flickr, or maybe 500px. When you think about how many of these apps you use, you realize the job is the distinction here, not you.

You haven't changed. There is naturally an overlap between jobs, just as an employee does extra work in the hopes of a promotion.

Focusing on the job rather than the persona helps highlight how features like red-eye reduction, multiple photo sizes, or filter effects are only useful for certain jobs.

You may well be a talented photographer with more cameras than a Las Vegas casino. But when you hire Facebook, quality isn't your concern. People are.

Facebook photos are not about quality images. They do the job of instant sharing so friends can see and laugh at what's happening. Facebook would benefit more from a speech bubble tool (to make embarrassment easier) than they would from exquisitely preserving photo quality and camera details.

Why do people hire your product?

People hire your product to do the job of _____ every
_____ when _____. The other applications for this job
are _____, _____, and _____, but your product will
always get _____ the job because of _____.

As Peter Drucker pointed out, the customer rarely buys what the business thinks it sells him. Sometimes the type of customer will define the job they need done. Sometimes the job itself is the only driving factor. It's often hard to spot the difference. It reminds me a of W.B. Yeats poem.

"O body swayed to music, O brightening glance, How can we know the dancer from the dance?"

– "Among School Children" by W.B. Yeats



Chapter 2

Understanding your real competitors
by Des Traynor

When people think about their competitors they tend to look at what's closest to home. If I run a pizza slice store, and you run a McDonalds, we must be competitors, right? But real competition is usually a playing field away.

Jobs-to-be-Done gives you a much better lens to think about your true competitors. It gives you the situational context that triggers people to use products. To use the above example, if I know my customers are choosing my pizza because they only have five minutes to spare, and need to eat while walking to a meeting, then I know my competitor isn't McDonalds. I'm really competing with a Snickers bar and the hot dog cart around the corner.

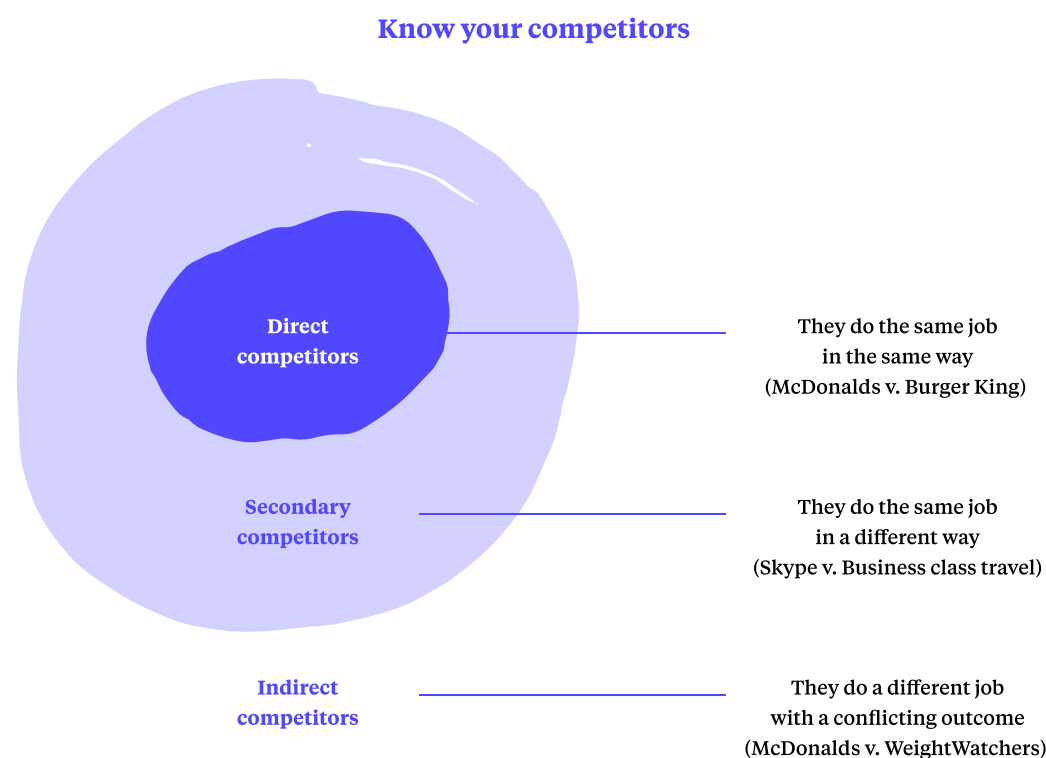
When you're blinded by thinking your competitors only exist in the exact same tool category you're in, disruption or destruction will come from oblique angles. The newspaper industry thought they were in the business of selling news printed on paper. Had they realized their business was "keep people entertained", or "keep people in the know", their new competitors like mobile games, Twitter, and Facebook would have been a more obvious threat.

So when you're thinking about competitors, it's best to ignore product categories and instead ask yourself who else is fighting for that same job.

...

Sometimes your customers really want to use your feature or product, but they also want something else that simply isn't compatible with it. People really want to be slim and healthy, but they also really want soft drinks and fast food.

McDonalds and Weight Watchers are selling wildly different products, but they're competing for the same customers. This is what we call indirect competition. Note this is different to competing on outcomes. Video conferencing and business class flights compete on outcomes, as they're both hired for the same job – business meetings.



In indirect competition, there are two different jobs a customer wants to do, but the jobs themselves are competing with each other. Software products experience these types of conflicts all the time:

- “I want to allow payments in my product, but I want to minimize the amount of third-party integrations we rely on.”
- “I want to add this analytics tool, but I also want to optimize response times.”
- “I want to know how my team spend their time, but also to show we’re a trusting work environment.”

It might go against logic, but humans are perfectly okay with maintaining multiple conflicting opinions and desires. We want to have our cake, and eat it too.

What can you do?

There are two conflicting forces here. The attractiveness of the outcome of your product vs the other product. Your marketing should work to make the alternative outcome less attractive, or reposition your product so the outcome is no longer in conflict.

A real example

One of Intercom’s customers was perplexed. Hundreds of companies had signed up for his A/B testing product, but very few had taken the plunge beyond a trivial test. Yet they all wanted to use the product, understood its value, and knew how to use it. Our customer used Intercom to message these users and identify what was going wrong.

The problem? As much they loved the idea of A/B testing their app, they also loved clean, legible, and maintainable code. They didn’t like adding JavaScript into their application to create meaningful tests, so they didn’t use the product.

To address these concerns, he added a message schedule downplaying the importance of clean code and upselling his product. He sent the following message to non-users on day three: “If no one is using your product, who cares how clean your code is?” On day seven, he sent another well-timed message: “This morning your team can add more code, or add more customers. Which do you want?”

These messages were effective. Many resulted in installs. Some resulted in technical debates. But most importantly, all of them produced extra insights into their business, which is what you need when you’re starting out.

The point is, customers don’t experience your product in a vacuum. They experience it alongside every other product, service, and idea fighting for their attention. Some of these will compete with your brand and some will contradict it. Understanding all these forces helps you counter them with your marketing efforts.



Chapter 3

**More than mattresses: using Jobs-to-be-Done
research for software** by Emma Meehan

I'll admit I was a little skeptical of Jobs-to-be-Done when I first joined Intercom as a Product Researcher. I'd ask myself: "Can't we just uncover the problems of our customers through classic user research techniques?" Oh, how wrong I was. It's given me a toolkit to uncover the motivations of why people buy our product, as well as unearth the real frustrations they face when using it. It's something other user research techniques just can't give you.

If you're new to Jobs-to-be-Done, you've probably asked the exact same questions I once did: "What has a mattress got to do with buying software?" This chapter will give you some practical advice on how to take the fundamental teachings of Jobs-to-be-Done and apply them to your software business.

...

There are lots of resources for getting started with Jobs-to-be-Done interview techniques. Yet most of these apply to physical products. If you work in a software company, you might rightly ask how the hell this relates to your product.

You can read up on [why a person bought a mattress](#), but not why they bought a piece of software. These interview techniques will tell you to hone in on the emotional triggers around a particular point in time that led to a person "hiring" or "firing" a product.

After conducting many, many interviews, we've found you can still apply the fundamental Jobs-to-be-Done interview techniques while adapting them to better suit the needs of software design.

Here's how we do that.

Extracting the first thought

One of the first things you'll notice about Jobs-to-be-Done interviews is an emphasis on "extracting the first thought" from a purchase or switching decision. For a physical product, this is easy – you can deep dive into the mindset and physical setting the customer might have been in during the time of the purchase.

- When did you start to look for another mattress?
- Where were you when you made this decision?
- Were you with anyone at the time?

When it comes to software, it can be tricky to extract the "first thought". Asking "Was it raining on the day you signed up for the calendar app?" is unlikely to jog someone's memory. We've learned the first thought can be multidimensional.

Take enterprise software. From a lack of budget, to a change in management, to multiple tools being used for one job, many things factor into why a company might switch to a new product.

These factors can be used in your interviews. Involve functional aspects from within a company,

along with emotional triggers from that time.

