UX matters

On a mission to help the world build better user experiences by demystifying UX.

Thousands of hours of research packed into **44** case studies.

As seen in:

MORNING BREW

Forbes

FT FINANCIAL TIMES

yahoo! finance

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

-

Delivering with Uber Eats

1 month ago • 18 minutes

#Delivery

-

Signing up to Cash App

2 months ago • 20 minutes

#Payments

-

Playstation vs Xbox

3 months ago • 22 minutes

#Gaming

-

Using Apple Fitness+

5 months ago • 24 minutes

#Apple

-

Selling an NFT on OpenSea

6 months ago • 14 minutes

#NFT

-

Can Gamestop beat Amazon?

7 months ago • 20 minutes

#eCommerce

-

YouTube Shorts vs TikTok

8 months ago • 22 minutes

#YouTube

-

Improving Apple's Airtags

9 months ago • 12 minutes

#Apple

-

Spending money on Twitch

10 months ago • 20 minutes

#Twitch

☕

Subscribe for more UX-caffeine

Free UX newsletter trusted by product-lovers.

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Finding conversations on Clubhouse

12 months ago • 18 minutes

#IOS

-

How Robinhood handled a crisis

Feb 01, 2021 • 13 minutes

#Investing

-

Creating a Signal account

Jan 18, 2021 • 17 minutes

#IOS

-

How Group Sessions on Spotify work

Jan 03, 2021 • 18 minutes

#Music

-

Buying a card on Moonpig

Dec 11, 2020 • 18 minutes

#Purchasing

-

Building a store

Nov 19, 2020 • 21 minutes

#UX of Payments

-

Making a payment

Nov 12, 2020 • 19 minutes

#UX of Payments

-

The card reader

Nov 05, 2020 • 21 minutes

#UX of Payments

-

Opening an account

Oct 29, 2020 • 20 minutes

#UX of Payments

-

How Klarna Works

Sep 17, 2020 • 18 minutes

#Payments

-

How Coinbase works

Aug 19, 2020 • 15 minutes

#Investing

Previous Page 1 of 2 Next

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

Woah. that doesn't exist.

We can’t find that page anywhere, sorry.

#Banking • 18 minute read

Monzo vs Starling: Overdrafts

-
- 4 Feb 22
- •
- 18 mins read

By Peter Ramsey

04 February 22

18 months ago I published an analysis of 12 of the most popular banks in the UK, which demonstrated just how far ahead of the competition Monzo and Starling were.

But whilst their core products are incredibly strong, the cracks start to appear when you get *even deeper* into their experiences.

This month I've been re-evaluating the UX of these two banks specifically, and highlighted a few things that we can all learn from.

So, is the experience of getting an overdraft better with Monzo, or Starling?

Case study

Select a presentation to watch

There are **2** case studies available, you’re currently watching 'Starling'.

There are **2** case studies available, you’re currently watching 'Monzo'.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

Yeah, understood

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 79

Close

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 50

Close

A deeper dive into the UX

> 👇 ***What are these?*** *Below are UX issues mentioned in the presentation, but that I felt were worth discussing in more detail. These are worthwhile conversations to have internally, and consider if they impact your product or service.*

1. Regret theory

Starling asks you how much you want to borrow, before completing any kind of assessment about your eligibility.

Plus, you can only make one request every 3 months, piling on the behavioural psychology and anxiety in a circumstance that's reminiscent of a gameshow.

But without knowing the maximum amount that you can borrow, it's **almost impossible to minimise regret**.

i.e., if you're offered the overdraft, you'll wonder if you could have asked for *more*. If you're rejected, then you'll regret aiming so high.

Even if they happened to stumble exactly on their eligibility limit, and had maximised their overdraft allowance, **they wouldn't know that they had**.

Product makers need to craft experiences in a way that consciously minimises the opportunity for regret.

Regret is a difficult emotion to avoid ***entirely***. For example, some people may regret signing up to your service simply because it wasn't very good.

But here are a few examples of UX features that can help mitigate the risk:

- 💾

  Auto-saving

  i.e., regret typing so much, if it logged you out and didn't save.

- 📃

  Waiting lists

  i.e., regret entering all your details, if you're not eligible to use the service today.

- 📖

  Content previews

  i.e., regret investing your time into content, if it turned out to not be what you were expecting.

- 💸

  Money back guarantees

  i.e., regret making a purchase, if it turned out to not be what you needed.

- ⚠️

  'Are you sure' modals

  i.e., regret clicking on the 'X' in the corner, as you've now lost all your progress.

In this instance, Monzo's execution was more effective, because it ran an eligibility check, and ***then*** displayed the parameters.

You can walk away from Monzo feeling confident that you *maximised* your Overdraft allowance.

2. Selectively hiding content

"We need it to be simple and clean, but also full of context and information."

I'm often asked how to reconcile these two goals, because intuitively they feel like opposing forces. But they don't need to be.

One technique, which Monzo execute brilliantly, is to **selectively hide content behind specific actions**.

When you see it, it looks obvious. But there's a nuance to executing this well: **specific content needs to be grouped, and hidden behind a clear and specific action**.

Specific content. Clear action.

An analogy that may help illustrate this point is to imagine *organising your loft*.

- 🖨️

  1. Individual items

  Over the course of a decade you just continuously place individual items on the floorspace. At first it works well, but over time it becomes a cluttered mess.

- 📦

  2. Boxes

  So you put everything in boxes, which improves the aesthetics, but then finding items becomes laborious. You can't predict what's inside each box.

- 📭

  3. Specific labels

  But by adding clear labels to every box, you make it significantly easier to find items. And the more specific you are, the better.

The point is this: UX works in largely the same way.

The more specific the labels (*on both the loft boxes, and the actions*) the more confident you can be what's inside, and therefore the better you'll feel about opening the box (*or clicking the link*).

3. Specific CTAs

Ponder this: what's the purpose of the label inside a CTA?

For example, if you removed the text '*login*' from the CTA below, would you be unable to guess what it did?

Most of us would agree that even **without** the content *inside* the CTA, we'd still be confident that it was a button to 'log in'. This is possible because of the context surrounding the action.

i.e., it's predictable given the limited scope of its context.

Okay, now what about this one?

This is more difficult, borderline impossible. So I'll repeat my rhetoric again: ***what is the purpose of the content inside a CTA***?

In short, the label provides an explicit description of the action that happens when you interact with it.

As products and services get more complicated (e.g., with additional features), the number of on-screen actions increases. As a result, each CTA typically has *less* context.

In other words; **product complexity creates context-dilution**.

And in a similar vein to the 'boxes in the loft' example, the more specific the label, the more confident the user can be in the action.

The user feeling confident almost always translates into a better experience.

If we compare the screens where you set your overdraft amount:

Starling's is generic, Monzo's is specific—which do you think is clearer?

When done well, a specific CTA label can actually be empowering. The user intuitively knows exactly what will happen next, and that feels very similar to being '*good at something*'.

People feel happy using products that make them feel good, it's often as simple as that.

Summary

Starling is typically world-class, but in this instance, they've dropped the ball. Requesting an overdraft was not a great experience.

But Monzo's is great, and kudos to them. They've crafted a beautiful and delightful experience, once again.

> 🔥 ***Simple UX exercise:***
>
> *Go through your product and consciously try to make your CTAs more specific.*

> *"Send" = "Send Invoice",*
>
> *"Pay" = "Pay £30.00",*
>
> *"Continue" = "Continue to checkout".*

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

More Case Studies

-

Opening 12 bank accounts

May 21, 2020 • 18 minutes

#UX of Banking

-

How categories and tags on Monzo work

Mar 18, 2020 • 6 minutes

#Banking

-

Open Banking

Jun 18, 2020 • 21 minutes

#UX of Banking

-

Signing up to Cash App

2 months ago • 20 minutes

#Payments

#Delivery • 18 minute read

Delivering with Uber Eats

-
- 6 Jan 22
- •
- 18 mins read

By Peter Ramsey

06 January 22

Most of you will be familiar with the experience of either *riding in an Uber*, or having food delivered by one—the convenience is enchanting.

Watching a tiny car bounce around a map as the driver makes their way to you is a UX delight so powerful, that it's been copied across many verticals.

But my experience of the *driver-side app* had been limited to me watching the driver's docked iPhone navigate us to our destination, during a ride. I didn't know what happened before or after.

So, I joined 4 million others, and became an actual Uber Driver.

I then immersed myself in the role; going online at different times of the day, completing deliveries, cancelling orders and hovering outside KFC waiting for my next ride.

Let's get to the point: **the UX is surprisingly awkward**.

Case study

Let's look at the experience of making my first delivery on Uber Eats.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 116

Close

A deeper dive into the UX

> 👇 ***What are these?*** *Below are UX issues mentioned in the presentation, but that I felt were worth discussing in more detail. These are worthwhile conversations to have internally, and consider if they impact your product or service.*

1. "I didn't even want pancakes"

Imagine that you're being served in a restaurant, and you're asked what you'd like to order. As you're reading the menu, you say "*Oh, the pancakes look great, do they...*".

Immediately—before you've even finished your sentence—the chef walks to your table with the pancakes, places them in front of you, and walks off.

It may sound trivial, but let's digest why that would feel **uncomfortable**:

- 🤐

  1. You accidentally made a decision

  i.e., you didn't realise that your actions constituted a decision.

- 😳

  2. You now feel regret

  i.e., did you even want the pancakes after all?

Accidental decisions carry the burden of regret, because they often feel like they were made against your will.

You want your users to feel like ***they*** made a decision, not that a decision was made for them. And in most instances, you can substitute 'decision', for an 'action'.

i.e., clicking on a CTA is both an action, and a decision.

As you saw earlier, **I accidentally accepted my first order**, when faced with this modal:

From a **data perspective**, Uber are probably happy with their conversion rates: a very high number of drivers will accept their first order—but it's a misleading metric.

It's undoubtably an awkward design, and I wonder how many drivers start their Uber careers by accidentally accepting an order, and then just rolling with it.

In short: every action should be intentional, and fake CTAs are almost never a good decision.

2. The knowledge gap

There's another decision-making issue with Uber: **new drivers starting deliveries prematurely**.

In the study, I tapped 'Start delivery' when I was sat outside McDonalds, not when I had collected the order—an error that could probably be eradicated with a clearer design.

Remember: there's essentially no in-app onboarding or education.

Original

Revised #1

Revised #2

This type of error is extremely common, and we call it a **knowledge gap**.

This is where the app is optimised for returning users, but fails to educate new ones.

A powerful UX exercise is to imagine what a human representative would say if they were physically sat next to a new user, whilst they were learning to use the app.

In this instance, I'm convinced that they would say: "*Okay, now go and collect the food, and **then** when you're back in the car, press this button*".

And if you believe that a human sat next to you would feel the need to say something, your app should be screaming it at you.

The **context** that you need to collect the food ***first*** is what was missing here. That may have been predictable after a simple knowledge gap exercise.

3. Split-concentration

Navigation apps (e.g., Google Maps, Waze, native in-car navigation) re-orient your map automatically, when your view is deemed no longer useful.

i.e., you'd zoomed into a particular junction, and now you're half a mile down the road.

The Uber Driver app doesn't, and it renders the navigation fairly useless.

To make this worse, the 're-centre' button was hidden behind a sticky footer.

This is an example of a task with **split-concentration** (i.e., split between driving the car, and using the app).

In a laboratory environment, the Uber Driver app may perform perfectly. But in reality—and when the user has a split-concentration—it fails.

Split-concentration isn't common enough to feature in a UX course, and designers rarely talk about it—but it does happen.

For example, using the Spotify app while running on a treadmill.

In these instances, you need to simplify the UI, and really focus on making things easy to use while completing another task.

Summary

Having used both the Uber Rider, and Uber Driver apps, it feels like they've invested most of their attention into the consumer experience.

Perhaps drivers are more willing to put up with bad UX, as they're earning money and have already invested time into the platform.

But a consumer has more choice—and is *paying* for the service.

So, despite this possibly being a rational trade-off in the short term, it's certainly a lacklustre user experience for the 4 million Uber Drivers worldwide.

> 🔥 ***The biggest UX takeaway from this study:***
>
> *Try the knowledge gap exercise with your team: walk through the process of someone using your product for the first time, and imagine that you're sat next to them as they do it.*
>
> *What do you feel the need to say to them? Which knowledge gaps are you filling with your voice? Most of these will feel intuitive—write them down.*

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Finding conversations on Clubhouse

12 months ago • 18 minutes

#IOS

-

Creating a Signal account

Jan 18, 2021 • 17 minutes

#IOS

-

Mario Kart Tour on IOS

Oct 10, 2019 • 5 minutes

#Gaming

#Payments • 20 minute read

Signing up to Cash App

-
- 2 Dec 21
- •
- 20 mins read

By Peter Ramsey

02 December 21

Cash App is a mobile payment service, which was created by Square Inc. back in 2013. Now it has more than 70 million active users, and boldly claims to be the ***easiest way to send money***.

When I read this, I wanted to demonstrate the nuance that's often missing when labelling something as 'easy'—and Cash App provides the perfect example.

There's a difference between something being ***easy to do***, and something which ***feels intuitive*** (i.e., easy) for a *new* user.

Experienced pilots may consider the controls of a cockpit as 'easy'—after all, most of the buttons are within reach. But for everyone else, it's a complicated mess.

With UX, the devil is not just in the detail, but also in **the context**.

And as you'll see, within the context of *learning* to use Cash App, it's certainly not the easiest option out there.

Summary:

- 🪂

  Aim for a soft landing

- 💸

  How not to give away $100

- 🍕

  Deciding what you'll have for dinner is hard

- 🤝

  Breaking the social contract

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 102

Close

4 key takeaways

1. Softer landings

As a general rule (of, possibly life), you want a *soft landing*. This concept applies to UX, in the form of **context-shifting**.

Context-shifting is the process of switching from one 'thing' to another. The role of UX is to lower the cognitive load of that transition.

In other words, to soften the strain of switching tasks.

As an example of a very ***hard*** landing, opening the Cash App (app) for the first time immediately asks for your phone number.

(i.e., there's no landing page or welcome screen).

One could reasonably argue that it's obvious that you're creating your Cash App account—but that's not the point.

Most context-shifts are *possible*, it's about making them ***softer***. And that's why you, almost universally, see a landing screen when you first open an app.

They look like this, and even with a wireframe you can demonstrate their value:

Fitting with the theme of this case study, it highlights an important distinction in UX: **you cannot reduce clicks indefinitely without eventually hurting the experience**.

As a user, you seldom notice the subtleties that soften your landing—they are often intentionally discrete.

They can be sub-conscious cues to prepare for a context shift, and can be valuable despite adding a click.

2. A free $100

Imagine that you've been enrolled in a social experiment, and have been asked if you'd like to receive $100, or nothing.

In front of you are two unlabelled buttons, and you're asked to make your decision.

This would undoubtedly be stressful, right? It'd feel like a game of chance.

After making your choice, the observer tells you that **regardless of which button you pressed, you were always going to receive the $100**.

In other words, there were two inputs, but only one output.

Now, let's consider the inverse of this: you've been posed the same question, but now there is **only one button**.

This would be similarly stressful, right? Is there a second button somewhere? Is it an option to *not* press the button?

Again, you're told afterwards that your input was irrelevant, you always get the $100.

What this thought experiment demonstrates is that the **negative emotions are not a direct effect of the outcome, but a byproduct of the decision-making process**.

In other words, it matters how a user feels about a decision, not just the outcome of that decision.

The very first screen on Cash App asks you for your phone number or email address. This works for both new and existing users.

In other words, **there's only one button, but two possible outcomes: creating a new account, or logging into an existing one**.

But this kind of approach only works if the user *understands* that they'll be automatically directed to their required route. And they may not assume that they will be, because there's no context at all.

When Cash App look at the data, this flow will convert very well, because there is only one input—so more or less everyone will press it.

But the data **won't** have captured the anxiety of the decision-making process.

3. What's for dinner?

A recent survey asked 2000 couples about their communication, and found that the average couple argues **156 times** each year about what they're having for dinner (and it takes an average of **17 minutes** to resolve).

The point is, many people aren't decisive—particularly when tired and hungry.

This partially explains why default options are so influential to build intuitive products: they provide an easy meal option.

Decisions that require creativity are particularly tiresome, and require the user to disengage their autopilot and *think*.

An example of this is showing the user a blank input field, and asking them to **create their unique Cashtag.**

And this doesn't just require minor creativity, but also *critical thinking*: they need to pick a username that none of the previous 70 million people chose.

There are two suggestions that could really help reduce the friction here:

- ✅

  1. Pre-fill an available Cashtag

  This could be a variant of their full name and/or location.

- 🧠

  2. Suggest alternatives

  i.e., other available variations.

With these changes, Cash App would probably find that people spent less time being frustrated by their own lack of creativity.

The flow would *feel* better.

4. The social contract

Although it's never explicitly mentioned, there's a de facto **social contract** between a product (in this case, Cash App), and it's users.

For example, if Zoom says that they don't record your screen without your permission, then you expect them to not record your screen. The vast majority of the time the social contract remains unbroken, and therefore unnoticed.

And there's no requirement for a social contract to be severe—it can be trivial; such as the promise of a 2-for-1 discount voucher after sign up.

Breaking that social contract may not stop you using the product (i.e., Facebook), but it hurts the user experience at a deep emotional level.

At the end of the Cash App sign-up process, you're asked if you want to provide the application with access to your contacts.

Here, the benefit of sharing your contact's information **is that it'll protect your account and prevent spam**.

Yet immediately after giving permission, you're prompted to refer all your friends for a financial reward.

There's no mention of the benefits that you were promised—nor did I notice that they were ever mentioned again.

1. Promise

2. Action

3. Betrayal

Having seen similar examples unfold at some of the best product teams in the world, I believe this was an honest mistake.

It's typically an oversight from product teams, who's focus is primarily on improving conversions, not the psychological joy of the experience.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

How Klarna Works

Sep 17, 2020 • 18 minutes

#Payments

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

-

Delivering with Uber Eats

1 month ago • 18 minutes

#Delivery

-

Playstation vs Xbox

3 months ago • 22 minutes

#Gaming

#Gaming • 22 minute read

Playstation vs Xbox

-
- 2 Nov 21
- •
- 22 mins read

By Peter Ramsey

02 November 21

There's a bias that I've been silently pondering for a few years: **the user experience of video game consoles is so much worse than general consumer apps, but it's rarely mentioned**.

Both Microsoft and Sony released 'next gen' consoles in 2020, which objectively have interfaces that are considerably worse than most of the software you'll use daily.

My initial rationale was that the complexity of having a _controller_—with buttons and joysticks, instead of a cursor—made it more challenging.

I've spent a few months with both the Xbox Series X and the Playstation 5, agonising over the small details. And whilst I still think that the controller is a limiting factor, it's not the reason for this bias.

And importantly, you **don't** need to be passionate about either console to (hopefully) enjoy this—it's designed in a way to have clear UX takeaways that are relevant to general software design.

Let's jump right in:

Benchmarking

Firstly, let me show you a good exercise to benchmark one product against another (*useful for competitive research*).

On the surface it looks too simple to be valuable, but having used this methodology for years—and over many industries—I can vouch for its effectiveness.

I've created a basket of 30 'common' tasks, split over 3 categories, and then recorded a series of datapoints during the completion of those tasks.

You can view them all here, but here are three examples:

- 🏆

  Viewing achievements

  Checking the achievements/trophies for a game, while you're playing it.

- 💬

  Starting a party chat

  Inviting two friends to a party chat.

- 💸

  Purchasing a game

  Searching for and purchasing a specific game in the digital store.

Albeit a crude instrument, it allows us to simulate and benchmark usage.

To start with, let's look at how many actions (button presses) it takes to complete our basket of tasks:

Total number of actions

On average, the PS5 required 29.8% more input than the Xbox to complete a comparative task.

There's a data-mine here, but it's important to also know the *distribution* of clicks and tasks.

As you can see below, the Playstation wasn't just slightly more awkward at *some* tasks, but required more effort for ***most*** tasks.

A notable example of this is the Control Centre, which defaults your focus onto the activity cards—forcing the user to hop ***down***, then along until they reach the menu item that they want.

But before we prematurely conclude that that Xbox has a better interface, there's another metric to lean on: **number of significant screens**.

As an example, when purchasing a game through the Xbox store, you'll see the following 3 consecutive pages:

1. Home

2. Store

3. Game

Experienced users will fly through these screens because of muscle memory. But **new users will need to digest each screen and all of the content in it**.

This mostly happens at a sub-conscious level, but it adds to their cognitive load—i.e., the more options you can see, the more you need to understand, and the more effort that requires.

When the majority of the content (or actions) in view changes, I call that a 'significant screen'.

If you look again at the Xbox example above, it's not *just the buttons* that have changed, but the layout, the colours, the fonts and even **the context**.

Understanding what you're looking at *requires processing*, and so you also need to consider the number of *significant screens* for each task.

Total number of significant screens

88

105

Interestingly it's the inverse of the 'actions' benchmark. Whilst most tasks do require more actions on the Playstation, the interface is actually more consistent and has fewer significant changes.

This means that on average, completing tasks on the Xbox require 19% more 'significant screen changes'.

If we look at the distribution again, we can see that there are no major outliers—something which is always worth double-checking.

But there's a nuance to consider when using the 'significant screens' benchmark: ***how much information* is the user having to process in these screens**?

i.e., as the number of possible actions (or routes) on each screen increases, so does the amount of effort that it takes to understand.

To demonstrate this, here's an abstract version of the Playstation dashboard, next to a modified version with 50% more buttons (by number, not volume).

Original

50% more

Original

50% more

You can see that as the number of actions increases, things _feel busier_—this is intuitive.

The Xbox has a greater number of significant screens, but that's far less relevant if it's not actually showing **more information**.

To test this, I've recorded every interactive element that is in view whilst completing these tasks. i.e., all the *possible* routes that the user could take, and therefore a better approximation of the **amount of cognitive load from each screen**.

Possible actions in view

885

896

After marvelling at how negligable the difference is (about 1%), we can draw a hypothesis.

Firstly, we can dismiss the notion that the Xbox requires less input *because it packs more options into view at once*. If that were true, then there would be far more 'possible actions' visible while completing tasks.

Instead, the Xbox requires less input because its menus are more efficient. It's configured in a way that makes every day tasks easier to do.

Or at least, easier than the Playstation 5.

Here's the kicker: they're both bad

The natural conclusion here would be to pick a winner. But that'd feel unfairly divisive, given that my opinion is that **both consoles have terrible user experiences**.

If you wrote a list of all the software that you use on a daily basis (*Google Docs, Mac Mail, IOS, Twitter, TikTok*...), they'd probably all be better than either console—**and considerably so**.

Both the Playstation and Xbox lack empathetic design or any real onboarding, and they make hundreds of obvious UX mistakes.

For example:

- **Not auto-focusing on fields** when there's only one field / action on the screen.
- **Not auto-proceeding** between the month and day fields in a date-picker.
- **Empty states without actions** (e.g_., "no WiFi connected" without a link to connect the WiFi_).
- **Poorly-labelled settings** (e.g., the difference between Playstation's 'Avatars' and 'Profile pictures').
- **Basic functionality requires customer service** (e.g., cancelling a pre-order on Playstation).

There's limited value in just listing 100 UX mistakes though.

Really, you don't just want to know what the mistakes were, but also how you can learn from them, and ultimately improve your product (*whatever industry that is in*).

So let's pick a few to go into more detail with.

4 in-depth examples of poor UX

1. Trophies vs Achievements

Both consoles have a system to track in-game milestones, which pop up during gameplay when unlocked.

These serve as a **psychological reward** for completing a task which otherwise may feel empty—for example, finding every collectable item.

But, you want the user to **experience the endorphins** (the reward), **without detracting** from the gameplay.

Playstation show a banner like this for 6 seconds:

On first glance this is *fine*.

But when you consider that you won't *always be working towards a specific trophy*, it's facepalm-inducing—**how do you know what this reward is for**?

The value of a reward is much greater if you know what it is, or why you're getting it. It enhances the experience.

Xbox leverages this psychological bias to much greater effect, by having a 10-second notification, **which carousels through key information**.

Importantly, this includes a description of the achievement, which tells you ***why/how*** you've earned it.

Sure, on both consoles you could press a button to see a summary of that achievement, but despite obviously detracting from your mid-game experience, that's missing the point.

The Playstation trophy notification is more akin to a loot box—you know you have one, but then you need to *take action to see what it is*.

2. Assumed technical knowledge

It's already well-documented that upgrading the internal storage of the Playstation is awkward—you need to literally unscrew the unit, which unless you're familiar with PC-builds, you've probably never done before.

But, rarely is the software mentioned, which arguably is just as bad.

Assuming that you're not familiar with taking apart PCs and rebuilding them, it's likely that the following will be true during this process:

- 🥶

  Nervous about failure

  i.e., you don't want to break your console.

- 😳

  Feeling uncomfortable

i.e., you're out your comfort zone.

- 😓

  Anticipating regret

  i.e., once you're halfway through, you may start anticipating that it's going to go wrong, and regret ever starting.

In short: your emotions are heightened, you're feeling vulnerable and out of your depth.

Now imagine that you've screwed everything back together, nervously plugged the console back in, and turned it on. Your hands are clammy, and you see a series of screens like this:

It's what I'd describe as a '**dev screen**'—all the information required, but no empathy or human touch.

Instead of calming the aforementioned negative psychology, it exacerbates it.

- 👎

  1. Implied doubt

  "If you experience problems..."

- 🧠

  2. Technical specification

  e.g., "M.2 SSD", "read speed", "5547.403 MB/s"

Besides, showing the read speed is only useful if the user actually knows what the read speed should be.

The most important thing is that these screens are reassuring and comforting.

As an example, one small change would be to add a line of text clarifying if the user's SSD is above the recommended read speed.

3. Accidental cognitive dissonance

During the initial set-up (for both consoles), you're asked which power mode you'd like the console to be in.

Here's the Xbox's:

This is likely to be an **impulsive decision**, given that without knowing the details—_i.e., how much energy it saves, or how much faster the start-up is_—it's difficult to make a rational one.

Which is actually okay, because it can be changed later, and doesn't affect your initial gaming experience at all.

But, a few steps after this, you're asked if you want to turn on remote features...

It's worth noting the subtle **context-shift** here. Previously, you were being asked to think about ***power***, and in the back of your mind you were considering the environmental implications (*hopefully*).

But now the context is on features (energy isn't even mentioned)—and who doesn't say yes to unlocking more features?

There are a few biases at play here, but one which you rarely see in software is **cognitive dissonance**: a form of psychological discomfort arising from holding two conflicting opinions.

i.e., once you select to *turn on remote features*, you need to decide if you were right *before*, or if you're right *now*.

- 💡

  I want to be energy efficient

  My previous decision.

- 🏎️

  I want those features

  My current decision.

This discomfort is only slight, and is otherwise a merely hiccup in the set-up process—but it still exists, and shouldn't happen.

The subtle mistake that Xbox made was not asking the user to **make a decision once they had all the information**. It makes no sense to ask it in two stages like this.

4. Efficient but unhelpful menus

Sometimes you'll have a friend round who wants to play a game, but you don't want them to make any progress on your account. Plus, many games don't let you have multiple saves in the same account (e.g., Call of Duty: Warzone).

Guest accounts exist to provide a quick way for you to load a temporary profile, without going through the whole 'create an account' process.

On the Xbox, there's a single button to add a guest:

This may be fine for experienced players. But it literally makes no attempt to teach new users how it works.

In contrast, Playstation have done a considerably better job.

To state the obvious: Playstation give you the *context* that you need to make a decision (and, to learn how it works).

'Quick menus' are often efficient, but for casual users—or infrequently used features—they can be less effective.

Largely I think that this a trend throughout the Xbox UI.

Things are built for efficiency (as demonstrated in the benchmarking), but the execution often was very poor from an educational perspective.

Summary

To finish, let me summarise this case study:

- For some reason, the UX of gaming is lagging behind general consumer apps.
- The interfaces of both the Playstation and the Xbox could be considerably better.
- This probably hasn't changed your mind about which to buy—which is fine, but hopefully it's given you some food-for-thought next time you play.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Spending money on Twitch

10 months ago • 20 minutes

#Twitch

-

Mario Kart Tour on IOS

Oct 10, 2019 • 5 minutes

#Gaming

-

Can Gamestop beat Amazon?

7 months ago • 20 minutes

#eCommerce

#Apple • 24 minute read

Using Apple Fitness+

-
- 29 Sep 21
- •
- 24 mins read

By Peter Ramsey

29 September 21

Most people find running on a treadmill boring—after all, how long can you really stare at your garage wall before losing motivation?

In an attempt to leverage some of the social psychology behind a group exercise class, Peloton released a premium spin bike for your home, with a giant screen mounted on the front. This lets you join live video classes, and cycle along with others in a pretty seamless experience.

But there's a catch: the equipment isn't cheap. The cheapest bike and treadmill being £1,390 and £2,295, plus a £39 monthly subscription for the live classes.

What if you could get the benefits of a that group psychology, without having to spend thousands of pounds?

Instead of buying a Peloton treadmill with an integrated screen for £2,295, couldn't you just use your iPhone, and a £400 treadmill that you picked up from Amazon?

That's more or less what Apple Fitness+ is—a subscription for home-workouts, that aims to ***utilise Apple hardware with standard fitness equipment***.

But is the user experience any good? That's what this case study is about.

Summary:

- 🪃

  Beware of the 'jump in' fallacy

- 🚧

  Why Apple intentionally adds friction

- 🧙

  Some UX wizardry by YouTube

- 😇

  Why to not constantly push for upgrades

- 😍

  How Apple strikes a perfect balance

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 114

Close

5 key takeaways

1. The 'jump in' fallacy

It's wrong to assume that showing 'options' quickly is ***always*** good UX—this is often under the pretence that it's allowing the user to 'jump right in'.

There's a nuance that's seldom considered: can the user actually engage with all of this content, or not?

i.e., ***if they wanted to***, could they actually use all of these options (or results)?

For example, on YouTube, you can watch any video you like—sure, you may not want to, but you can.