- What tool were you using before you bought <software>? Were you also involved in buying that?
- What was it like working in <department> for <company> back then?
- Can you remember if anyone else was involved in the decision? What was their role in the company at that time?
- Tell me about the <old solution>. Can you remember how well that was working? Was it just your department using it?

The four forces

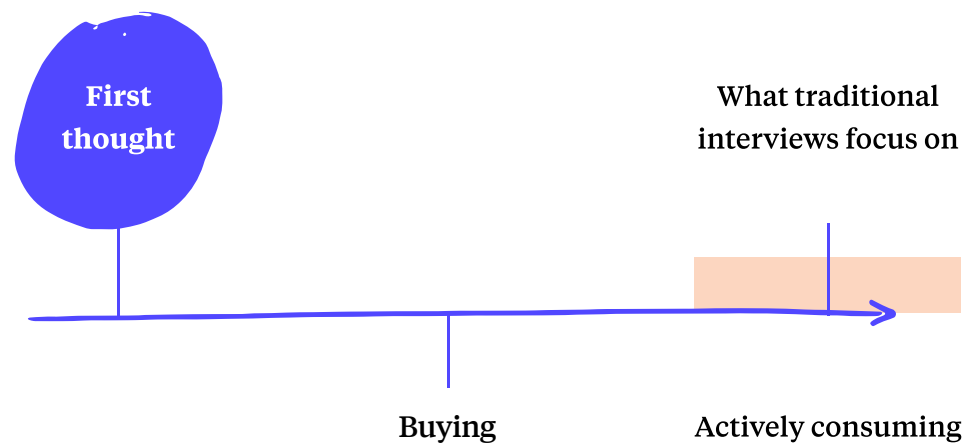
The ReWired Group have identified four forces pushing and pulling customers away from making a purchase. In our mattress scenario, the forces are:

- The push of what is happening currently: “This mattress is pretty uncomfortable. I’m waking up multiple times in the night with back pain.”
- The pull of a new solution: “If I get a new mattress, I can sleep better. I’ll be in a better mood at home and at work.”
- The anxiety of what could happen: “What if the new mattress turns out to be just as bad as the old one? I can only try it out for a few moments in the store.”
- The attachment to what you currently have: “I’ve had this mattress since college.”

Our reasons for buying software can lean more toward business goals like increasing engagement or revenue, rather than personal goals like increasing overall happiness (though if software is well-designed, it can do both). You could argue for software, the purchase decision can be influenced by multiple previous purchases rather than just the push factors of the current tool. In a software scenario, the forces are:

- The push of what is happening currently: “We’re not converting users at a rate that we’d like. We can’t afford to keep paying so much per month for this tool.”
- The pull of a new solution: “If we switch to a new tool that has more features around conversion, we can start hitting our targets.”
- The anxiety of what could happen: “What if the new tool doesn’t integrate as well as we’d like? We’ve tried three other tools for this job and none have been good enough.”
- The attachment to what you currently have: “We’ve got workflows and integrations set-up and it’d be a pain to get them set up again.”

Building a consideration set



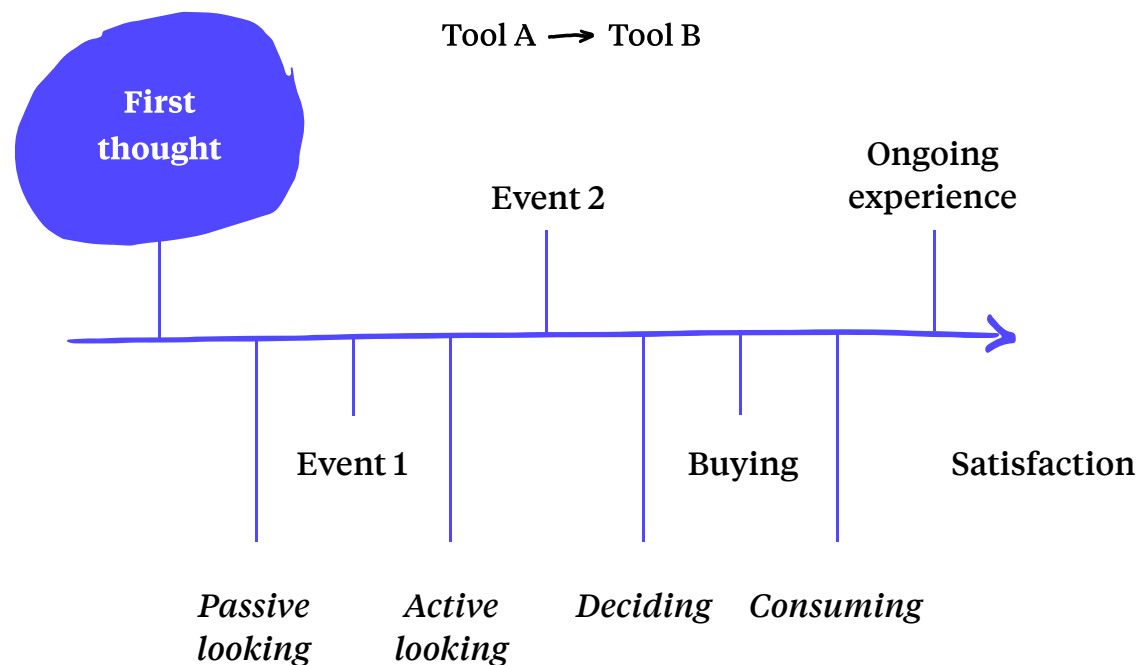
The consideration set involves the time in which a person (or company) is looking at all the possible solutions for a job they want to fulfill through a product. It's during this time “active” or “passive” looking takes place. Active looking usually stems from a moment of particular frustration the person has been through with the product. Passive looking can be ongoing for weeks/months while the person is uncertain if the product is doing its job well enough.

For software businesses, asking questions around the consideration set can help you uncover if there were situations where multiple people affected the purchasing decision:

- Once you figured out you wanted to buy a new solution, did you do much research to figure out which tool was right for your company?
 - Were you the only one who was looking for something else at that time?
- Where did you first hear about the tool you picked?
- Can you recall how you came to purchase the specific tool you did?
- When you were looking around, did you (or anyone else) try out any other tools?
What were they?
- How long did you look around before you clicked “buy” on the <tool's> website?

The timeline

Jobs-to-be-Done timeline

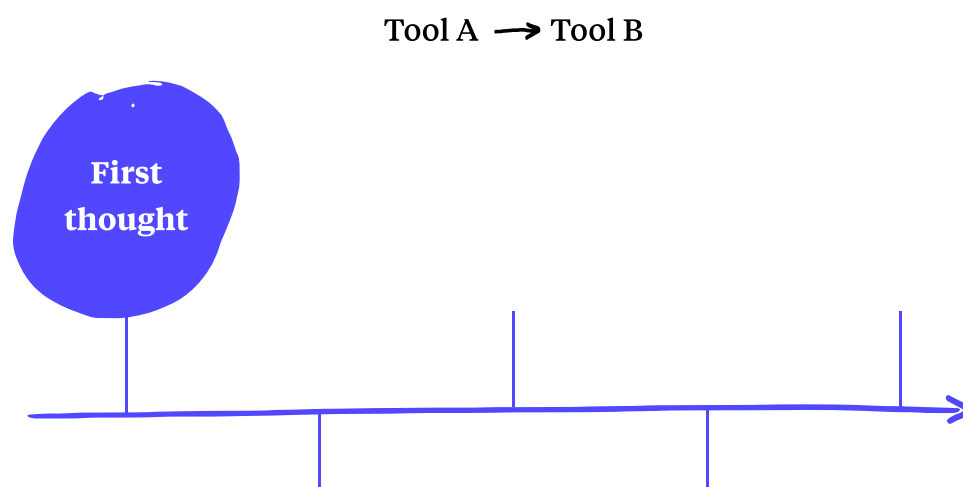


Creating a timeline of events helps bring the interviewee back through their purchase journey. We've discovered for software purchases, there are often multiple timelines at work.

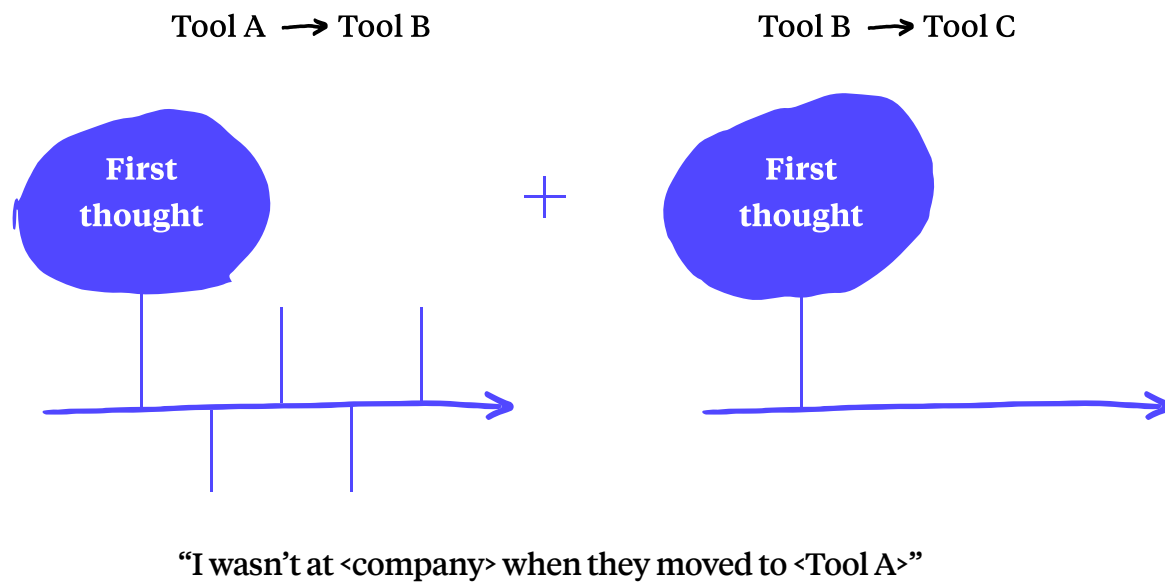
In the past, we've started interviews with the objective of uncovering why a company switched from Tool A > Tool B. We actually ended up with two timelines; one relating to the reasons why a company moved from Tool A > Tool B and then from Tool B > Tool C.

Once again, this is down to there being numerous people involved in the decision to buy software. You're really looking for the right people to talk to, the main decision maker. There are so many times you'll hear: "I wasn't at when we bought but I'm the main person who uses it!"

What you came to get



What you actually get



It’s like interviewing a parent who bought a mattress for their child. Talking to the child to understand how the mattress is working out now is useful, but really you want to talk to the parent about why they bought the mattress in the first place. Unearth the main decision maker behind buying software by asking questions like:

- What year did <company> move to <product>? Were you there at that point?
- Who decided to make that move? Are they still at the company now? What role are they in?

Knowing who the other decision makers are will give you a broader picture of where you need to innovate, and how you can support jobs your product is getting dropped for.

There’s more than mattresses

Jobs-to-be-Done can be great for understanding why companies hire and fire products, and unearthing areas where you can innovate. But remember what works for physical products does not directly translate to software products. Purchases are multidimensional, have multiple buyers involved, and are spread across multiple timelines. By adjusting interview techniques to your own needs, you’ll have a framework you can apply to any modern software company.



Chapter 4

Getting customers to switch
by Des Traynor

Most purchases are made through habit, with no real consideration of alternatives. People order the same type of coffee, in the same coffee shop, every morning, because the cost of reassessing their options every day isn't worth it. In short people rarely switch, but when they do, there's an interesting set of dynamics at play. Whether it's coffee or software most companies take the myopic route, and simply try to make their product look better than everyone else's. The appearance of your product is only ever one part of what makes people switch.

Switching is also about removing fear or hesitation associated with trying out your product and making sure you remind them of the problems with their current solution. There are very real reasons why people still want to use Microsoft Word, no matter how much better your text editor works.

The way you motivate somebody to make a switch is the same for a friendship, a relationship, or a software product – identify the struggling moments your customers are experiencing and build around that. Emphasize why the existing way does not make sense, why it's safe to switch to your product, and why they don't need to worry about leaving the existing way behind. If you can solve all those things, you'll get customers to switch.

...

“To create an anxiety relievable only by a purchase...that is the job of advertising.”

David Foster Wallace, *Infinite Jest*

The same can be said not just of advertising but of all marketing. Marketing increases familiarity, reminds consumers your product exists, and spreads product news. But marketing also does another job: it defeats inertia.

Why customers switch products

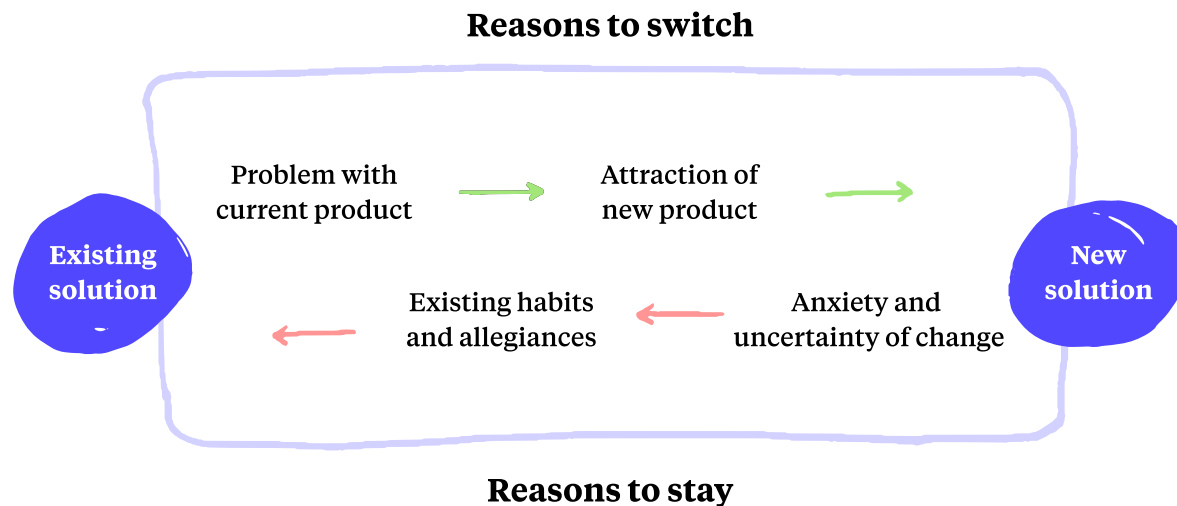
People don't hate progress, they just prefer inertia. This stops them from buying your product, even when it's the logical choice.

It's not that they're happy running projects over email, tracking expenses in files called “Expenses-march-12-v3-final.xls”, or taking notes from meetings in a programming editor. It's just not important for them to switch.

Switching is a big deal, you see. Customers don't buy a product, they switch to it from something else. Most businesses try to motivate a switch by having the best design, the best performance, or the most features. That only affects your product quality, one piece of the puzzle.

The ReWired Group identified four forces at play. Two of them are on your side, and two work against you:

The 4 forces influencing a customer switch



You can look at this diagram from a customer acquisition perspective – how can we get more people to switch to our product? Or you can look at it from a retention perspective – how can we make sure no one switches away?

Advertising lets you manipulate these four forces. Specifically you can:

1. Increase the push away: Show how bad their existing product really is.
2. Increase your product magnetism: Promote how well your product solves their problems.
3. Decrease the fear and uncertainty of change: Assure consumers switching is quick and easy.
4. Decrease their attachment to the status quo: Remove consumers' irrational attachment to their current situation.

You might remember an advertising campaign from a few years back that focused exclusively on influencing these four forces.

Apple's "I'm a Mac" campaign was hugely successful. It highlighted the pain points of Windows (1), demonstrated how great a Mac was (2), showed how easy it was to switch (3), and presented those who wouldn't switch as buffoons, reducing their attachment (4) to old habits.

In short, they created an anxiety relievable only by a purchase.

The 9X Effect

In a widely cited article called Eager Sellers & Stony Buyers, John T. Gourville presented what he calls the "9X Effect". He argues consumers overvalue what they already have by a factor of three, while companies overvalue their innovations, also by a factor of three. Here's what it looks like:

The 9x Effect

Happy consumers overvalue existing product by 3x

The Endowment Effect

People overvalue what they already have regardless of market

Status Quo Bias

People stick with a product even when a better one is presented

Gains and Losses Theory

Any loss in feature/performance is grossly overestimated

What customers think they have

3x **OLD** vs. **NEW** 3x

What customers
actually have

What companies
actually built

Innovating companies overvalue their new benefits by 3x

Self Design

Product based on designers/developers needs are historically hard for end-users to truly adopt

Over-Serving the Market

Companies improve products in ways that customers can no longer appreciate

Missing the Job-to-be-Done

Companies fail to understand what customers really use their product for

What companies think they built

To understand why inertia is so strong, you need only look at the comparison a customer makes when considering a switch.

This mismatch is why new products showing only incremental improvements over incumbents rarely get anywhere. It is the genesis of the idea that any new products need to be 10 times better to get mainstream adoption. It's only at that point switching becomes a no-brainer. That's where you want to be.

When marketing your product, consider all four forces to win customers over. Often your product is more than good enough, but there's another force blocking you. Advertise against that and you'll see more movement.

Understand that consumers overestimate their current solution just as much as you overestimate the quality of your product. This means products with only marginal improvements don't break through. The lead will only change hands when a 10X product arrives.



Chapter 5

Where one job stops and another starts
by Des Traynor

Software companies are always faced with the question of whether to add one more feature. The answer they arrive at is usually yes. When you're good at building software, you want to solve every problem in the world with software.