But on Apple Fitness, you can't. You may not have the right equipment, or the physical space to do the workout. The value of a cycling class to you is null, unless you have a bike.

This means that whilst you ***technically*** can 'jump right in', and click on any workout, you'll find that an ineffective means of browsing content.

Instead, you'll need to—probably sub-consciously—go through a process of working out which options are appropriate for you. This is made more difficult because the options on Apple Fitness aren't labelled in any meaningful way.

i.e., do I need weights for a 'hiit' workout? What about a 'Core with Kyle' workout?

Remember: if the user cannot get value from every option shown, help them to quickly work out which ones they should invest time in.

2. Intentional friction

Making a purchase is typically a moment of high intensity. The anticipation for using the product is growing by the second, and you're excited about the new benefits you're about to have.

You know what is guaranteed to kill that momentum? Asking for access to your data.

1. Select package

2. Pay

3. Give us data

Providing access to your data is a selfless act—the direct benefit is to the company, not you. So why would Apple do this immediately after sign-up? Are they unaware that it damages every user's experience?

The answer is probably that they are aware, but have made a calculated trade-off: ***the data about your first-time usage is more important than the negative impact on the user experience***.

This is intentional friction.

This begs the question: when is it okay to add friction like this?

A good rule of thumb that I follow is to only add friction, if there is a **trackable and provable long-term benefit**.

As an example, 'collecting data' is only valuable if you actually do something with it. So the question is, what ***do*** you do with it? ***How*** does it impact the long-term user experience?

And ultimately, can you ***prove*** that benefit (if not, how will you ever know if it was a successful trade-off)?

3. YouTube mastery

We've all been in this position: you've clicked on a video, the first 10 seconds are boring, so you skip forward a bit to try and bypass the introduction.

Except, it didn't work. Now you don't know what's going on, so you rewind a bit—but still, you lack the context to efficiently listen-in to the conversation.

This is how many people, myself included, used to watch YouTube videos.

YouTube have solved this issue with a feature called 'Chapters'—which works like chapters in a book, or in a film.

You may have noticed them: they're rectangular bars in the footer of the player, which have been labelled by the uploader. This is an important innovation for YouTube, because their retention rates are based on creators uploading engaging content, which is out of their control.

And, as more YouTubers fill the first 60 seconds of their videos with ads, the ability to efficiently skip portions of the video becomes more important.

> *Note: even DVDs had menus and chapters for this reason—without them, you'd need to scrub the entire film to rewatch a specific scene. New technology = new UX problems.*

It's likely that this is reducing churn, and at a conceptual level, is something that Apple should be looking to introduce to the videos inside Fitness+.

As an example, I was 45 seconds into of the workout introduction videos, and all I'd learned is that you shouldn't stand on a moving treadmill.

So whilst the production value is high, they're long, slow and boring. Most people won't have the patience to watch them.

4. Pressure to upgrade

Every time you start a workout it automatically tries to connect to an Apple Watch, and makes you take an action to ***not*** use one.

That's key, because it's making you ***do something*** every time.

This is likely to have two psychological effects:

- 👆

  Increased motivation to purchase an Apple Watch

  i.e., to maximise the value from the Fitness+ subscription.

- 👇

  Lower perception of value

  i.e., a constant reminder that you're not getting the most value from the subscription.

This trade-off is really common, especially with SaaS products. How do you promote an upgrade, without making the user feel like what they have *now* is insufficient

It's essentially the fear of missing out—which as you probably know, isn't a solely positive emotion.

It's also another example of **intentional friction**: a finely tuned A/B test of how much pressure they can put on you to upgrade, without you getting annoyed enough that you stop using the app.

When building products, you can't keep amping up the pressure to upgrade, without eventually risking a net negative effect.

5. The perfect balance

Unlike Peloton, classes on Apple Fitness aren't actually live—they're pre-recorded. This makes it inherently less interesting, as it's literally a video of someone else jogging, without any scope for interactivity.

So, when designing the experience of actually doing a workout, Apple had a fairly nuanced challenge; **how to make the UI engaging, without being distracting**.

This is even more difficult when you consider that the primary goal of the video *isn't* to capture your full attention—like a YouTube video would be—but to guide you through a workout.

i.e., if the video sought your full attention, you'd not be able to focus on the running (e.g., breathing, pacing, stride length).

In my opinion, Apple have found an impressive balance, and the core experience is fantastic.

But it's unlikely that you're also building a live fitness app, so what can you learn from Apple's execution?

Well, it's a masterclass in ***timing***. The UI changes frequently enough that it stays interesting, without ever being overwhelming. Elements collapse when they're not needed, and expand when they are.

Everything on screen has a purpose, at all times.

And that mantra can be applied to almost every dashboard, SaaS product or onboarding flow—how can you effectively show the user what they need, when they need it, and then hide it when they don't?

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Finding conversations on Clubhouse

12 months ago • 18 minutes

#IOS

-

Creating a Signal account

Jan 18, 2021 • 17 minutes

#IOS

-

Mario Kart Tour on IOS

Oct 10, 2019 • 5 minutes

#Gaming

#NFT • 14 minute read

Selling an NFT on OpenSea

-
- 19 Aug 21
- •
- 14 mins read

By Peter Ramsey

19 August 21

NFTs (non-fungible tokens) are unique digital assets—like an image, or a song—that are stored on a blockchain.

And like traditional art, some NFTs are worthless, while others are hyper-valuable. In February 2021, an NFT from the artist Beeple sold for more than $69 million.

Every day, art that you probably wouldn't even hang on your wall is selling in a JPEG format for thousands of dollars. It's a total NFT-feeding-frenzy.

So people have flocked to **OpenSea**: the biggest platform for buying and selling NFTs. But is the user experience of actually creating and selling one any good?

To test that, I've created my own collection, and the short answer is: *it could be much better.*

Summary:

- 🚗

  1. Why you need to use 'Actionable routes'

- 📦

  2. The trap of sub-tasks

- ✅

  3. Don't forget to auto-save

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

-
-
-
-
-
-
-
-
-
-
- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 92

Close

3 key takeaways

1. Actionable routes

You may have heard of the concept under another name, but an **actionable route** is essentially a *preferable next step* at any given time.

For example, after finishing an episode of something on Netflix, one actionable route would be to watch the next episode, another may be to find a different series to watch.

As a designer, you only really care about the actionable routes that are common and **commonly preferable**. i.e., the user may want to rewatch the same episode at 2x speed, but it's unlikely enough that you wouldn't prioritise it.

It's evident that OpenSea haven't spent much time thinking about these. As an example, look at this:

At this point, the user has clicked into the 'collections' field, whilst adding a new NFT. I imagine that these would be the likely actionable routes:

- 🌱

  Add a new collection

- 🧠

  Learn more about collections

- 🤷‍♀️

  Do nothing

And so the UX for that empty state really writes itself: you help the users get to their actionable routes as **efficiently as possible**.

2. Sub-tasks

Studies have shown that multi-tasking increases stress and decreases productivity. Despite the evidence, people still think that they can do it effectively.

One possible reason for this may be the issue of *defining* 'multi-tasking'.

To clarify, because you only have one cursor on a computer it's difficult to literally do two tasks at once. But, the term 'task-switching' and 'multi-tasking' are interchangeable here.

Or rather: the process of **switching between tasks** carries a penalty. It's *mental multi-tasking*.

As an example:

- ⛳

  1. Creating an NFT

  The user has a clear task: to create an NFT.

- ☝️

  2. Creates a collection

  The user has to switch tasks, to creating a collection.

- ⛳

  3. Back to creating an NFT

  They need to re-orient their focus back to this.

My point isn't that you should always refrain from any type of task-switching—it's often unavoidable.

But rather, that bringing the user back into the original task is more challenging, and what OpenSea have done here is actually make it harder for the user.

Here's the nuance: it's easier to engage with a new sub-task, than it is to refocus on your original task.

OpenSea should be bringing the user **immediately back to their original task**, with a **context buffer** like this:

This would confirm what has just happened, and provide transitionary context—i.e., *that sub-task is now complete, and you can get back to creating your NFT*.

OpenSea would probably find that this approach makes the process of creating an NFT and a collection ***feel*** more intuitive.

3. Auto-saving

A major factor in managing the sub-task transition is to ensure that the user hasn't lost any work. You never want them to have to re-enter information.

This can often be solved by **auto-saving**.

There's another common scenario where auto-saving is immensely valuable: encountering flow-breaking errors.

For example, during the collection-creation process I kept running into this bug where clicking the main CTA would give the impression that it's doing something, and then just spring back into it's default state.

Click.

Wait.

Click again?

You can't avoid every error. It's not even a realistic expectation to build helpful error messages for ***every*** potential error. Sometimes things just break.

But, auto-saving can be used like a UX-airbag.

And it would have prevented a lot of frustration here, because I had to refresh the page on OpenSea, and enter everything again.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

How Coinbase works

Aug 19, 2020 • 15 minutes

#Investing

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

-

Delivering with Uber Eats

1 month ago • 18 minutes

#Delivery

-

Signing up to Cash App

2 months ago • 20 minutes

#Payments

#eCommerce • 20 minute read

Can Gamestop beat Amazon?

-
- 22 Jul 21
- •
- 20 mins read

By Peter Ramsey

22 July 21

Gamestop—the previously dominant brick and mortar store—is going through a transformation. With new leadership, no company debt and a war chest of more than a billion dollars in cash, their focus is now largely on technology and eCommerce.

Mainstream stock market pundits would have you believe that Gamestop is *already dead*, and that their only sign of life has been gifted by Reddit and retail investors, but that's not true. They're alive and kicking.

There is a pretty fundamental question to be answered though: can Gamestop beat Amazon?

You may dislike Amazon for a number of reasons, but it'd be hard to argue that the ***experience*** as a customer is anything other than incredible. In many cases, they've reduced the purchasing process down to a single click.

This UX analysis may be about Gamestop, but more generally it covers common pitfalls of eCommerce. How do you make shopping online *feel* intuitive?

Summary:

- 🍪

  How not to do cookie notices

- 🖼️

  The importance of image resolutions

- 🧯

  Why users need to understand the 'engine'

- 🌪️

  The annoyance of loading

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 95

Close

4 key takeaways

1. Cookie notices

Imagine if every time you walked in a physical store, someone immediately approached you, asked you to read a 1000-word legal document and to sign your name on a clipboard. Only ***after*** doing that, were you allowed to see what items they sold.

Not only would you ***not*** actually read the legal text, but it'd also be incredibly frustrating.

This is essentially what happens with cookie notices on the internet right now.

But despite all cookie notices being annoying, some are more annoying than others.

There are two main rules to follow here:

- 👀

  Be concise and load everything above the fold

  i.e., the user should never have to scroll.

- ☝️

  Make it a single-click action

  i.e., one click to accept and dismiss, no scrolling.

Although there are many variants of cookie notice, as long as they follow these two rules, the negative impact is minimised.

But Gamestop don't do this—they have a triple-action process. You need to scroll, click and then click again.

1. Scroll

2. Click

3. Click

It's unnecessary, frustrating and dampens the immediate experience of using their website. This would absolutely be a 'quick win' for them to improve.

2. Image rendering

There's a compromise to understand about high-resolution images: the larger the file size, the longer it'll take to load—which then has many knock-on effects to search engine results, conversion rates and bandwidth.

Most people don't realise how significant this issue is, because the platforms they use **automatically** compress their files.

For example, I uploaded an image to Twitter, then re-downloaded it, and it had silently been **compressed to 26%** of it's original size.

And whilst the computation that goes into compression is complicated, from a UX perspective one thing is clear: the images should be in focus.

Frustratingly, Gamestop's website is littered with poor imagery. This includes inconsistent image resolutions, often a lack of retina support and images displaying at incorrect scaling ratios.

The end result, is that most of the images looks blurry.

What I often see in large companies, is that the website *content* is handled by a marketing department—who've not been trained in image compression, and probably don't even know the native size that each image will be displayed at.

When that happens, a key component of the user experience has fallen outside the scope of the designers.

What looked pixel-perfect in Figma, derails into an accidental smorgasbord of image resolutions.

> ***UX tip:*** *This is why image templates and guidelines are so critical for content teams—a set of rules that find a good balance between page speed and image quality.*

3. Predictability

To build an intuitive eCommerce experience, customers need to be able to reliably (and predictably) find products.

Or rather: **is it obvious how to *get to* a specific product when I need it**.

This is just one of the reasons why using Amazon *feels* so easy, despite not having the most aesthetically imaginative design.

The nuance here is that the intuitiveness of Amazon is a byproduct of how predictable their website is.

Gamestop **needs** this predictability. They need people to understand how the product architecture of their site works, and be efficient at finding the items that they want to buy.

But currently, it's not. For example, when making a search for '***PS5 console***':

There's no doubt that finding products on Gamestop is more awkward than it needs to be.

The category tag system on mobile works poorly too. You can only ever seen the next immediate category options, and not the sub-categories within them.

This is a '*must-fix*' for Gamestop.

> ***For example:*** *if you search for 'PS5 console', you'd need to select the 'Video Games' category, wait while the page loads, then you'll see the 'Playstation 5' sub-category. If you click and wait again, you'll see super-sub-categories.*

4. Movement while loading

When a webpage loads, all of items—like images and widgets—need to be loaded individually. It doesn't all happen at the same time.

Often, this causes elements to **bounce around** the page as they, and everything else around them, load.

For example, try to keep track of the 'Pre-owned selector' in the GIF below:

FREE shipping on orders over $35

GameStop

Search games, consoles and ...

Video Games / Nintendo Switch / Games

Bestseller

Trade Value

ESRB Rating: E (Everyone)

Product Description

Escape to a deserted island and create your own paradise

This is terrible UX for a few reasons, mainly:

- 🤷

  Uncertainty on the final state

  i.e., it's not obvious when it's finished loading.

- 🎯

  Inaccuracy of interaction

  i.e., you may think that it's loaded, click something, just for that button to be replaced by another.

- 🌪️

  Disorienting

  i.e., you'll keep readjusting to key elements as they bounce around.

This is an example of when back-end development and UX meet—**everything should load in its correct place**, either by having fixed containers, or lazy load placeholder content.

During development, often the gap between '*Figma design*' and '*real product*' is left to the discretion of the back-end developers.

In other words, nobody has specified what should happen ***while*** everything is loading.

Conclusion

Is the eCommerce experience of Gamestop as good as Amazon? **No, right now it's not even close**.

Amazon is far more intuitive than people give them credit for, and the current Gamestop website is fairly poor by comparison.

But Gamestop are in the middle of a rapid transformation, so that could change any day.

People's expectations about how an experience should *feel* have grown significantly in the last 5 years.

Gamestop will need to invest in their UX in the same way that they have been with their distribution centres, technology and customer support.

> ***Full disclosure:*** *I own shares of Gamestop—have done for months—and plan on holding them for the long term.*

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Spending money on Twitch

10 months ago • 20 minutes

#Twitch

-

How Robinhood handled a crisis

Feb 01, 2021 • 13 minutes

#Investing

-

Mario Kart Tour on IOS

Oct 10, 2019 • 5 minutes

#Gaming

#YouTube • 22 minute read

YouTube Shorts vs TikTok

-
- 24 Jun 21
- •
- 22 mins read

By Peter Ramsey

24 June 21

YouTube has a history of releasing features which sound promising, but after a lack of sustained engagement, are either killed off, or just fade into irrelevancy.

YouTube ***Red***, YouTube ***TV***, YouTube ***Premiere***, YouTube ***VR***, to name a few.

But after seeing the explosive growth of TikTok, YouTube reacted, launching an almost identical concept under the title of **YouTube Shorts**.

But to convince creators to invest time into this new format, YouTube will need to generate serious engagement.

So is the user experience of YouTube Shorts as good as TikTok? Is it intuitive enough to convince people to use it?

Summary

- 🧠

  The importance of selective attention

- 🤣

  How YouTube is pranking their users

- ⚡

  The power of a live search

- 💡

  How to avoid frustration with a simple message

- 🍩

  Why you should always close reward loops

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 104

Close

5 key takeaways

1. Selective attention

Like computers, the human brain can only process a limited amount of information at once. So in a world *full* of stimuli, how do you know what's important?

Psychologists call it selective attention—the automatic and subconscious method of selecting what to pay attention to.

Selective attention is necessary for survival. It's what helps us focus on traffic while crossing the road, or to listen out for the rustle of a predator while foraging.

And that's where UX design comes into play: **to help your automated processes focus on what's really important**.

Or rather, when designing products, you need to be aware of where a user's attention will be—and often more importantly, where it isn't.

For example, below are two screenshots, one before adding music into your video, and one after.

Notice how the 'Adjust' tool appears in the right menu after adding music. This is how you change the part of the song that's playing.

But many people won't notice this new tool, because they're not looking. They'll have assumed that nothing has changed.

This is for a few reasons:

- 👬

  The pages generally look similar

  It's not immediately noticeable that anything has changed.

- 🤷‍♀️

  They're not expecting a change

  They've already seen that page, and there's no reason why you'd expect there to be a new tool.

- 👀

  The change is subtle

  The specific change is very subtle and blends in with the existing design.

In this situation, YouTube would likely increase the use of this tool by providing the user with a prompt. Something to trigger the automatic part of their brain into thinking "*wait,* ***something has changed, we should pay attention***".

2. A sticker of a button

A few years ago, a prank where people stuck *images* of power outlets onto walls in public spaces went viral.

People would see them from a distance, and *without thinking*, attempt to plug in a charger.

Someone would be secretly filming, and catch the disappointment and confusion wash over the victim's face.

This worked so effectively not because they looked so realistic, but because people **weren't paying attention**.

i.e., people see what *looks* like a plug socket, and they just assume it is a real one.

> ***Note:*** *this is similar to the selective attention bias above, but the point I'm making is slightly different. I could have used this example to demonstrate both concepts.*

YouTube are accidentally playing a similar prank on their users, because once you've added music to a video, the button goes slightly more transparent and inactive.

It essentially turns into a sticker of a button. You even get this circle appear where you've tapped—which acts as a focus for the camera.

Instead, the button should **still exist**, but tapping it should ***do something*** that explains the context behind why it's inactive.

Without this, how would you know the difference between an intentionally disabled button, and a broken one?

3. Live search

When implementing search functionality, there are two main approaches:

- ☝️

  Static search

  When you need to press a button to initiate a search.

- ⚡

  Live search

  When results start appearing while you're typing the query.

And despite the occasional infrastructural benefits of a static search, the **user experience of a live search** tends to be **considerably better**.

One reason for this is that with a static search, the user has to make two decisions: *what* to search for, and *when* to click search.

> ***Note:*** *An example of where a static search would be better is if the process of searching literally _slows the user down—_i.e., if hitting the server with loads of queries has a noticeable impact on the front end speed.*

Let's look at the experience of adding music to a Shorts video, as an example:

Imagine that you had a song stuck in your head, but couldn't remember the exact title.

"Oh, it's 'rolling in the dark', or 'riding in the dark', or maybe 'riding in the deep'".

It's possible that none of those searches will show the result that they wanted; 'Rolling in the Deep', by Adele.

But if this were a live search, they'd have a higher chance of finding the correct result, and do so with much less effort.

Plus, in the instances where a live search is more **efficient** at showing users their desired results, it'll also be more enjoyable to use, and likely see **higher engagement**.

4. Deleted without warning

Imagine that you've just recorded a **rare moment** on camera—say, a baby's first steps—and you want to record a second clip of your reactions.

You go to press the record button, but in your excitement you miss by a few pixels, and accidentally delete that first clip *forever*.

This scenario is entirely possible on YouTube Shorts, because the delete button is perilously close to the record button, and **there's no 'are you sure' modal**.

This could easily be avoided by adding an 'are you sure' message, like TikTok and Instagram do.

TikTok

Instagram

If they want people to capture original, moving and engaging content, then don't make it easy for people to mistakenly and **irreversibly** delete all their hard work.

More generally, when building products you need to be aware of where the points in the process are that people can accidentally lose data.

For example, if clicking 'back' during a sign up form will clear all the fields that they've already entered, you need to warn them.

5. Completing the reward loop

The journey of uploading content to YouTube Shorts is likely to be:

- 🧠

  Deciding

Creating the concept of what to film.

- 🎥

  Filming

  Actually filming those scenes.

- 😬

  Editing

  Adding effects and filters.

- ☝️

  Uploading

  Naming the video and clicking upload.

- 🏝️

  Reviewing

  Watching it back live.

That final step may sound arrogant, but content-creators know that this is an important part of the process.

Aside from checking that it has uploaded correctly, and that it looks good, it **completes the reward loop**.

It's the end of their natural journey, and the content that they created is out in the world.

Which is why I'm surprised that YouTube don't immediately take you to your own video after uploading it.

Instead, they take you to someone else's content, and then display a banner asking you if you'd like to watch your own video.

This is a perfect example of how data can be misleading. The immediate conversion rates of this process will be unaffected by this annoyance—because the user has already uploaded the content.

But, if they found the experience less **rewarding**, then it's possible that they'll not feel as incentivised to upload more content in the future.

And tracking that hypothesis between platforms will be difficult, given the many other variables (i.e., audience size, content length, monetisation).

One thing is for sure; **a rewarding *loop* is addictive**, and the YouTube loop doesn't feel as gratifying.

In other words, by never completing this reward loop, uploading to YouTube is probably less addictive than uploading to TikTok.

Closing thoughts

In my opinion, YouTube Shorts is considerably less addictive than TikTok, both as a viewer and as a content-creator. And that's a problem for YouTube.

Shorts may still succeed, but if it does, it's down to the sheer size of the YouTube community, not because the experience was world-class.

TikTok isn't perfect, as I demonstrated in my TikTok case study, but they've undeniably created a product that people love.

If you've used both YouTube Shorts and TikTok, you may get a vague feeling of one being better without being able to put your finger on why.

This is why—it's a perfect example of how impactful thoughtful UX can be.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

How TikTok works

May 19, 2020 • 16 minutes

#Social

-

How Zoom works

Apr 15, 2020 • 12 minutes

#Video Call

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

-

Delivering with Uber Eats

1 month ago • 18 minutes

#Delivery

#Apple • 12 minute read

Improving Apple's Airtags

-
- 2 Jun 21
- •
- 12 mins read

By Peter Ramsey

02 June 21

Apple are indisputably a leader in intuitive design—on many occasions raising the bar for how a product ***feels***, and delivering some of the best user experiences I've ever seen.

But after spending a few weeks using AirTags, I'm disappointed.

I also noticed that the majority of the reviews I'd read online used stock imagery, whilst demoing the tags in a controlled office environment. It's likely that the rush to release reviews _quickly_—for relevancy—meant that very few people tested them thoroughly.

But to appreciate the intricacies of a world-class UX, you need to really get 'into the weeds' of a product. Which is what I've done.

This isn't a *review* of the Apple AirTags, but rather an intimate analysis of how they could improve very specific parts of their user experience.

Even Apple misfire from time to time.

Summary

- 🗺️

  1. Why you need real-time positioning

- 🔋

  2. Your AirTag isn't reachable

- 😱

  3. The most stressful notification ever created

- 💬

  4. The issue with custom messages

- ⏱️

  5. The search for milliseconds

5 ways to improve the UX of AirTags

1. Why you need real-time positioning

If any Apple device gets within range of a lost AirTag (approximately 100 meters), then it silently—and unknowingly to the passer by—pings its location through the 'Find My' network, telling the owner where the item is.

Or at least, that's how it's *supposed* to work. My experience was underwhelming, and it took me hours to eventually trigger an alert:

But despite opening the notification immediately, it only showed me the location of the device **10 minutes ago**—as in, 10 minutes before the notification was sent.

This location being historical, and in not real-time, causes a few problems.

There's a nuance to location-tracking which turns a datapoint (a location), into useful information: the ability to predict a future location.

Most of the time this is subconscious and situational—i.e., when tracking a parcel, you *know* that the item is moving. But in the case of AirTags, you actually need two pieces of information:

- ⚡

  1. Where is my device right now?

  i.e., the exact location of it.

- 🚗

  2. Is my device moving?

  i.e., has someone picked it up, or is it still on the ground?

Without this context, it's hard to make a rational decision about what to do next. "*Is my bag in a bush by the side of the road, or in a car being driven away?*".

It's likely that Apple have the data to provide a far more useful breakdown, **even *without* knowing the live location**:

Adding 'previously spotted' markers to the map like this would hugely improve the user experience of *actually recovering a lost item*.

2. AirTag not reachable

If your AirTags are close enough to you, then you can tap a button within the app, and the device will make an audible chirp from its internal speaker.

It's designed to help you locate the precise location of a tag.

But if you're ***not*** within range, you see this:

"Move around to connect" is so ambiguous. It doesn't explain the cause of the issue.

From my research, there are many reasons why this message could appear, for instance, if someone has removed the battery of the AirTag.

But it doesn't even warn you of that being a possibility. Instead it's prompting you to 'move'. And what does 'Move around' mean? Wiggle my phone? Walk around my house? Drive around the block?

It'd be really easy to improve this *and* educate the user by making the notification clickable, which would open a modal with more information:

A non-interactive message telling you to 'move' is thoughtless and unhelpful—which is very rare for Apple.

At worst, people could find themselves walking round and round their house, not knowing that the AirTag battery *could* have been removed, and no amount of movement will help you find it.

3. The most stressful notification ever created

One of the news-worthy features of the AirTags is that if you try to use them to **secretly** track someone's location, then—if they have an iPhone—it'll send that person a notification to let them know that they're being tracked.

It looks like this:

And Apple have done a great job of explaining ***what*** this is, ***why*** it's been triggered and ***where*** it's tracked you.

> ***A quick note here:*** *it managed to accurately track me throughout the entire time where it was "not reachable" and wouldn't update the owner. (i.e., my previous suggestion of showing where the AirTag has been should be entirely possible).*

But besides removing the battery, there's not much that you can do at this point.

The only information that you can see about the owner is their phone number—which, if they've actually been trying to stalk you, will likely be fake, as there's no validation when entering it.

In other words: if you're really being stalked, then this information will be useless, so Apple may as well proactively help those experiencing false positives.

It's hard to imagine a more anxiety-inducing notification than being told someone is mysteriously tracking you—so the damage of a false positive is significant.

Here are a few things that Apple could provide access to, which may cut the amount of time that the user is panicking for:

- 👀

  Has the owner marked the tag as lost?

  This may change your perception of the situation entirely.

- 🗺️

  Where were you when the tag was first spotted near you?

  Where were you with other people (whose numbers are your contacts)? Where were you when the tag was first near you? How long was the owner of the tag with you, before you were on your own?

- 🚗

  The tag's name

  If it was called 'BMW Tag', then you may immediately remember that you've borrowed your friends car for the day and it's sat on your driveway.

- ⏱️

  How long ago the tag was activated

  If it was first activated the day it started tracking you, that's more suspicious than a tag which has been active for a year.

Although you may believe that you'd be rational and immediately think of these factors _anyway_—such as borrowing someone's car—you'd be surprised how irrational people can be.

My point isn't that Apple should add any of these data points specifically, but they should be proactively trying to make the UX of a false positive less stressful.

4. Custom messages

The process of marking an AirTag as *lost* is easy—it's a few taps inside the 'Find My' app.

It even starts with some light onboarding, explaining that you're going to leave a number **and a message** for the person who finds it.

They then show a *disabled input field*, with a pre-set message of "*This item has been lost, please call me*".

Confusingly though, that message is **non-customisable.**

Which begs the question: if this message is never customisable, then why is being displayed as an inactive input field?

For clarity, the confusion here is that having it *look like* an inactive field implies **it is sometimes a field**. But it's not, it's never customisable (at least, not yet).

But using a different element—like, basic text—would be far easier to understand, and introduce far less doubt. And a core aspect of building a great user experience is reducing the amount of time it takes to digest information.

5. The search for milliseconds

Maybe it's a British thing, but occasionally I find myself in a public area, such as a train station, and notice an A4 piece of paper with 'Wet Paint' written on it.

"Okay, but what paint is wet? All of it? The handrails? The walls? The doorway?"

Now imagine a very similar sign that says "*wet paint **on handrail***", with an arrow literally **pointing to the handrail**.

It's incredibly likely that in both scenarios I walk away without painty hands, however with the latter, I *worried less* about how to navigate the situation.

Or rather: even a basic 'wet paint' sign reduces painty hands, but its ambiguity creates anxiety and takes longer to process.

Apple have created this 'compass-like' mini-game to help you find items once you're within close proximity to them.

And in a well-lit office environment it works well, but if there is insufficient ambient light, then it doesn't work at all. It just shows a "***More light required***" message.

They haven't explicitly explained why, but the most popular theory is that the iPhone uses the camera to work out positioning and movement.

Just like a 'wet paint' sign, a few changes to the content—such as literally lowering the messaging on the screen, to be nearer the torch button—would certainly reduce the amount of time it'd take someone to understand and react.

Sure, we're talking about milliseconds—but it's a perfect example of how subtle design changes can create quick wins.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Using Apple Fitness+

5 months ago • 24 minutes

#Apple

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

-

Delivering with Uber Eats

1 month ago • 18 minutes

#Delivery

-

Signing up to Cash App

2 months ago • 20 minutes

#Payments

#Twitch • 20 minute read

Spending money on Twitch

-
- 20 Apr 21
- •
- 20 mins read

By Peter Ramsey

20 April 21

After Amazon acquired Twitch—the video game streaming service—for nearly a billion dollars in 2014, it was obvious how important monetising the service would be.

Because as YouTube have demonstrated, monetising video content can be tough. Particularly when people can install a free AdBlocker in a few clicks.

Often the pursuit of revenue comes at the cost of poor UX. For example, ads almost always worsen the user experience.

So how does Twitch make money? And more importantly, **do these revenue streams make the experience worse**?

Summary

- 💵

  How Twitch is similar to a fairground operator

- ⚓

  An example of the Anchor Effect

- 🪴

  De-risking decisions in CTAs

- 🗺️

  When to use self-discovery onboarding

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 107

Close

4 key takeaways

1. A pocket full of change

Fairground operators want their attendees to have disposable income (coins) in their pockets, as it's psychologically easier to spend coins of lower value, than notes of higher value.

i.e., how many people would break into a $100 bill to play a coconut shy?