At Intercom, we've learned to appreciate where our product's jobs stop. If you follow Jobs-to-be-Done too rigorously, you can end up attempting to solve the entire working day of your customer through your product. You might start as a time tracker, add invoicing, then payroll, and before you know it you have perfect solution for a very small set of users. You have to learn where your product stops.

...

The most important thing a product manager does is decide where their product stops and someone else's product takes over.

If your product does too little then it isn't worth the cost of installation, let alone the actual purchase price. Similarly if a product does too much, it will clash with some other pre-existing software or workflow users are already happy with. It's a Goldilocks problem – you need to find the product that's just right.

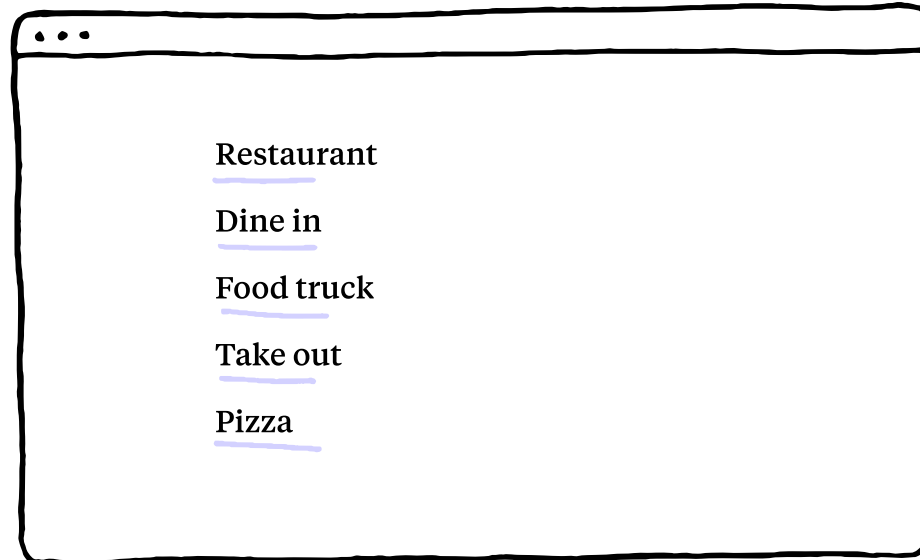
Understanding your product's workflow

Take a time-tracking product as an example here. At an absolute minimum, time tracking is just the sum of a list of numbers. Now, if that was all a software product had to offer, it would be useless. Excel or Google Sheets does that job already. It's at this point we realize simplicity is overrated. Even the best designed interface can't help a product that isn't earning its keep.

At a maximum, time tracking can involve project management, budgets, contractors, invoicing, receipt tracking and employee monitoring. Applications incorporating so many surrounding tasks tread on the toes of products already in place; in this case, Xero, Wrike, Basecamp, Teamwork, etc.

Products exist to solve problems that occur in a workflow. They have a start and end point within it. To understand where these points should be, you must understand the entire workflow. Let's break it down for a team ordering lunch every day.

Team lunch app



If you're building an app helping teams order lunch every day, the workflow might look like this:

1. Someone gets hungry.
2. He or she communicates this to the rest of the team.
3. Debate ensues about whether to go out or order in.
4. Second debate about where to order from.
5. Menus for different restaurants are passed around.
6. A decision is arrived at quickly.
7. One person is appointed to gather everyone's orders.
8. That person then places order.
9. That person communicates delivery time & cost to everyone
10. Time passes.
11. Food arrives, and is eaten.
12. Orderer checks if everyone paid enough & who still owes money.
13. Finances are settled, or the settlement is postponed until tomorrow.
14. Some will talk about the food on Twitter or Facebook. Some will post pictures on Instagram. Others will review on Yelp.
15. Everyone returns to work.

When you understand the full workflow, you can focus on the most concise, painful subset your product solves, or alternatively the piece you can make more fun or interesting. Ask yourself – is my product a vitamin or a painkiller?

Where should you start?

Start your product at the first step where you can add value. For our lunch example, that's probably step four. Starting any earlier would mean taking on chat products or email, rarely a good idea.

A real world example would be Instacart. Instacart is an online grocery delivery service. They could have tried to replicate the entire grocery supply chain – buying warehouses, trucks etc. But Instacart couldn't add value there. The first point they can add value is actually ordering the groceries online, something big retailers had traditionally struggled with. By understanding the entire grocery workflow, and the point at which they could add real value, Instacart designed a great solution.

And they did so by leveraging the existing grocery store infrastructure, rather than trying to build a solution from scratch.

Other products do this well too. Instagram starts with importing your social network. A time-tracking product could start by importing projects from Basecamp. Good APIs and import features help your users get off to an easy start.

Where should you stop?

Your budget, whether time or money, should restrict but never define your scope. A large budget should define how well a problem is solved, never how many problems are tackled. Attempting to tackle an entire workflow from start to finish for all types of users is near impossible.

Your product should stop when the next step:

- has well defined market leaders looking after it (e.g. Apple, Netflix, Expedia), and you don't intend to compete.
- is done in lots of different ways by lots of different types of users (e.g. trying to process salaries in a time-tracking app would be difficult).
- involves different end-users than the previous steps (e.g. managers, accountants, etc).
- is an area you can't deliver any value.

Identify and eliminate meaningless steps

Email client



If a user is finished with your app and their next step is to download a file so it can be emailed elsewhere, that's a meaningless step. If it's to export their expenses to CSV so that file can be downloaded and re-saved as XLS then mailed to their accountant, that's a meaningless step. If completing a project then means downloading all the files, zipping them and emailing them to the client for safe keeping, that's a meaningless step. Emails are almost always a source of meaningless steps – they rarely carry enough information (“Someone posted a comment”), or they don't link up the obvious actions (confirm, mark as resolved etc).

Any time your user is clicking through a defined series of steps, adding no insight and making no decisions along the way, it's a sign there are steps to be killed.

Future iterations

Always fill in the gaps before expanding the product. The shift from shrink-wrapped to SaaS rewards products that are reliable and complete, not buggy and bloated. Expanding your product to solve a larger problem can work wonders, but can only be done on a solid foundation. If your fundamental product isn't holistic, then adding more features only makes things worse.

So much is written about the pursuit of simplicity these days but often there is a confusion. There is a fundamental difference between making a product simple and making a simple product.

Making a product simple emphasizes removing all unnecessary complexity so every user can solve their problems as efficiently as possible. Making a simple product, however, is about scoping down and choosing the smallest subset of the workflow where your product delivers value. This minimum viable product approach runs the risk of being labelled a point solution, or worse, “a feature but not a product”.

When shooting for a “simple product”, be careful where you draw the line.



Chapter 6

A worse product can do a better job
by Des Traynor

Most products are obsessed with the intricacies of their own categories. Weather apps focus on precipitation depths, air pressure, and all sorts of forecasts no one really cares about. Their users are usually trying to answer simple questions like “will it rain?” or “how warm will it be?”

We learned this one the hard way in Intercom. After we launched our maps feature, it proved disproportionately popular. We couldn’t work out exactly why, but we were happy to see the engagement regardless. It turns out people were using the feature for a very specific job, one we had totally stumbled into and certainly didn’t design it for.

Paradoxically, to make it a better feature we made it a worse map. And we learned a lot from that.

...

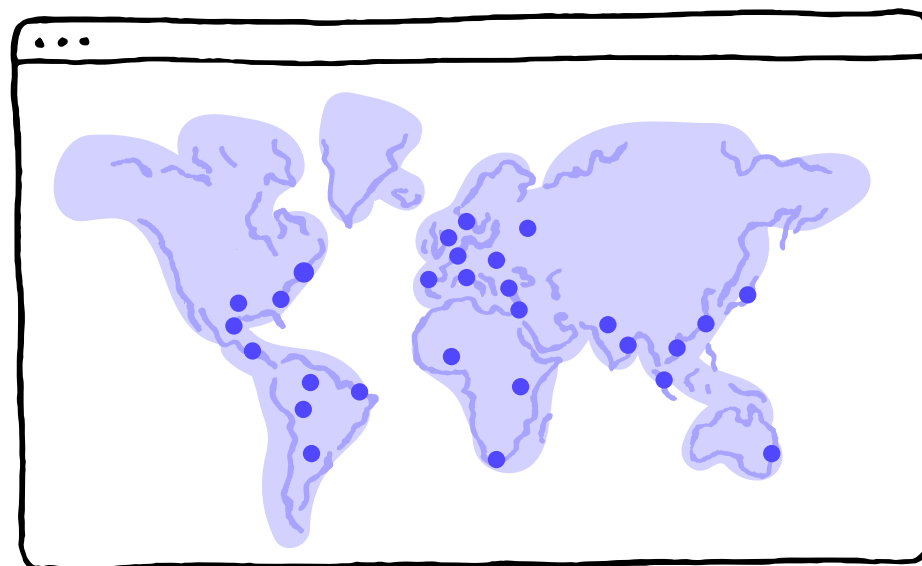
Customers will always surprise you with the creative ways they use your product. They don’t do it deliberately – they’re just adapting your product to their needs.