One way to increase the amount of 'spare change' circulating in their park would be to price the admission at something like £10.50. Often people will need to break into a larger note, and then they'll have a guaranteed £9.50 burning a hole in their pocket.

It's a subtle trick which seldom works with digital payments, because spending money online—or via a debit card—doesn't ever leave "change".

But, it works brilliantly with platform-specific currencies, as those 'coins' are worthless in the outside world. i.e., you have to spend them in the 'fairground'.

Twitch essentially does this on IOS, with their virtual currency 'Bits'.

It's unlikely a coincidence that the the second tier of Cheermote costs 100 bits, but you can only buy ***95***, or ***245***. This forces you to buy more 'Bits' than you need—creating a pocket full of digital change.

This is a system designed to increase how much you spend, not one that's optimised for the best user experience.

On their desktop site you can buy 'Bits' in more convenient quantities, but it feels like on IOS they're capitalising on the power of defaults—i.e., because buying through IOS is easier and can be done with Apple Pay.

So, here's a quick way that Twitch could improve their UX: allowing people to be more efficient in purchasing Bits—i.e., increments of 10, 100 and 1000.

2. Anchor effect

If a group of people are asked to guess how many sweets there are in a jar, then you'll likely find that whoever guesses first sets a price *anchor*.

The **Anchor Effect** describes a bias that people have, where they'll become attached to an initial piece of information. For example, it'd be unlikely for you to guess 50,000, if the person before you guessed 5,000.

This doesn't only happen with price, but with almost anything. On Twitch, it happens when you're asked to make a decision about how **many months you want to subscribe for**.

i.e., once the user has decided to subscribe for 6 months, they're anchored to that amount of time, before they see the price.

When a new user subscribes to a channel for the first time, they're shown the benefits of subscribing, and then asked how long they want those benefits for, ***before*** being shown a price.

This is also what ultra-expensive watch shops do—they hide the price so you're not anchored to a good 'deal', but rather the benefits of a nice (and probably expensive) watch.

To be clear, although this looks manipulative under a microscope, it may be unintentional.

One thing is for certain though: **it's terrible UX**—if nothing else, because buying 3 months of membership required two separate transactions, and flicking back and forth between these two screens.

3. De-risking decisions

Most people don't like admitting that they were wrong, or having their beliefs challenged. This is so prevalent, that people don't even like *challenging their own past decisions*.

This means that once someone has decided to do something (e.g., to buy tickets to a show), they're unlikely to challenge that decision unless faced with new information—as that would mean that they were wrong before.

The key is this: the decision-making process happens before clicking on a CTA, not at the end of the process.

For example, the self-commitment of clicking 'Checkout now' whilst looking at your basket actually makes you more likely to complete the checkout, even if there's still 10 steps remaining.

When designing products, you need to be aware of where your key 'decision-making' moments are, and ensure that they're orchestrated in the correct way.

In general, you should encourage the user to make the hard decision (i.e., to commit to subscribing), after understanding all the benefits.

Twitch's 'Subscribe' modal is a great example of what ***not*** to do—they tell you nothing about what subscribing actually means, and then forces you into making a decision.

There's no reason for this to be such a hard fork, and they'd likely increase their conversions through this funnel by de-risking the decision at this stage.

To clarify, the button would go to the same place—the user still needs to learn about what subscribing means—but **their self-commitment will be challenged at a later point**.

Or simply: you're letting them learn about subscriptions, without the burden of committing too early.

4. Self-discovery onboarding

A popular type of onboarding is **self-discovery**—i.e., letting your users' curiosity be the driving force behind learning.

It can be really effective when done properly. The feeling of independence can be liberating, and the reward loop of learning can feel more natural.

But, for this to be effective, people need to actually be able to teach themselves how things work—it needs to be obvious.

As you'll see in the case study, there's very little onboarding or context around Emotes, Cheermotes and Reward Emotes. And that's a problem, because **it's not self-explanatory**.

Part of the issue is that the output of each (essentially an animated emoji) is very similar, and Twitch have done very little to differentiate them *conceptually*.

Differentiating between these may not be an issue for regular Twitch users, but it creates a **steep learning curve** for new users.

By taking more time to onboard users about these concepts, they would probably see an increase in spend —which benefits the streamers too.

In other words, investing time into their onboarding is a win / win.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Mario Kart Tour on IOS

Oct 10, 2019 • 5 minutes

#Gaming

-

Can Gamestop beat Amazon?

7 months ago • 20 minutes

#eCommerce

-

Playstation vs Xbox

3 months ago • 22 minutes

#Gaming

#Gaming • 16 minute read

Playing Among Us

-
- 7 Apr 21
- •
- 16 mins read

By Peter Ramsey

07 April 21

Among Us set a new record for the highest number of monthly players for a mobile game: approximately **500 million**, in November 2020.

For comparison, GTA V has sold 140 million copies since 2013, and at its peak, Pokemon Go had 147 million monthly players.

So, clearly Among Us is not only a much-loved game, but arguably the most successful mobile game ever created.

But, how could the user experience of Among Us be improved? What tricks have the developers missed?

And what **lessons about building great UX can we take from Among Us**? What can we transfer into other industries?

Summary

- ✏️

  Utilising predictive actions

- 👀

  The value of spatial context

- 🚧

  Clarifying what's intentionally inactive

- 📋

  Why Among Us needs a clipboard

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 101

Close

4 key takeaways

1. Predictive actions

It’s incredibly frustrating when you select an online lobby to join, but by the time you’ve clicked on it, it’s already full.

And whilst developers can’t really stop that happening, they can make the **experience of what happens in these instances** better.

Given a set of circumstances—like trying to join a lobby that is already full—you can predict the user’s next desired action: they’d try to join another.

I call this a predictive action: reducing friction by understanding what a user is likely to do next.

In this instance, Among Us kicks you back to the home screen, just to show you a modal saying that the game has already started.

But, given that the user will probably close this modal, go back to the available games, and try to join another, why not just show a simple message on the lobby screen itself?

Saving two clicks may sound insignificant, but in the world of user experience its not.

2. Spatial context

Some *things* are given context (and more broadly, meaning), purely by being close to something else. I don’t mean *similar to*, but actually physically positioned nearby.

For example, a ‘_keep clear_’ sign only makes sense when viewed in context of its location—if it were 100 feet down the road, everyone would be confused. I call this **spatial context**.

In the physical world this happens intuitively. If you see a ‘keep clear’ sign near a driveway, or a gate, you’d connect the two and understand the instruction.

But people often forget that pixels on a screen have the same impact.

So now let’s look at the lobby settings of Among Us:

How would a player modify these settings (i.e., to modify the game rules)? Well, they’d need to stand near the in-game laptop, press ‘Customise’ and then go to the final tab.

For regular players, this may work fine. But it’s not intuitive for new players, because it’s **out of sight**, and away from the *context* of those settings.

One simple change would be to add a button to modify these game settings, and put it literally near those settings.

One nuance to mention here is that often games will trade friction for immersion—i.e., forcing users to go into their “rucksack” to get an item, even if it’d be easier to just display the item at all times.

But, I’d argue that creating a game lobby isn’t an intentionally immersive or gamified experience. But rather a functional activity that requires clarity and context for new users.

3. Intentionally inactive

Imagine that you’re stood at a lift, pressing the call button. Nothing happens, so you press it again, and again. After waiting a few minutes you’d probably conclude that the lift was broken.

Now, how would your perception change if there was a sign explaining that the lift had been temporarily disabled for cleaning.

An intentionally disabled function without any explanation is just a broken feature.

With that in mind, let’s look at Among Us’ decision to disable (well, limit), the chat function based on how old the player is. They have a simple fork:

In principle, this may be sensible. However, there is **no noticeable difference** between an active or inactive chat box. There isn’t even any feedback when you interact with it—it just doesn’t do anything.

It’s the equivalent of pushing a broken call button, waiting for a lift that never comes.

And like my analogy, Among Us needs a sign, otherwise people may assume that the app is broken. Here’s how they could fix it with two simple changes:

4. Tap to copy

The Nintendo 64 didn’t need a clipboard, because it was *just* a games console.

But now games are built on feature-rich platforms (IOS, Xbox, Windows), with plenty of functionality available to their developers—and the utility of a clipboard is no longer questionable.

When you create a game on Among Us, you need to share a 6-digit code with your friends, which in practice means typing it out in a message, or saying it over voice chat.

But that first scenario is a bit awkward, because you can’t *copy* the code. The digits are rendered in-game, and so neither selectable using the native IOS functionality, nor does Among Us have a copy function.

And so players need to memorise the code, close the Among Us app, open their messaging app, and then type it out without errors.

A simple tap-to-copy feature here would almost entirely remove this friction.

If there were a 'heatmap of clicks' for this screen, I'd be willing to bet that a statistically significant number of people click on the code expecting this to be how it works anyway.

I know I did.

Conclusion

I get asked to do case studies on video games all the time, and have been pretty reluctant generally. This is because there's so much ***nuance*** to game creation, which breaks the fundamental laws of experience design.

Just to name a few examples:

1: Friction is intentionally added to increase the complexity of a challenge, even though the reward may be static (e.g., diminishing XP rewards from grinding particular tasks).

2: UI is designed to be aesthetically immersive, and not to maximise efficiently (e.g., the player's inventory may be designed as a scrollable rucksack, even though a basic *list* of items would be far easier to use).

But Among Us felt like a good opportunity to showcase the opposite: **how similar the concepts of building great UX for video games, and software can be**.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

More Case Studies

-

Spending money on Twitch

10 months ago • 20 minutes

#Twitch

-

Finding conversations on Clubhouse

12 months ago • 18 minutes

#IOS

-

How Robinhood handled a crisis

Feb 01, 2021 • 13 minutes

#Investing

-

Creating a Signal account

Jan 18, 2021 • 17 minutes

#IOS

#IOS • 18 minute read

Finding conversations on Clubhouse

-
- 9 Mar 21
- •
- 18 mins read

By Peter Ramsey

09 March 21

Although most founders *believe* that their app will see a 'hockey stick' growth trajectory, it very rarely happens. Even rarer whilst competing with some of the largest, most influential tech companies in the world; Twitter, Facebook and Snapchat.

In other words: the Clubhouse story is the perfect example of how exponential growth turns UX molehills into mountains.

You see, Clubhouse—the audio-chat social network—has gone from a few hundred thousand users, to around 10 million within just a few months.

But, scaling that quickly introduces new complexities.

Whilst early-adopters are typically forgiving with UX issues, the millions of people being invited to the app each month will be far less tolerant.

So let's look at the **experience of finding new content** on Clubhouse.

Summary

- 🏠

  Why Clubhouse's homepage is confusing.

- 🤯

  What people get wrong about 'aha moments'.

- 🔍

  How UX can be used as a crutch.

- 🚧

  Be aware of broken 'hooks'.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

Free UX newsletter trusted by product-lovers.

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Close

4 key takeaways

1. Homepage curation

Content feeds are notoriously difficult to get right. Which posts should you see? How should you order them? How do you filter out the noise?

On Clubhouse, once you've scrolled past all the available rooms in your feed, you're prompted to **follow more people, to see more rooms**.

In other words, they're describing how they decide what content you see.

i.e., your homepage is a curated list of rooms, based on people you follow.

Except there's a problem: I don't follow half the people who already appear in my feed.

To be clear, the issue here is the ambiguity about **how the feed is curated**.

How does this work? Am I seeing these random rooms because I showed an interest in a particular topic whilst signing up? If so, *how can I purge those interests*?

Over the long-term—when the initial curiosity has subsided—people invest less attention in a tool with unpredictable outcomes.

If the Clubhouse homepage is going to be an oasis of interesting content, people need a basic understanding of where the water comes from.

And so, instead of the vague instruction to "follow more people", they should take the time to actually explain **which levers control the content** that you see, and **how to operate them**.

2. The misnomer about 'aha moments'

In the early days of Twitter, the team noticed something in their data: *when people follow at least 30 others, they're far more likely to stick around*.

This is often described as an '**aha moment**'—the moment that the utility of a product really clicks for the user.

Or simply: if Twitter can get a new user to follow 30 people, then it reduces the chances that they'll get bored and delete the app.

This story has become start-up folklore, and I've worked with many companies who take this message too literally, forgetting the nuance of what they really found:

It's not enough to just follow *30 random people*. You need to follow 30 people who you genuinely care about.

Clubhouse have clearly adopted a similar methodology, by **pre-selecting 50 people for you to follow** whilst signing up.

Have you noticed that some people have accumulated millions of followers really quickly? It's because *the same people* are almost always recommended—I tried creating accounts with polar opposite interests, and the same people were pre-selected almost every time.

And at no point does it explain that following those 50 people will directly impact the content that is available to you, or that if your homepage gets uninteresting (as I mentioned earlier), you'll need to unfollow these people individually.

But they should, and it could look like this:

Right now, Clubhouse is designed for explosive growth—they just want to maximise the number of connections made.

But at some point they will likely need to shift to sustainable and longer-term *value* growth. This will mean encouraging fewer, but more meaningful connections.

Look at Twitter below: they explain *how* their timeline works, and don't pre-select people to follow—you have to opt-in to every single one.

3. A UX crutch for their search

Following the ***right*** people on Clubhouse is necessary to find interesting conversations, so it's important to know that you're following real people—e.g., the real Elon Musk.

Over time it'll be necessary to verify identities, like the other social networks do, but it's an unfeasible task for the small team that run Clubhouse currently.

Yet, they've built an app that is as **simple** as their rivals.

But therein lies a problem: established social networks can be so simple purely because they do so much behind the scenes.

And whilst a verification system is probably the *best* solution, it's also very expensive. Instead, Clubhouse could see some of the reward, with just a few subtle UX changes:

1: Over time you'd expect the real people to accumulate significantly more followers than the fake accounts. By **exposing their follower count** on the search page you could increase the confidence that people have about their authenticity.

2: Say there were 5 identical profiles, all with the same name, profile picture and a similar follower count, how would you know who's real? Well, by showing **their unique usernames** (e.g., @elonmusk), people could filter out the most obvious imposters.

4. A broken follow hook

It's no secret that social media companies want you to be addicted to their apps. They're designed to be infinity pools of content—always available to give you another quick shot of dopamine.

One of the most common tools used to create these habits can be described as a 'hook'—essentially a pattern of action and reward that is addictive.

And one of the key components of a hook is **an investment**, in this instance, the user investing time into *following people*, expecting more great content in the future.

But there's an issue: the reward for that investment is neither guaranteed, nor instant. And I don't just mean that the reward is *variable*, it can be entirely non-existent.

Ask yourself this: would you follow a Twitter account that'd never tweeted? Possibly, but you probably wouldn't follow 100 of them.

See, on Twitter, at the moment where you're deciding if you should follow someone, you can *see their content*. In other words: **the value is obvious and immediate**—their tweets are visible in your feed right away.

This gives you some confidence that they'll tweet again in the future, and therefore your *investment* in following them will be rewarded.

But on Clubhouse you could follow hundreds of people who you never interact with at all. In fact, you don't even know if they've *ever* joined a room.