Remember the earlier Peter Drucker quote about customers rarely buying what the company is selling? The implication is to improve a product you must first understand what it is being used for.

Let’s explain with an example.

Not long after we launched Intercom, we added a map feature, so you could see where your customers were around the world.

Intercom map



It was a classic “this is cool but we don’t know why” type of feature. And its traffic showed us that it became popular quickly. But marketing the map as a feature was difficult. It was hard to work out why you’d use it.

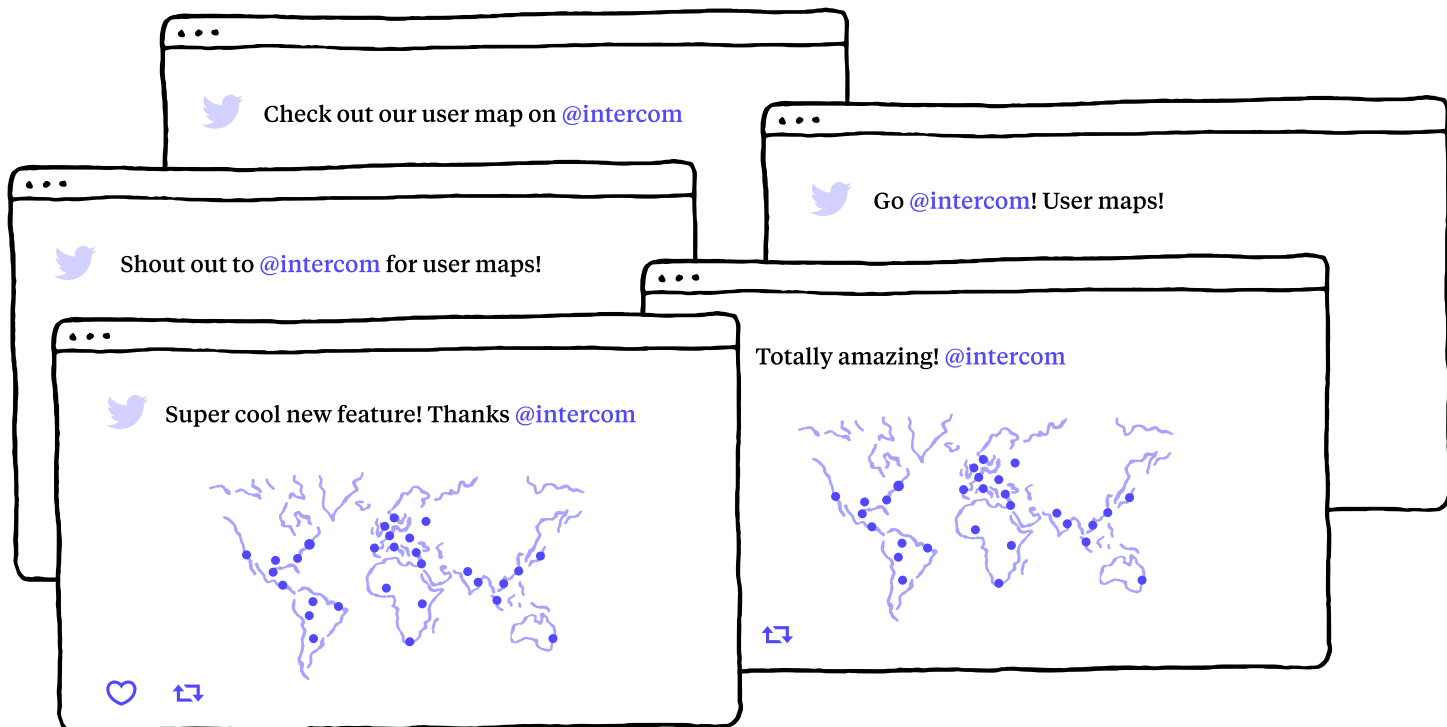
Here's a few ways we thought it could be used:

- Work out where you had most customers? No, lots of products do that already.
- See what customers are in a given city? No, our user list does it much better.
- See how many users you have in a given country? No, our user list does it better too.

So what does it do, aside from look impressive? There were three ways the map was used:

- People like to show it off at trade shows & conferences.
- People like to show it off on Twitter.
- People like to show it off to investors.

So what job did the map do? It looks impressive, and it makes our customers look impressive.



Improvement based on usage

If we tried to improve the map before we knew how it was used, we'd have tried to make a better map. Here are the types of things we would have focused on:

- geographical accuracy
- clustering
- better country/city borders
- drag to create "regions"
- various other cartographical improvements

All of those "improvements" would have taken us months and would have resulted in a worse product. Because the customer wasn't buying what we thought we were selling. It's not a

map. It's a showpiece.

So what would make the map better at that job?

- A map designed to look good, first and foremost.
- A map that hides sensitive data automatically, making it shareable.
- A map easy for customers to share.

So we built a beautiful animated map, with a unique shareable URL.

A worse “map” does a better job

When you focus purely on how a feature is used, ignoring the product category it's in, or the type of feature it is, you quickly learn how to improve it. Rather than trying to address a hypothetical markets of would-be consumers, you can make improvements that resonate immediately.



Chapter 7

Abandoning personas: the story behind Job Stories
by Paul Adams

At Intercom we're constantly re-evaluating our process for building great product. Asking ourselves what the best process is for building product people find valuable and useful, a product they love.

One thing we place a huge emphasis on is research. We hire people with direct research experience, and everyone on the product team talks directly to customers: the product managers, the designers, and of course our research team. We also hired a Director of Research (Sian Townsend, who previously led the research teams for Google Maps) much earlier than most other startups.

While it's obvious you should be talking to customers frequently to try and understand their needs, it's not obvious what the best tool to do that is. We constantly think about things from first principles, and so very early on we applied that lens to how we were talking to customers.

We were big fans of the Jobs-to-be-Done framework, but most of what was written on Jobs-to-be-Done was applied to milkshakes and chocolate bars. There was little published research on how to apply Jobs-to-be-Done to software. So we created our own process based on what we knew.

Over most of my career I used personas and scenarios to understand customers. Popularised by Alan Cooper in *The Inmates are Running the Asylum*, they have become one of the most widely used tools in a research and design team's toolkit. Cooper also wrote a fantastic book called *About Face*, and I recommend it to all designers who join my team. But I tell them to skip the chapters on personas.

When I worked at Google, I created dozens if not hundreds of personas across many projects. We followed Cooper's method carefully, along with iterations of our own. Universally, I found their value limited. They often helped with building empathy amongst employees who were disconnected from their users, but they rarely led to breakthrough ideas or fresh thinking about a problem.

And we have never used personas at Intercom.

The first time I really started to question the value of personas was when I left Google and joined Facebook. Facebook has amassed an incredible quantitative data set about what people actually do with their product, and talking with the data science team never failed to educate me.

One of the striking things about this data was how similar people's behavior was. Personas had led me to believe that people are really different, with really different goals. But the similarities were far greater than the differences, and across everything you can imagine – race, age, gender and so on.

For example, the motivations behind a married mother of three in the USA posting photos of a family barbecue are basically the same as a Korean teenager posting photos of the house party the night before. The goals and attributes look totally different, but their motivations are the same. Personas would never lead to the same product being designed and built for both these audiences. While best-in-class personas focus on goals (what drive people's behavior) as well as attributes, the reality is that most personas focus on attributes alone, even when goal driven.

Personas artificially break apart audiences. And critically, they artificially limit your product's audience by focusing on attributes rather than motivations and outcomes. Once I made this observation, it blew apart my faith in personas.

Designing for motivation is far better than designing for attributes. It's the key difference between personas and Jobs-to-be-Done. Personas look at roles and attributes. Jobs-to-be-Done looks at situations and motivations. Personas explain who people are and what people do. But they never fully explain why people do something. And why people do things is far more important.

So in mid-2013 at Intercom, we found ourselves in search of a tool that would help to uncover why customers do things. We were talking to dozens of customers each week, and our customer support team was talking to many hundreds more, gathering feature requests, and understanding problems and constraints of our product. Having this direct connection to our customers has been invaluable, but there were two challenges we needed to overcome:

1. People are experts in their problem, not the solution. However, it is more natural to suggest a solution in the form of a feature request. Describing a suggested solution is easier than describing a problem, but you need to go back to them with questions to really understand their problem.
2. When they reply, their initial answer will tell you what they want, in the form of attributes, but not why that matters. So you need to keep digging into their motivations. So it was critical that we found out what problem our customers were actually trying to solve, and why they needed to solve it.

Once we understood the problem, how could we boil these insights down into something actionable for the design team? Something concise, that was easy to communicate and remember. I can't begin to recall the number of times at Google that we had people participate in research and co-create personas with us, only to ignore them afterwards because they were too hard to remember and too detailed to parse. Personas are just not concise enough for fast-moving product teams.