Given these circumstances, which are inherent by design, Clubhouse should be using a thoughtful user experience to synthetically complete this hook as much as possible.

1: Reframe the reward — you cannot make it instant (unless they coincidentally join a conversation immediately), but you can remind the user that the value is in *not missing future conversations* they join.

2: Give them confidence that this person *will* be active — by showing scheduled events that they're attending, and previous conversations that they've been a part of, you can make people more confident that they'll join a Clubhouse room in the future. This will help the user *anticipate a reward*.

Conclusion

To be clear, I really like Clubhouse, and use it on an almost daily basis. And whilst the mystery behind how conversations appear in my feed may be exciting at first, I'm already finding myself tired of it.

Their minimalist design is simple—and yes, you can do almost anything in a few clicks—but it's not intuitive.

I know exactly how to control the content I see in my Twitter feed, but I still have no idea how to stop seeing random Clubhouse rooms in other languages.

And if *following people who you care about* is the primary driver behind building a great selection of rooms, then:

1: Why follow 50 *random* people at the start?

2: It needs to be easier to find the *right* people.

3: It needs to be rewarding to both follow the right people, and purge the wrong ones.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Creating a Signal account

Jan 18, 2021 • 17 minutes

#IOS

-

Mario Kart Tour on IOS

Oct 10, 2019 • 5 minutes

#Gaming

-

Using Apple Fitness+

5 months ago • 24 minutes

#Apple

#Investing • 13 minute read

How Robinhood handled a crisis

-
- 1 Feb 21
- •
- 13 mins read

By Peter Ramsey

01 February 21

Last week Robinhood took unprecedented action and, without warning, temporarily restricted trading on immensely popular stocks like GameStop (GME).

The circumstances of these actions were controversial and upset millions of people.

But this isn't a hit piece, and my opinion of whether the steps Robinhood took were necessary, fair or illegal are irrelevant.

Instead, this provides a perfect example of why UX matters in a crisis.

In this instance, even the best user experience wouldn't have stopped people from being upset, but would have acted as damage limitation.

As you'll see, the total lack of context, explanation or empathy only fuelled the mob.

So what can we learn from Robinhood's actions / inactions?

Summary

- 🔎

  Removing GameStop from search results.

- ✋

  Blocking people from buying GameStop shares.

- 💰

  Made fractional shares unavailable.

- 💸

  Creating sell orders on your behalf.

- 📋

  Failing to get statements.

5 key takeaways

1. Removing GameStop from search results

The first action that Robinhood took was to suddenly stop people from being able to purchase shares of GameStop.

But they also decided to actually **remove GameStop from any search results**.

This was an awful decision for a few reasons:

- 🤷

  1. Adds ambiguity

  People knew that GameStop should be there, so just removing it creates ambiguity. Is this a bug? Should they close the app and try again?

- 🤫

  2. Provides no explanation

  By removing it entirely, you've not given any chance for the user to learn more about why they're not seeing it.

- 🤨

  3. Adds suspicion

  During a crisis, you want to calm your users down, not give them reasons to suspect you're trying to mislead them.

But it didn't have to be that way—this is what they should have done (right).

Great UX is about being definitive and clear. The absenceof information is the opposite of that.

How long would it have taken Robinhood to add a few sentences, and a link to an additional resource? 15 minutes?

And if they didn't have time to write a full article explaining the situation, then they could have linked directly to a Twitter thread. Or, *literally anything* giving some context about this decision.

2. Blocking people from buying GameStop shares

But, even though GameStop shares were hidden from the search results, it was still possible to get to the screen where you can normally buy and sell them.

Which introduced another complexity: **how should Robinhood stop people from buying the shares, while still allowing them to sell their shares**?

What they did was:

1: Disable the 'buy' button.

2: Add an unspecific notice explaining that you can't buy shares right now.

Robinhood's users were missing out on what was being touted as the trade of a lifetime, and there's not even an explanation why.

It wouldn't have taken much to make the UX of this considerably better. Even just a link to an explanation would have helped.

3. Fractional shares are unavailable

After some time, Robinhood announced that they would allow people to purchase a limited number of these shares, this included **not allowing people to purchase *part* of a share**.

For context, Robinhood is renown for fractional investing. This allows you to invest a small amount of money in companies with a high share price, for example, one share of Amazon costs more than $3,000, but a Robinhood user could invest $300 for 1/10th of a share.

This is the message they showed to people who tried buying a fraction of a share:

There are a few nuances here worth pointing out:

- ⏱️

1. Clarify that this restriction is temporary

Is Robinhood not facilitating fractional shares anymore? Will this feature come back? Is this related to the other restrictions?

- 💸

  2. Explain what “enough available” means

  During this crisis there were many restrictions, including not being able to buy more than 3 shares. So is “not enough” related to the number of available shares you can buy until you reach that limit?

- 📃

  3. Add an explanation

  As we’ve demonstrated before, it’s important to give context and rationale behind a decision. Why did they have to take this step?

4. Creating sell orders on your behalf

As the chaos unfolded, people started complaining that Robinhood were creating sell orders on their account, without their consent. Once these orders were placed, the user could not cancel them, and were instead shown an error message.

If this wasn’t bad enough, many people were complaining that the selling price was below market value, given how volatile the price was.

So, if there was ever an example of where a company **needs to take an extra 5 minutes to really explain the situation**, this is it.

Yet instead, Robinhood users saw an error message—and to demonstrate how rushed this entire process was, they forgot to put a full stop at the end.

I’ve made a few important changes here:

- ☝️

  1. Change the title

  This isn’t an ‘error’, so labelling it as one feels misleading. Instead, be clear and tell the user what’s happening.

- 🙌

  2. Adding empathy

  This isn’t just an inconvenience, it’s potentially costing the user a lot of money. Somehow, saying “we’re sorry” at the beginning just doesn’t cut it.

- 💬

  3. Add an explanation

> You probably saw this one coming—this is a necessity in this instance, and it's easy to see why many people flocked to Twitter to complain.

5. Failing to get statements

This controversy unravelled over a few days, and led many people vowing to leave Robinhood and take their portfolio elsewhere.

But there was a problem: other brokerages were asking for your existing portfolio statement to initiate a transfer, and for some people, Robinhood's statement download function was broken.

So lets look at—and improve—the error message they displayed when people were trying to download their statements:

Unlike the other examples above, this error message doesn't require more ***context***, but it would have benefitted from suggesting an alternative step (e.g., emailing their customer support).

This is a task that could be completed manually, and may be urgent. To simply fail is not good enough.

Or rather, imagine how frustrated you'd be if your bank's app was broken and you couldn't make any payments. You'd appreciate a suggestion of: "_call us on [this number] to make a payment over the phone_".

Conclusion

One thing is certain: Robinhood were scrambling to react to these spontaneous decisions, and the execution was poor.

But the rush to push changes is not a valid excuse for a poor experience. In fact, it just fuelled the fire—probably contributing to the hundreds of thousands of 1-star ratings on the app stores, and Robinhood trending on Twitter.

It's in moments like this, more than ever, that a thoughtful and empathetic user experience is required.

So let's summarise the key points here:

1: Context matters — take those extra few minutes to explain to your users *why* you've made a difficult decision.

2: Remove ambiguity — give people realistic expectations for when certain restrictions will be lifted, and what they can do in the meantime.

3: Show empathy — apologising isn't enough, you need to demonstrate that you understand why your users will be suffering, and how hard it is for them.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

How Coinbase works

Aug 19, 2020 • 15 minutes

#Investing

-

How Freetrade works

Jul 21, 2020 • 19 minutes

#Investing

-

Can Gamestop beat Amazon?

7 months ago • 20 minutes

#eCommerce

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

#IOS • 17 minute read

Creating a Signal account

-
- 18 Jan 21
- •
- 17 mins read

By Peter Ramsey

18 January 21

Elon Musk, Jack Dorsey and Edward Snowden are just 3 of the influential advocates for Signal—a messaging app that's like WhatsApp, but with more of a focus on privacy.

And when WhatsApp announced changes to their data-sharing policy, Signal exploded.

But this rapid growth has introduced a UX complexity: the app is now being used by a much broader group of people.

Signal *may* be more private than WhatsApp, but if they ever want to rival their 2 billion users (yes, you read that correctly), then they'll need a product that's **just as easy to use**.

The golden question is this: when the hype dies, will people be able to convince their lazy friends, who don't care about internet privacy, to sign up?

The answer to that depends on a number of factors, one being how easy the experience of migrating from WhatsApp to Signal is. So let's take a look.

Summary

- 🧙

  Some Harry Potter UX magic.

- 🔒

  How Signal could improve their PIN creation process.

- 😍

  What a 'beautiful error' looks like.

- 🗺️

  The art of discovery.

- 💬

  Why friction may be tethering referrals.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 92

Close

5 UX takeaways

1. Contexto patronum (🧙‍♀️⚡)

Context is essentially UX wizardry.

It converts confusing and awkward tasks into things that people do on autopilot, without hesitation. For example, imagine if someone asked for a copy of your passport:

With that in mind, it's ironic that an app which so heavily forces the conversation of privacy, also asks for access to sensitive information **entirely out of context**, twice, during the account creation process.

📋 **What Signal asks for**

💡 **Missing context (i.e., when to ask)**

Access to all of your contacts.

This is the first thing you're asked for, before even providing your name.

When you try to start a new conversation, prompt the user that they can find other Signal users from your contacts.

Access to other devices in your house.

This is asked immediately after creating an account, with no real explanation why.

When the user is trying to transfer their Signal account to another device, prompt the user to search for nearby devices.

Here are a few questions to ask yourself while designing permissions modals:

1: Is it obvious **why** this information is required? (i.e., is the benefit specific).

2: Is the user **currently** trying to do a task that would benefit from this?

In Signal's case, the answer to both of those is no.

2. You failed, try again

I often find myself repeating the following: *if you're asking the user to set a password, tell them all the password rules*.

At this point it's a well-versed 'Built for Mars sin'. But there's a more frustrating version of this: failing a password rule, and **still not being told what the rule was**.

And frustratingly, this is what Signal do while setting a PIN number.

That's it, that's literally all you're given, which is problematic for a few reasons:

1: What makes a PIN code *stronger*?

2: How *much* stronger does my PIN code have to be?

But if we step backwards for a moment, you can see why this may not have been a problem up until now. Those who were motivated to sign up to Signal presumably had an interest in internet privacy, and probably also had at least a basic understanding of what makes a strong password.

But as people migrate from WhatsApp to Signal, their 'average user' has changed. And what's there today is no longer self-explanatory.

There are a few ways that they could tackle this:

- 👩‍🏫

  1. Teach the user more generally about passwords

  i.e., a link to learn more about what makes a password 'strong'.

- 👮‍♀️

  2. List specific rules

  i.e., "must not be a series of the same digit 4 times".

- ☝️

  3. Show tips for a strong password

  i.e., "you can make this password more secure by using up to 6 digits".

3. The beautiful error

And after that validation issue, I submitted a new PIN and… Signal errored. I later found out that the flood of new users may have been the cause of the problem.

But I'm not even mad, because this error handling was beautiful.

To understand why, let's dissect it into 4 parts.

1: What just happened — it's clear right away what the error was (in plain English, not an obscure code).

2: Why this affects you — this is the result of that error, and why you should care.

3: What happens next — this is how the error will be handled. Without the promise of this automatic reminder, the user may be *anxious* about having to *remember* to set a PIN in the future.

4: Here's the next action — this gives the user a clear next action, and lets them continue *using the app as normal*.

Most of the time when designing software, people don't remember (or don't want to invest the time into) good error handling.

I see this (left) all the time:

Tip: if you're building software, it's worth the investment to handle errors well.

4. The art of discovery

There’s an onboarding strategy which can be difficult to get right, often backfires if done incorrectly but has the potentially to be immensely rewarding: **encouraging discovery**.

The feeling of using an app and stumbling upon a ‘hidden’ feature is enjoyable and addictive. It gives depth to a product, that otherwise might have appeared quite shallow.

Great UX is built on these moments of discovery, and Signal nailed it with their ‘Note to self’ feature.

If you go to start a new message, and search for ‘S’…

New message

Search 'S'

Boom

You can now privately send messages to yourself—and yes, it’s very similar to the ‘your space’ feature on Slack.

But the point isn’t that it’s innovative, it’s that Signal took a ‘hands off’ approach, and let the user stumble across this in their own time.

✅ **When this technique works**

❌ **When it doesn’t work**

Additional features which **aren’t** required to get the core value from the product.

Core features which need to be **understood immediately**.

Features that would be used **infrequently**, or have a *relatively* **low ‘value’** to the user.

Features that would make their experience **considerably better** *(it’d be annoying to find a game-changing feature after using the product for a year)*.

The nuance here is to know when to proactively teach someone how to use a feature, and when to let them discover it on their own.

5. Friction is your Achilles heel

Here are two reasons why word of mouth is so important to Signal:

- 🤷‍♀️

  1. WhatsApp and Signal are substitute goods

  i.e., most people probably won’t routinely use both in the long term.

- ☝️

  2. The network effect

> i.e., they both get more valuable when more people in their network also have them. For Signal to be as useful as WhatsApp, you also need your friends to switch.

Which introduces a UX challenge: how can you design an experience which maximises the number of new users that then bother to invite all of their friends?

Certainly an influential factor is how much effort it takes to **create an account** (people are more likely to share things that they thought were easy to do).

Signal should be designing a process that gets the user to a point where they feel like they've replaced WhatsApp, in as few steps as possible.

That means removing anything from the initial onboarding experience that doesn't directly help the user get to that milestone.

And yes, having to decide whether or not to give Signal permission to access all the devices in your house *does* take effort.

It's not just the physical action of reading the modal and then clicking a button, but the cognitive attention required to make a ***decision***.

Conclusion

I spent about 40 hours using, analysing and writing up my experience of using Signal, and—in addition to the privacy angle—there's certainly a lot to like:

1: *Some* of the error handling was great.

2: I was delighted to stumble across the 'Notes to self' feature.

3: The core product is very similar to WhatsApp, which is a good thing.

But there's still a lot of work to do, and dare I say it; **the UX wasn't as good as WhatsApp's**.

I created a new WhatsApp account to test the most recent version, and I don't think anybody can deny that their product is world-class.

Signal *could* get there, and to start they should improve the following:

1: For an app so focused on privacy, they do a really poor job of asking for permission to access your *stuff*. It's often totally out of context.

2: *Some* of the error handling was really poor.

3: Remove any unnecessary steps, so the process of replacing WhatsApp *feels* easier.

When the hype around Signal dies down, and people are trying to get all their friends to migrate from WhatsApp to Signal, it'll be the UX that makes all the difference.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Playing Among Us

11 months ago • 16 minutes

#Gaming

-

Finding conversations on Clubhouse

12 months ago • 18 minutes

#IOS

-

Mario Kart Tour on IOS

Oct 10, 2019 • 5 minutes

#Gaming

-

Using Apple Fitness+

5 months ago • 24 minutes

#Apple

#Music • 18 minute read

How Group Sessions on Spotify work

-
- 3 Jan 21
- •
- 18 mins read

By Peter Ramsey

03 January 21

The music industry has been transformed since Spotify was founded in 2008. Music streaming has become so successful that many people wouldn't even consider owning a physical format of music.