Aside from personas, another very popular tool is user stories, popularised by the Agile movement in software. We had never used user stories either. For a start they are not empirically research driven, and are engineering driven, rather than customer driven. They are formatted to describe functionality to be built, rather than motivations people have.

After thinking about this problem from first principles, we invented Job Stories. It didn't have that name at the time (Alan Klement later named it for us), but the process was there, and was working well for us. We first surfaced it in a blog post lamenting the "[Dribbbblisation](#)" of design. We had been thinking long and hard about Jobs-to-be-Done, and ended up creating our own process that focused on situations, motivations and outcomes:

[When ____] [I want to ____] [so I can ____]

"When ____" focuses on the situation, "I want to ____" focuses on the motivation, and "so I can ____" focuses on the outcome. If we understood the situation in which people encounter a problem to

solve, understand the motivation for solving it, and understand what a great outcome looks like, we were confident that we would be building valuable product for our customers.

Job Stories are now a key tool for us. They ensure the team is research driven, and that we understand the problem so well that we can capture it in a concise format. It means our summary of the problem is actionable for all the design and engineering team.

Before we start any project at Intercom we create a one page brief for the project (you can download a copy of the template at bit.ly/intermissiondownload). This is very simple, and must be completed in one-page printable A4 page. It has a section on Job Stories, which helps remind us of the problem or opportunity we're tackling, and keeps us focused throughout the project.

Intermission Name
MM/YY

What problem we're solving and why

Job stories

How we will measure success

Scope

Personas definitely have their place. I've found them useful to get buy-in in political environments where people only pay lip service to customer's actual needs. They can also drive a stronger connection to actual users of a product. But personas are laborious to create well, focus too much on the differences between people, and are hard to remember and reference. Turns out, many people with diverse attributes have very similar motivations, those motivations are fast to research, and the focus on what you build can be captured in a series of short, memorable sentences.

If that doesn't convince you, remember that personas artificially constrain the total market for your product. That's why we're all in on Job Stories.

The background is a solid brown color. A large, white, textured rectangle is positioned in the center. Inside this rectangle, the text 'Chapter 8' is written in a large, white, serif font. Above the text, there is a small blue dot. Below the text, there is a thick blue diagonal line and a larger blue circle.

Chapter 8

Designing features using Job Stories
by Alan Klement

When Des asked me to contribute to Inside Intercom, I had just introduced Jobs-to-be-Done to my co-founders at Aim – an advertising marketplace for the real estate industry.

Our design process was riddled with ambiguity and lacked consensus. Myself, the CEO, the CTO and the head of design all had suggestions on what to design. But we didn't have any effective way to support one idea and challenge another.

Then it dawned on me. Without making any reference to Jobs-to-be-Done, I began pointing out the most successful suggestions my co-founders had proposed. I explained the struggling moment our customers faced, and described how their life would be better once it was solved. Then I explained how their suggestion was a good fit for this design problem. In effect, I described the customer's job and what it was like for them when it was done.

Using Jobs-to-be-Done language to communicate, we were able to knock out our first design in only a few minutes. We never looked back.

...

Personas and user stories made sense when customers and product teams were far from each other. That's no longer the case.

Traditionally, who the customer was and what they needed fell within the responsibility of marketing, business development, or even sales. Once this information was gathered, it would be synthesized into a portable format and then pitched over the fence to the product development team.

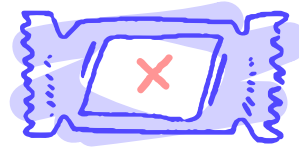
The casualties of this waterfall process are the subtleties essential to creating great products: causality, anxieties, and motivations. Jobs-to-beDone is the philosophy of focusing on these subtleties. And a granular way to bring this concept into a product is to use Job Stories to design features, UI, and UX.

How personas fail

Why did Peter buy the Snickers?



- Satisfy his hunger



- Is 35 years old
- Has a degree in marketing
- Likes peanuts, chocolate, nougat, and caramel
- Loves Snickers and eats one everyday
- Has an active lifestyle
- Has two dogs
- Hangs out at Jack-in-the-Box, but eats Taco Bell
- Stubbed his toe yesterday

As Des discussed earlier, personas are often defined by attributes that have nothing to do with causality. For example, someone's age, sex, race, and weekend habits doesn't explain why they ate a Snickers bar. Having 30 seconds to buy and eat something which will stave off hunger for 30 minutes does explain it though.

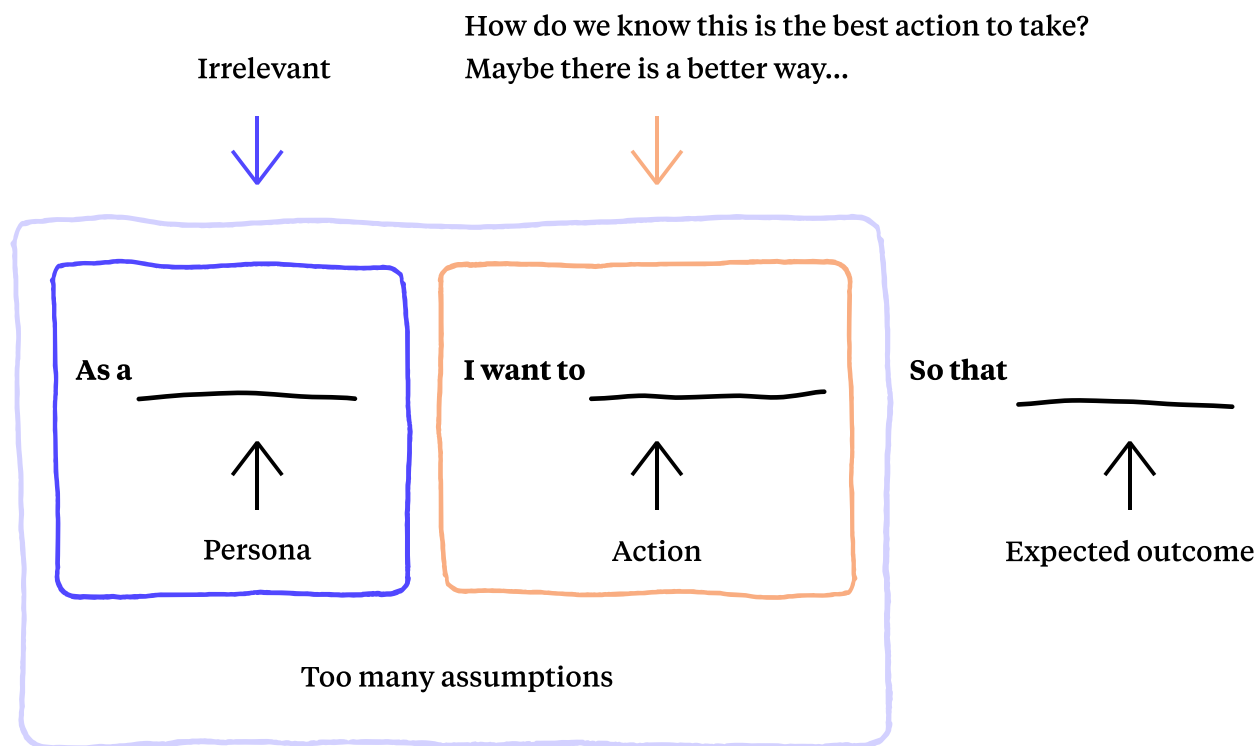
How user stories fail

"As a user, I can indicate folders not to backup so my backup drive isn't filled up with things I don't need saved."

User stories, such as the one above, have three big problems:

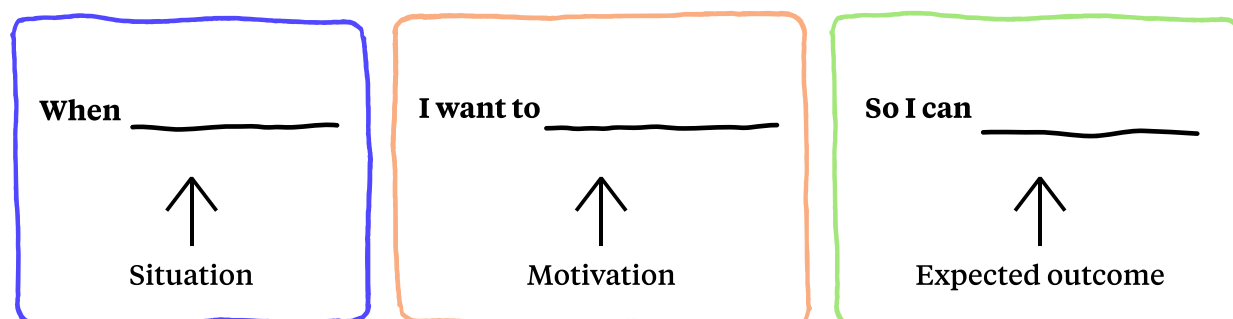
1. They use personas.
2. They couple implementation with motivations and outcomes.
3. They ignore context, situations and anxieties.

Features fail often. If a feature was defined by a user story, discovering why it failed will be difficult, because implementation was coupled with motivations and outcomes. Because of the coupling, how can anyone know what was wrong? Was the implementation wrong, or were the assumptions about the motivations wrong?



Enter the Job Story

First mentioned by Paul Adams on the [Intercom blog](#), and [developed here](#), Job Stories are a different way of designing features. But how does a team implement them into their workflow?

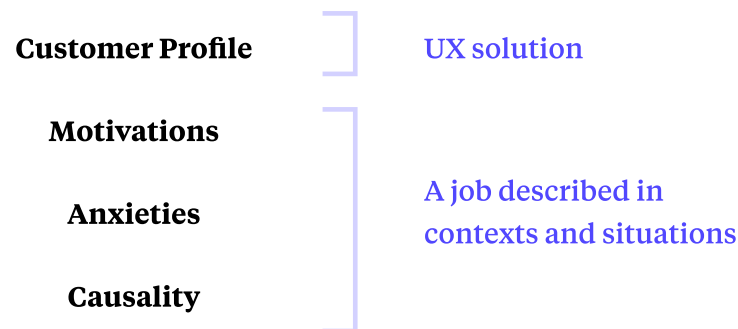


Here's one approach:

1. Start with the high level job.
2. Identify smaller jobs that help resolve the high level job.
3. Observe how people solve the problem now (the job they currently use).
4. Come up with a Job Story to investigate the causality, anxieties and motivations of what they do now.
5. Create a solution which resolves that Job Story.

For example, consider how a team might design a profile view for a product that helps car salespeople secure loans for customers.

Designing a profile view



Early in the design process, the team was discussing what the profile would look like, and what features should exist there.

At one point Sara, a team member, stood up and drew a simple wireframe on the whiteboard. She pointed to a box and said, “This is the salesperson’s profile”.

The team is not immediately convinced by her rationale. They asked a series of “whys” for each part of the profile view. Even after all of these questions, the team didn’t coalesce for or against the idea.

At this point, the question was asked:

“Why should the product have a profile view? Why should it be in one place or another? What information should it display? What situations is it resolving? What job is this profile view doing?”

To resolve this, the feature was reframed through Job Stories.

Start with the high level job

The product’s high level job is to help a car salesperson secure a loan for prospective customers. Currently, the customer and the salesperson have to fill out a lot of difficult paperwork.

Identify a smaller job which helps resolve the high level job

In order to fill the loan application out correctly, the salesperson and customer need to enter

lots of information about the car and the terms of loan. Because the information is sensitive, the customer needs to feel their personal information is safe with the car salesperson.

Observe how people solve the problem now (i.e. what job they currently use)

When filling out this type of information, the buyer analyzes (usually visually) the salesperson and car dealership and deduces if they can be trusted. They generally fill in their sensitive financial information in close physical proximity, on a piece of paper, with the salesperson. This helps them feel confident the information is filled out correctly, and won't get passed around to people who shouldn't be looking at it.

Come up with a Job Story that investigates the causality, anxieties and motivations of what they do now

1. When car salespeople and their customers interact with each other via the product...
2. ...customers want to feel like they can trust the organization, process, and the salesperson.
3. Salespeople are going to want to be confident their process makes their customers feel comfortable...
4. ...so clients will feel safe entering their financial information into a process.

The above frames the situation into a Job Story. It can be fleshed out by providing more situational context – such as when and where it's being filled out (e.g. at home or at the dealership) – and anxieties each party will have about having and viewing a profile.

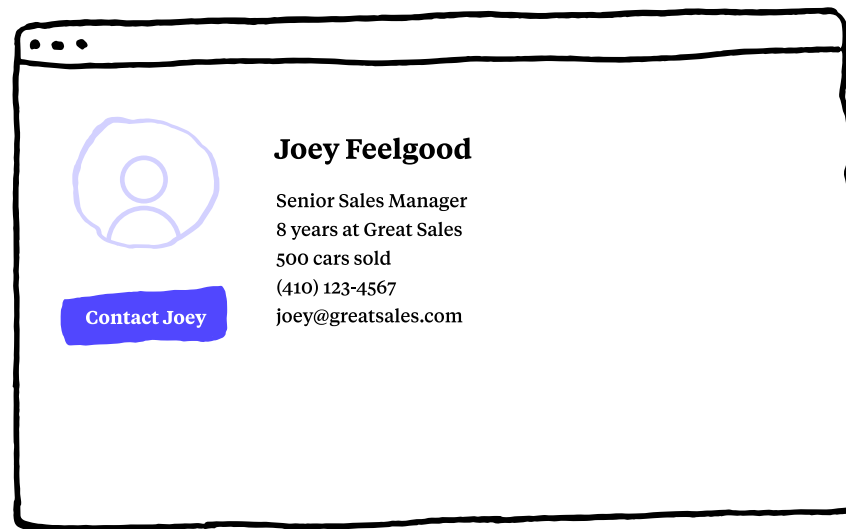
Create a solution which resolves that Job Story

To resolve the job, the team decided how the profile view should be designed. Too little information and the profile view won't solve the original job, and too much information could have negative effects. So the team decided:

1. When the customer uses the product, the salesperson's profile view would always be visible (to ease customers' anxieties about not being physically with the salesperson).
2. The profile would have a picture of the salesperson, job title, cars sold and years at the company (to ease customers' anxieties about whether the salesperson is reputable and can be trusted).
3. The profile view will enable easy ways for the customer to get in touch with the salesperson, such as their phone number, email address and a button saying "Ask [salesperson] a question" (to ease customers' anxieties about filling out the form incorrectly).

Here is an example of a solution:

Your Sales rep!



Here is a breakdown of the UI – the job each UI element is doing, and what situation(s) it's resolving.



We're now left with a product in which every element can be traced back to resolving a job: ensuring the customer feels safe when exposing personal information.

Design for real people, design for causality

Designing successful products means observing how real people solve problems now, exploring the context of the situation they are in, and then understanding causality,

anxieties and motivations.

Abstracted attributes and coupling implementation with motivations and outcomes are distractions for a team. If the team digs deep and learns about a customer's Job-to-be-Done, they craft solutions more effectively. And using Job Stories to design products is one of the best ways to do it.



Chapter 9

Asking why the right way
by Dan Ritzenthaler

Before I discovered Jobs-to-be-Done, finding the “root cause” of the problem I was solving was difficult. So was boiling down a problem into a simple statement. How can I simply articulate this problem to the rest of my team? It turns out understanding the problem across several dimensions was required to really make progress. And asking why is one of the best ways of uncovering deeper needs of what customers are trying to do.

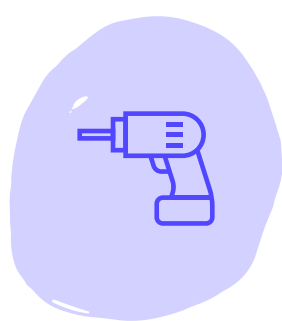
There’s plenty of abstract research advice out there about uncovering causality, such as the “Five Whys”. But the following piece is all about aiming for the middle ground. To move through levels of depth in a conversation, but not too fast to miss potentially valuable insights. The key is to find the unique insight at each “Why?” before moving on.

...

For any declarative statement you might make, I can’t stop myself from asking “Why?” and then, I’ll ask “Why?” again. It’s an occupational hazard.

So you bought an electrical drill? Why? Oh, because you want to hang some framed photos. Why do you want to hang framed photos? Oh, to make your home more personal. Why is making your home feel more personal important to you?

When it comes to product design, there’s such a thing as getting down to the root problem too quickly. The value in figuring out the “Why?” behind things isn’t so much about the destination; it’s about the journey.



Product



Use



Outcome

This is especially true for customer interviews. Slowing down, taking your time with each layer of the “Why?”, exploring lots of different ideas together; it all makes your time together much more productive. And it yields far more interesting and usable insights.

The productive layers of the conversation

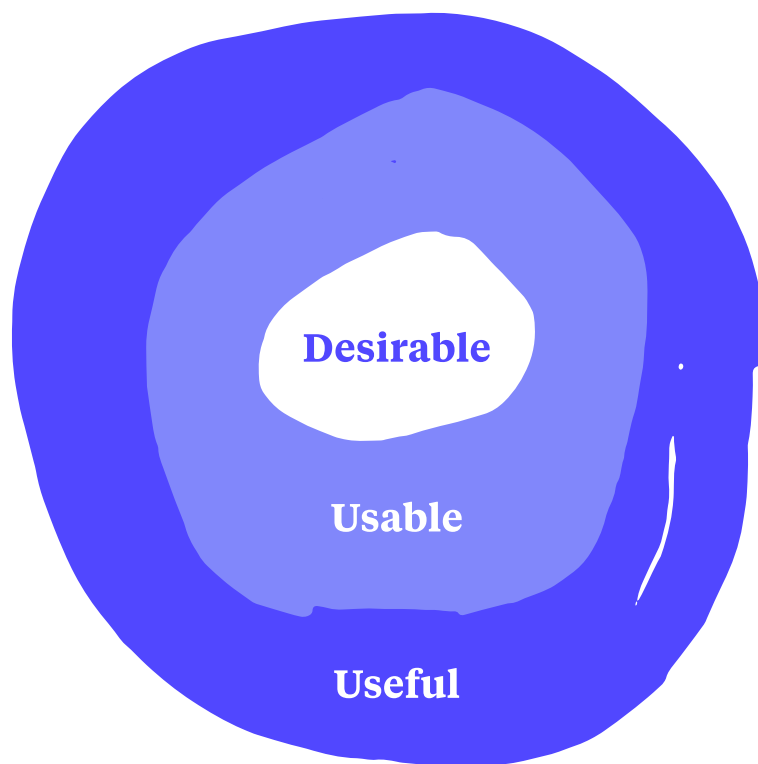
For any given behavior, you can dig as deep as you want. There are as many layers as you have time to discuss, each one relating to a deeper need. A deeper understanding of “Why?”