For some—aside from live events—the ***entire experience*** of music is online.

So when Spotify started beta testing a new feature called 'Group Sessions', which lets friends listen to music together remotely, I expected their UX to be world-class.

But is it? Well, let's take a look.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

Go fullscreen

You can navigate the slides using your keyboard.

Yeah, understood

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 80

Close

4 UX takeaways

1. Usernames vs display names

Although they're often confused, there are two *types* of 'name' for your users:

Username: typically unique, and references a *specific* user.

Display name: friendlier and can be anything, such as 'James'.

And there's no better place to demonstrate the importance of this distinction than Twitter.

Display names serve an important function: they allow for personalised experiences within large databases, without the incumbency of everything being unique.

For example, the version on the right is better because it's immediately recognisable as a **real person**.

Yet, if you connect to Spotify through Facebook, then you're arbitrarily assigned a display name—which is my case was a random 11 digit number.

Spotify would improve their UX by prompting the user to change their display name to something more genuine ***before*** creating a Group Session.

1: It's a task that only needs to be done once, ever.

2: Avoids any confusion over who '_11142311180_' is.

3: Creates a more 'real' and personalised Group Sessions experience.

2. Be one move ahead

Imagine if every time you clicked on a link, it opened in a new tab, but ***didn't*** then set that tab as active. i.e., you had to *then* click into that tab.

It'd be infuriating to use, and is a brilliant example of a subtle UX trick that turns a 2-step process into a single action.

- ☝️

  Action 1: Open a new tab

- ✌️

  Action 2: Set new tab as active

This works because browsers have correctly assumed that if someone opens a new tab, they probably also want to be ***on*** that new tab.

Or rather: they're confident that they know what your next action would be (going on that tab), and so they just do that for you.

And Spotify nail this technique when copying a 'Session' link to share with your friends.

They've combined the actions of **copying a link**, and **clicking to go back** into the group lobby.

The result is a pretty seamless and uninterrupted experience, which works because they understand that once their users complete *Action A*, they're incredibly likely to want to do *Action B*.

Tip: Don't just *assume* what your users will want to do. Instead, look at the data. If less than 80% of your users complete both actions in perfect succession, you should probably leave them as two separate steps.

3. A decision without context

As soon as you've clicked on someone else's link, you're faced with a decision: **how do you want to connect to this session**?

- 📱

  1. All control the music on one device

  i.e., multiple devices picking music for one speaker at a party.

- 🌍

  2. Listen on your own devices

  i.e., control the music for everyone listening simultaneously, like a radio station.

These are totally different group experiences. So you'd hope that the selection screen is primed with enough context about each option, that you can comfortably make a decision.

But unfortunately, **it's not**.

It's practically a law of UX: if you're asking the user to make a decision, give them enough information to actually be able to decide without the fear that they chose the wrong option.

And Spotify could do this fairly easily, with a few additional lines of text.

4. "React and explain"

There are many approaches to onboarding, but let's look at two:

Clearly, sometimes you need to onboard users **before** they do something—to explain the consequences of an action.

But, if you've got a product with lots of functionality, it's often not appropriate to attempt to onboard every possible action or feature.

Another technique is reactive onboarding: explaining why something just happened.

And—unless they add more proactive onboarding—this is the bare minimum of what Spotify **should be doing** while people are using the Group Sessions feature.

To be specific, any participant can pause, skip and play songs, for *everyone* in the group. And yet, it neither explains how this works in advance, nor does it explain what's **just happened**.

So by adding a simple notification when another member of the group takes one of these actions, it'd do a few things:

1: It teaches the user that anybody **can** take control.

2: It clarifies *why* the music has stopped / skipped.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

How TikTok works

May 19, 2020 • 16 minutes

#Social

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

-

Delivering with Uber Eats

1 month ago • 18 minutes

#Delivery

-

Signing up to Cash App

2 months ago • 20 minutes

#Payments

#Purchasing • 18 minute read

Buying a card on Moonpig

-
- 11 Dec 20
- •
- 18 mins read

By Peter Ramsey

11 December 20

Where do you buy Christmas cards if you can't physically get out to a shop? Well, there's a good chance you'd turn to Moonpig.

Moonpig is an online greeting card store, and although they do sell other types of gifts, the core of their business has been cards since they were founded more than 20 years ago.

So you'd think that after all this time, the user experience of buying a card would be great by now, right? If it's the absolute heart of what they do.

Summary

-

What the doorway effect is.

- 🌎

  The gravity of interesting content.

- 📝

  How to handle blank pages better.

- ☔

  Why you need to track error messages.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 94

Close

4 UX takeaways

1. The Doorway Effect

Do you ever walk into a room, and immediately forget what it was that you were about to do? We all do occasionally, and it's delightfully named the 'Doorway effect'.

As a crude summary, this momentary lapse in memory is caused when our brain switches between tasks with different contexts and environments, such as leaving the kitchen where you were cooking dinner, to entering the garage looking for a screwdriver.

But this isn't just limited to the physical world—it happens digitally too.

For example, after personalising and adding a card to your basket, you're shown a generic account creation screen, and then asked if you want to be reminded about buying cards *next* Christmas.

Generic 'create an account'

Extra 'remind me' feature

Ordering a card should be a **very focused** experience. Interrupting the user's state of flow with the combination of these steps may create that moment of unnecessary confusion.⛱️

- ⛱️

  1. Change of environment

  The generic sign up pages look different. They have a new design, and nothing on the page explicitly mentions purchasing a card.

- 🚦

  2. Change of context

  This additional step changes the context from a purchasing decision (this card), to a planning task (for next Christmas).

To clarify, most users won't encounter this issue at all. Just like how most of the time I can remember what it was that I entered a room for.

But occasionally, perhaps more often when there are influential external factors—like trying to simultaneously have a conversation—this user experience will fall over.

And when it does, the user may find themselves sat for 20 seconds trying to remember what it was that they were doing.

2. The gravity of interesting content

Imagine that you're walking through a foreign city and looking for somewhere to eat. In the distance you see the golden arches of McDonalds. What happens next?

Well, it's possible that you'll go into autopilot and ignore the other options around you. The *ease* of getting a McDonalds outweighs the *effort* required to weigh up new options, despite you knowing that healthier and tastier food is probably available.

It could be said that McDonalds has a **sort of gravity**, that pulls hungry people in.

This also happens in the digital world. Sometimes it's called clickbait, but often it's just the gravity of really interesting or relevant content.

Moonpig have an example of this, take a look at their Christmas cards page.

For someone who's browsing Christmas cards in mid-December, the promise of being able to get it delivered *before* Christmas is almost impossible to ignore.

And that's **not** inherently a problem. It can be used to great effect, and I'm sure it has an incredible click through rate.

But when designing a product, you need to be aware that this gravity comes at a cost: people basically ignore everything around it.

As it turns out, the user could actually save a click by using the functionality above that section, and jump straight into more filtered results. But they won't, because the promise of delivery before Christmas is just too enticing.

So by adding a specific banner with such a high 'pull', (*rather than the heading of the entire page explaining the delivery schedule*) you may actually give the users a slightly worse experience overall.

Or to reuse the analogy, you thought that having a McDonalds next to your bakery would bring you more customers, but instead, all your customers are eating Big Macs.

3. This page is intentionally left blank

Some cards come with content printed on the inside, and so it makes sense that Moonpig lets the user see what's *inside* them before purchasing one.

Here's a quick challenge for you: which of the following is a **blank page**, and which is of an image **waiting to load**?

You'd never be able to tell, as they're identical—and that's a big UX problem.

The problem isn't that the images load slowly—they're actually very fast—it's that a new user doesn't *know* if an image is loading, and that **ambiguity can be confusing**.

The only real option that the user has is to wait for enough time to pass until they're confident that if there *were* an image, then it'd have already loaded.

Which is crazy, really. Imagine if no product on the internet ever said that it was out of stock, and instead you needed to wait and watch a 'checking stock' spinner, until you were satisfied that there were none available.

Really, there needs to be a prompt. Something that lets the user know that the page has intentionally been left blank, even if it's just a subtle cue.

4. Tracking errors

You've just personalised a card, added it to your basket and now you want to proceed to the checkout. Looking at the screenshot of your basket below, **where do you click**?

A very large percentage of you would have clicked on the red **Checkout** button on the right-hand side. After all, that's how most checkout processes work.

But you'd be wrong, because you'd see an error message telling you to add a recipient first.

And that's frustrating, because with a few basic design tweaks, Moonpig could probably stop people from ever seeing this error message again:

- ✅

  1. Disable the checkout button

  It could have been disabled until the user has added a recipient.

- 🎯

  2. Button placement

  The checkout button is just too clickable, as it's located exactly where you'd expect a checkout button to be—so people won't keep looking for other options.

- ☝️

  3. An additional step

> The ‘Checkout’ button could just take you to a new screen that asks you to add recipients for all your cards. No error required.

What’s more frustrating, is that if Moonpig have set up their analytics properly, then this error will be so obvious.

But you’d be surprised how rare it is for companies to actually be tracking error messages effectively. I see it all the time.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

How Klarna Works

Sep 17, 2020 • 18 minutes

#Payments

-

Buying a domain on GoDaddy

Mar 05, 2020 • 7 minutes

#Domains

-

How creating an event on Eventbrite works

Feb 21, 2020 • 4 minutes

#Purchasing

-

Signing up to Notion

Jan 17, 2020 • 7 minutes

#Productivity

#UX of Payments • 21 minute read

Building a store

-
- 19 Nov 20
- •
- 21 mins read

By Peter Ramsey

19 November 20

Up to now, this publication has focused on very specific parts of the ownership experience.

But it's time to take a step back and consider a more general question: **what it is like to run a store** with one of these devices?

And, while we benchmark that experience, it'd be unfair to not include the billion-dollar behemoth that is **Shopify**. So for this final chapter, I'll be throwing them into the ring too.

> ***Summary: Verifone*** *are so far behind with their software—compared to iZettle, SumUp, Square and Shopify—that I doubt they'll* ***ever*** *catch up.*

Key takeaways

- 🙈

  Why you can't blame the chimpanzee.

- 📈

  Why mundane tasks hurt so bad.

- 🏌

  The importance of timing.

- 👮‍♀️

  How data lies.

- 🚶‍♀️

  The danger of MVPs.

Case studies

Select a presentation to watch

There are **3** case studies available, you're currently watching 'iZettle'.

There are **3** case studies available, you're currently watching 'Square'.

There are **3** case studies available, you're currently watching 'Sum Up'.

iZettle

iZettle

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

Yeah, understood

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 72

Close

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 88

Close

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 48

Close

Benchmarking the experience

If you Googled 'iZettle vs Square', or something similar, then you'd be inundated with comparison tables.

But most of these have an achilles heel: they're all comparing stuff that you don't really care about.

> *"iZettle has feature A, SumUp has feature A, Square has feature A…"*

Instead, let's look at 5 'real-world' scenarios—things you'll actually do, and miss if they're not there.

This includes the new Verifone portal that I've been given early access to, which is currently in a limited Alpha.

What's possible with each service?

iZettle

Square

SumUp

Verifone

Shopify

Saving products in the app, so the seller can quickly add to a 'checkout' later.

✅

✅

✅

❌

✅

Offer **custom discounts** or vouchers.

✅

✅

❌

❌

✅

Print receipts from a _normal_printer, from a mobile phone.

✅

❌

✅

❌

❌

Automatically manage an **inventory** of items.

✅

✅

❌

❌

✅

Offers **integrations** to major **accounting tools** (or IFTTT).

✅

✅

❌

❌

✅

I know—I was disappointed at Verifone's portal too. Perhaps that's not the aim of this platform, but then what is the point in it?

Credit where it's due, iZettle excelled here. They've got a really deep feature-set, solving even minor annoyances.

But as always with Built for Mars, what was the more general experience like, and how could these services improve their UX?

The experience

1. We're all chimpanzees, really

If you gave a chimpanzee a screwdriver and some Ikea furniture to build, you'd probably be disappointed when you came back a week later and found that they're only using the tool to fish for termites in mounds of soil.

Even if these chimpanzees had a perfect comprehension of the english language, you _still_couldn't blame them, because you didn't explain the core principles:

- 🏠

  What the end result is

  i.e., what is the Ikea furniture?

- 🔥

  The benefit of the end result

  i.e., why should I want to have this furniture?

- 🧠

  How the tool works

  i.e., when do I need the screwdriver and how does that work?

Without this foundation of understanding, it's unreasonable to expect somebody—especially a chimpanzee—to use your product in the right way.

As a product designer, it's your responsibility to convert new users into power users, not theirs.

This example sounds *silly*, but this is more or less what has happened to many of the 'power-user' features inside these payment services.

A good example of this are the coloured tags that you can associate to products.

Literally, at no point is it explained **why** anybody would want to use these, **when** they should use them or **how** to use them effectively.

As a result, I doubt many people ever *do* use them—at least not in a productive way.

2. Exponential effort required

iZettle, SumUp and Square all allow you to create **variables**—for example, if you were selling T-shirts, then they'd be 'small', 'medium' and 'large'.

But what would happen if you had two variables, like size **and** colour?

The answer is that the number of combinations grows exponentially. A T shirt with 4 sizes and 4 colours has 16 possible combinations.

Now imagine that a coffee shop owner is adding 'latte' as a product, and there are 3 variables: 4 sizes of cup, 4 types of milk and 4 types of bean—there would be **64 combinations**.

The point is this: unless your software **automatically** creates these combinations based on your variants, then it quickly becomes unmanageable.

iZettle and Square do, but SumUp doesn't.

Technically, this is just the lack of a feature. But the frustration for the user is amplified by a few psychological biases.

- ⛰️

  Peak end rule

  You're likely to remember this mundane and boring experience, even if the rest of the app is great.

- 🗺️

  The planning fallacy

  People underestimate how long tasks like this will take, so it's likely that they'll be frustrated with how long they spend doing it.

In short: this would be a major point of friction for many users, possibly even deal-breaking.

3. How not to time onboarding

When it comes to onboarding, timing is **everything**.

If you show the user how to do something too early, then they won't pay attention. Too late, and well, the damage has already been done.

Imagine if the instructions for how to build your Kinder egg toy were carved into the chocolate shell—you'd have eaten the instructions before you realised that you even needed them.

It can be frustrating to use **Square** as a new user, because although they have 'learn more' links dotted around the app, they're often a few actions **too early**.

For example, when creating options and variations for your products, there is literally no guidance at the point where the user will actually need it.

4. Data lies, (and is lying to these services)

Imagine that you're trying to buy tickets to a popular event, and halfway through the process you're shown this screen:

What would you do? Well, you'd most likely just click that *proceed* button. And therein lies a problem.

If you were looking at Google Analytics, this page would have a normal conversion rate. There would be no symptoms of a bad experience in the data.

I call these *leap of faith* pages. They're essentially pages where there is only one obvious action—so regardless of how awful the UX is—the user takes a leap, clicks the button and proceeds.

And these types of pages don't necessarily have to be ugly like in my example, they can be the opposite: totally empty, like Square's customers page.