But when you're exploring the value of a product, there are only so many productive layers to discuss. Here are the layers I've found the most valuable to fully explore:

1. The immediate layer relates to usefulness. What do you actually do with the thing? I use the drill to make holes.
2. The secondary layer relates to usability. What result comes from using it? I've made holes to hang photos.
3. The tertiary layer relates to desirability. What's different now that I've accomplished my goal? I've hung photos and now have a more personal home.

Beyond these three layers, it becomes more difficult to relate things back to the product. For example, asking "Why do you want a more personal home?" is unlikely to lead to meaningful answers for a drill manufacturer.

You may still learn a lot about the person and what drives them, which is always valuable. But for me, it's too abstract and is rarely immediately actionable.



The useful layer

In the electric drill example above, the initial response is the person needs to make holes. You can then dig into all of the possibilities to make the drill more useful.

Where will the holes be drilled? In concrete? Wood? Metal? What size of holes? Tiny holes to set smaller screws? Big holes for larger bolts?

How might that change the nature of the drill? Different power, torque, speed? Maybe speed settings? Different size drills? Changeable drill bits?

These are all great questions to flush out what exactly the drill should do and how to make it more useful. Unfortunately, it doesn't reveal anything about how to make the drill easier to use.

The usable layer

The next layer of using the drill is to help someone hang their framed photos. If the outcome of using the drill is having framed photos on your wall, you can dig into a lot of possibilities of making the drill more usable.

What's the behavior around hanging framed pictures? Are people comfortable using a drill? Are they standing on chairs or other dubious household objects while drilling? Are they nervous about doing it wrong? If we were designing a drill that only drilled holes to hang pictures, how much better could it be?

How would this change the nature of the drill? Does it require smaller batteries since it's not so frequently used? A smaller motor and housing to reduce the weight? More prescriptive settings for typical house walls to reduce anxiety?

These questions reveal what the person is doing with the drill and ways to make it more usable. Unfortunately, it still doesn't reveal anything about how to make the drill more appealing.

The desirable layer

The next layer of using the drill is to make your home more personal. If the outcome of using your drill is you have a more personal home, you can dig into a lot of possibilities for making the drill more desirable.

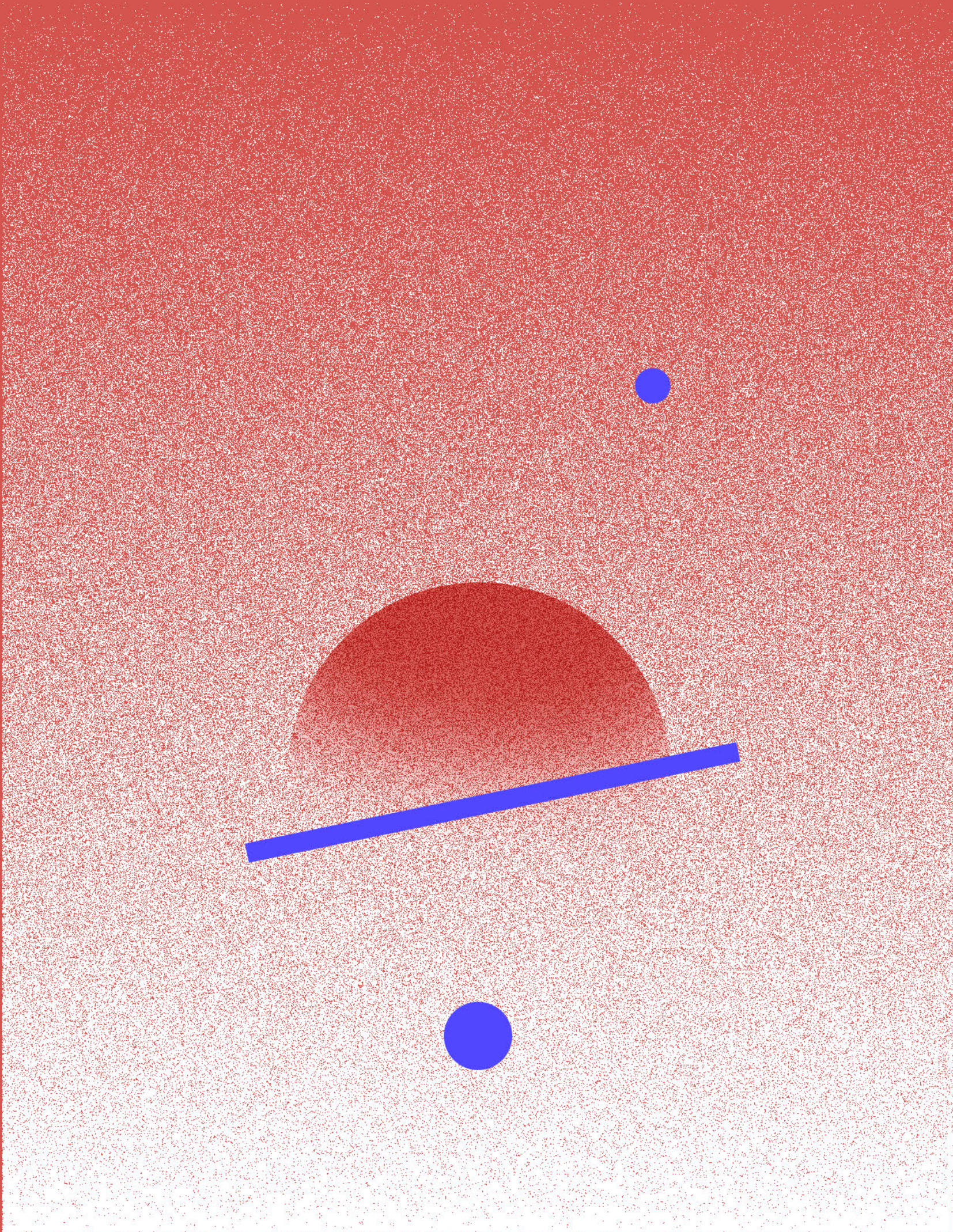
How important is it for the drill to be ready to go when you come home with your next framed picture? Where do you put the drill when you're done? Is it in a place where others might see it?

How does this change the nature of the drill? Does it need a docking station so it's easy to put away? Or should it be plugged into the wall so it's always ready to go? Does it need to look good if it's going to be on a desk or shelf somewhere?

All great questions that can make the drill feel as though it's been designed specifically for you. You can imagine it fitting perfectly into your life and satisfying all of your home personalization needs.

Take your time

When talking about the jobs a particular tool can do for a person, try not to rush to the ultimate, deepest answer. That isn't always where the gold lies. Often, there isn't just one root answer. Even if there is, it may not be all valuable on its own. Explore each layer thoroughly for the unique value it can provide.



Conclusion

At Intercom, we made a huge bet on Jobs-to-be-Done, initially as a way to inform our product strategy, and later as a way to go to market. Five years later, it's still the foundation of our product and marketing strategies. We've architected the entire company around the idea people experience problems in their life or business, and they buy products to solve those problems.

I can't think of one area of our business Jobs-to-be-Done hasn't improved. Product, marketing, sales and support have all benefited heavily from what we've learned by focusing on it. Initially we guided our decisions with our intuition about the various reasons people would buy Intercom, but after a year or so we conducted exhaustive research to learn more about our customers' jobs.

This research can be as simple as talking to new users who've successfully onboarded to your product. If you're wondering how to talk to specific users, [Intercom can help](#). However you decide to collect feedback, remember it's about focusing your company on solving real customer problems.

Our approach to Jobs-to-be-Done is still evolving. This book is only the start – a collection of our best thinking on the theory to date. As our company grows and our customers' needs expand, our thinking will evolve. In five years' time, we'll likely have several books' worth of advice and opinions.

If you've had similar experiences using some of the strategies in the book, I would love to hear from you. I'm des@intercom.com – drop me a note.

Thanks for reading,

Des Traynor,
Co-Founder, Intercom

[Facebook](#) • [Twitter](#) • [LinkedIn](#)