You see, other than the main menu, there's really only one clickable item for the user, and that's the **three dots** (top right).

Most of Square's users who see this screen will click those 3 dots, and stumble across the 'create new customer' link. But that doesn't mean it was comfortable.

Is this screen self-explanatory? No. Is this good UX? No. Will the data show that users have an issue with this? Unlikely.

5. Most MVPs suck

Here's an ugly truth: most companies don't understand how to produce an MVP (minimum viable product), and they're a total waste of time.

It feels like the whole purpose of an MVP has been forgotten, and companies use it as an excuse to release half-baked services—or as a buzzword to postpone taking the time to _really_understand their users.

MVPs should **not** be buggy, or frustrating and awkward to use.

A more accurate definition of an MVP is this: the minimum product required to actually learn from your users.

Their purpose is to **validate assumptions** about a product's requirements. But how can you determine if people *like* or *need* a feature if it's horrible to use?

You often can't.

With that in mind, Verifone have given me access to the MVP of their new portal, which is essentially their take on the software layer that iZettle, SumUp and Square have been building for more than a decade.

The data that Verifone accrue from this version may be wildly unrepresentative of how users would *really* like to use their software.

For example, if we just focus on the screenshot above:

- 📋

  How often will people check their reports?

  Probably not very often, because it's awkward to do. Does this that mean people don't want to check their reports? No.

- 🗳️

  Which filters do people want?

  Verifone won't know, because when users realise how fiddly they are to use, they won't bother using any of them.

- 🔭

  Do people search for individual transactions?

  Because the table is so hard to use people may rely on using the search function. Verifone could read into this as "people want a better search", which might not be true.

The risk here isn't just annoying your users, but building the wrong thing based off misleading data.

Conclusion

The irony of incumbent organisations building software, is that in addition to being far worse at actually building it, often they don't *understand* it enough to know how **far ahead** the competition are.

And I worry that Verifone is in a similar position. I doubt they'll ever catch up.

Yes, as I demonstrated in chapter 3, Verifone is the cheapest provider. But, honestly, I'd rather pay for **any** of these 3 services. I'd probably pay double what they're charging now—that's how much better they are.

One thing is for sure: none of them are finished. The user experience of building and running a store using these devices hasn't peaked.

All of these companies will need to listen to, and understand, what drives small businesses—their goals, frustrations and hardships. And then take the time to obsess over, and fix, the tiny annoyances that stop people *really* loving their products.

And that's exciting, because if there are still improvements to be made, then starting a business in 5 years should be even easier than it is now.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Opening an account

Oct 29, 2020 • 20 minutes

#UX of Payments

-

The card reader

Nov 05, 2020 • 21 minutes

#UX of Payments

-

Making a payment

Nov 12, 2020 • 19 minutes

#UX of Payments

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

#UX of Payments • 19 minute read

Making a payment

-
- 12 Nov 20
- •
- 19 mins read

By Peter Ramsey

12 November 20

The experience of making a payment is undeniably the most important user journey for these services.

It's a process that on the surface looks very simple to build, but is subject to awkward use cases which make it incredibly difficult to get right.

Which of these payment service providers has the best experience—for both the employees, and the customers?

Let's dive in.

> ***Summary**: Most iZettle, Square and SumUp customers would save money by switching to Verifone. But that **doesn't** mean they should.*

Key takeaways

- 🙋‍♂️

  Why you'll need to turn off your phone's notifications.

- 🏝️

  Why Verifone isn't suitable for everyone.

- ⚡

  How (not) to time permission modals.

- 👈

  SumUp's lack of an undo button.

- 👀

  Why the consumer benefits from a screen.

- 🌍

  The problem with using Square at a car boot sale.

Case studies

Select a presentation to watch

There are **3** case studies available, you're currently watching 'iZettle'.

There are **3** case studies available, you're currently watching 'Square'.

There are **3** case studies available, you're currently watching 'Sum Up'.

iZettle

iZettle

Square

Square

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-
-
- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 25

Close

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 34

Close

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 28

Close

Benchmarking the experience

These services make money by charging you a fee as a percentage of every transaction. So if you sell a coffee for £3, you'll pay a few pence.

Percentage fee (on my package)

iZettle

Square

SumUp

Verifone

1.75%

1.75%

1.69%

1.00%

And although the difference between 1% and 1.75% may seem insignificant, it quickly becomes noticeable.

Here are three scenarios, showing what the total costs would be in the first year of usage, including purchasing the card reader.

Estimated year 1 costs (device + card transaction fees)

£500 in monthly revenue (12 months)

128

136

140

140

£2,500 in monthly revenue (12 months)

380

542

548

560

£5,000 in monthly revenue (12 months)

680

1049

1073

1085

Total cost for 12 months (GBP)

At £5,000 in revenue per month, Verifone will save you hundreds of pounds every year.

These prices don't scale infinitely though—they all offer custom pricing to larger companies. But how much do you need to be making before iZettle, Square and SumUp will even consider giving you a discount?

iZettle

Square

Sum Up

Advertised monthly turnover to **qualify** for a discount.

£10,000

£20,000

£8,300

Annual **savings** at *that* point if you were with **Verifone**.

£855

£1,743

£642

And how realistic is it for a small business to ever reach these thresholds?

A recent study suggested that 43% of cafes have a revenue of less than £100,000—the minimum required to even be considered for SumUp's custom pricing.

And for many industries—like those who sell homemade goods at weekend events—£100k in annual revenue is well out of reach.

I think it's reasonable to suggest that many—if not most—of iZettle, Square and SumUp's customers could save money by switching to Verifone.

Essentially, customers are paying for access to the iZettle, Square and SumUp apps. That begs the question: **are they worth it**?

The experience

Here are 6 ways that these services could improve their UX.

1. Handing your phone to strangers

Square's entry-level device stands out for a few reasons, one being that it doesn't have **a keypad**.

So, when contactless payments aren't suitable, the user needs to input their PIN number on the device that *the seller is using*.

Not only does ***entering*** your PIN number into a strangers mobile phone require a serious level of trust, but it's also **uncomfortable for the seller**.

"Okay, enter your PIN when ready, please don't read my messages".

And yes, you **can** still receive notifications while on the PIN-entry screen.

How many people would feel comfortable passing their mobile phone back and forth between customers all day?

And as a customer, how would you feel knowing that the seller probably plays with their 'card reader' on the toilet?

2. A virtual basket

> *"Can I have 3 lattes and two shortbreads please?"*

This introduces a new variable: **selling multiple items at once**.

If you're using **Verifone**—and hadn't invested in any additional retail hardware—then your only choice would be to add up the value of these 5 items, and then type *the total* amount into the card reader.

And this is where the software of iZettle, Square and SumUp can demonstrate why it may be worth paying for—they offer virtual baskets.

iZettle

Square

SumUp

Being able to add items into a basket like this fixes a number of user experience issues that Verifone face:

- 🔔

  No calculators required

  i.e., it adds it up for you, and gets it correct every time.

- 🌯

  No need to remember exact values

  i.e., was adding cheese to the burrito +20p or +30p?

- ⚡

  Therefore, sales are quicker

  i.e., so you can serve more customers in an hour.

For some, not offering this feature would be a deal-breaker. It'd depend on how complicated your sales were.

3. Timing permissions modals

In most instances, you want to ask the user for permission to access their data in the context of ***doing*** that action.

> *i.e., only asking for permission to access their photos at the point that they are trying to add a photo.*

But there are a few exceptions, including this one:

Don't ask for permission for something at a moment of high intensity—such as while making a transaction.

iZettle require access to your location in order to accept card payments, which feels totally reasonable. However, they ask for your permission *whilst* you're making your first sale.

In the excitement of making a sale, you may be surprised by this request and click 'Don't Allow' out of habit. Perhaps you didn't even read what it was asking for.

And what would happen if you did **accidentally click 'Don't Allow'**?

Well, the modal closes, and it looks like the payment can go ahead. But when the customer taps their card…

If this happened during a real transaction, then you'd have to ask the customer to wait while you dive into your iPhone settings.

Instead, iZettle should request this permission *before* the sale is initiated—perhaps when they're connecting the card reader. If nothing else it'd help avoid any potential embarrassment.

4. Cancelling a payment

Imagine that you've initiated a payment, but ***before*** the customer taps their card, you realise that you've entered the wrong amount.

You'd obviously want to cancel the payment and start again.

But with **SumUp**, this is more awkward than you'd probably imagine. You only have two options:

- 😐

  Wait 60 seconds

  After which it'll automatically cancel the payment.

- 👉

  Click a physical button on the card reader

  It's the power on/off button.

Or rather: you can't cancel a transaction via the app. Instead you're left starring at this inescapable screen.

This is a process that you can only **initiate** using the application, and then only *cancel* using a separate device.

There are obvious reasons why this is frustrating, but perhaps the most practical one is that the card reader might not be in your hands—it might even be in your customer's hands.

And as Covid-19 changes the way we interact, it's entirely possible that the card reader is on the 'customer side' of a sheet of perspex.

5. Tap whenever you're ready

There's a light on my oven which tells me when it's pre-heated to the correct temperature. However it's entirely useless to me, because I can never remember if the light goes on or off when it's ready.

And that's the problem with a single and unlabelled LED—you never know if 'on' is good or not, or what it does.

With that in mind, look at the 'ready to pay' states for each of these devices:

With Square's device, how does the customer know when to tap their card?

Sure, in a nice and relaxed coffee shop the employee could ***say*** "tap your card when you're ready". But in a chaotic, noisy and under-staffed environment, that may not be possible, or audible.

The screen acts as a **visual cue**. It's not really for the benefit of the employee, but for the end customer. It's a subtle signal that helps the logistics of every transaction go slightly smoother.

And as an example of how important these cues can be, imagine if there was no *beep* sound when making a contactless payment. How would you know when to withdraw your card?

You wouldn't, and it'd be annoying.

6. The other benefit of the screen

But knowing ***when*** to pay isn't the only benefit of a screen. There's something else it does which is so obviously important that it's hard to imagine how a device like this could exist without one.

The screen shows the user how much they're about to pay.

Imagine that you're on holiday in a foreign country, at a Christmas market stall. You know that you'll never return, you don't speak the native language and you'll have no way of contacting the seller ever again.

Would you feel comfortable making a contactless payment **without seeing the value of the payment** on a screen?

Consider this:

- 🔔

  Not every bank has immediate push notifications

  I did the research, they don't.

- 📱

  The seller may not show their screen

  i.e., the app says what the payment amount is, but the seller could just not show the buyer their screen.

- 🇲🇲

  Language barriers

  i.e., the verbal nuance between 6 and 16 sound very similar in some languages. But numbers are universal.

As a customer, buying something through a screen-less device requires trust. Why would any seller make it harder for themselves to make a sale?

In some contexts—such as the Christmas market above—Square's device is so basic that the general user experience suffers.

I'm repeating myself, but I need to clarify this here: Square *do* offer a reader with a screen, it just costs £199.

Conclusion

Most iZettle, Square and SumUp customers would save money by switching to Verifone. But that doesn't mean they should.

I'd argue that the software that these services provide are not optional, they're almost essential to run a retail business in 2020.

And it's also hard to recommend the entry-level card reader from Square. Sure, it's £10 cheaper than the others, but the **utility of the keypad and screen are far more valuable**than that.

The bottom line here is this: building a great payment experience is hard. In a test environment, sure, you can build one that's awesome.

But then you put it into the real world—which is full of chaos, noise and employees who don't read instructions—and you notice the gaps.

These UX edge cases can, and likely will, be softened over time. The real question is whether or not the incumbent providers—who currently have no real software—will ever catch up.

In the next chapter we'll look at this software in more detail.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Opening an account

Oct 29, 2020 • 20 minutes

#UX of Payments

-

The card reader

Nov 05, 2020 • 21 minutes

#UX of Payments

-

Building a store

Nov 19, 2020 • 21 minutes

#UX of Payments

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

#UX of Payments • 21 minute read

The card reader

-
- 5 Nov 20
- •
- 21 mins read

By Peter Ramsey

05 November 20

Hardware is notoriously difficult to get right. Once a device has been shipped, there are limitations to what can be changed.

So this chapter explores the physical card reader, and what the experience of setting up this device is really like.

It also compares these readers to new mobile point-of-sale (mPOS) initiatives, such as the Visa 'tap to phone' technology.

> ***Summary:*** *The incumbent provider Verifone performed* ***considerably*** *worse than iZettle, Square and SumUp.*

- 🥒

Why your power-on times matter.

- 💡

  What we can learn from video games.

- 🌍

  Why you should be generous with accessories.

- 🧙

  How removing a click may create a moment of magic.

- 📖

  The problem with instruction manuals.

Case studies

Select a presentation to watch

There are **3** case studies available, you're currently watching 'iZettle'.

There are **3** case studies available, you're currently watching 'Square'.

There are **3** case studies available, you're currently watching 'Sum Up'.

iZettle

iZettle

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

Yeah, understood

-
-
-
-
-
-
-
-
-
-
- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 60

Close

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 80

Close

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 50

Close

Benchmarking the experience

Firstly, how easy was it to connect and activate my card reader? Well, for iZettle, Square and SumUp, this was all done via the app.

Number of clicks to connect card reader

50

58

62

N/A

Clicks

Verifone took a different approach. Because there's no accompanying app, they set up your device for you, before sending it out.

This would have been fine, if it worked.

Verifone sent my device out misconfigured. It was associated to a business called 'Grapes Cafe', and because of this, it was totally unusable.

I assume that this was a rare event, because it took 3 business days and more than 2 hours on the phone with technical support to work out what had happened.

And this highlights why hardware is so problematic; they couldn't debug it remotely—I had to run all the diagnostics myself.

This was exacerbated by the fact that the technical support staff had to jump around between ~**10** different internal portals.

Ultimately, I was left for 5 calendar days without a working device.

The experience

But, at last I had 4 working devices, and could compare the broader experience of connecting and using this hardware.

1. Power on time

Imagine this; you've just opened up your cafe for the day, and have served your first customer. But while reaching for your card reader, you realise that you forgot to turn it on as part of your morning routine.

You both have to wait while the device powers up. How long would this process need to be before small talk dwindled and the embarrassment set in?

So, I tested how long it took for the devices to be ready and ***usable***, after being off for more than 12 hours.

Seconds to power on card reader

3.77

4.11

21.98

63.41

Seconds to power on

iZettle's and SumUp's devices power on more than 5x quicker than Square's, and 15x quicker than Verifone's.

Verifone consistently took **more than a minute** to turn on. Try and fill a minute with small talk, it's really difficult. And even Square's 22 seconds feels like an uncomfortably long amount of time.

And these devices turn themselves off after a certain period of inactivity. So this isn't just a once-a-day issue, but could happen many times.

Time to power off card reader

~4m

~9m

120*

∞

Minutes to auto-power-off

Note: Square's time of 2 hours is what *they* say. I can't test this because it has no screen, and so no way of seeing when it's off—without waking it back up. And Verifone let you change the auto-off settings, but the default value—which most people will stick with forever—is for it to never power off.

This demonstrates two approaches to battery life preservation:

1. Quick start. iZettle and SumUp are so fast to turn on, that they can afford to go to sleep very frequently.

2. Sleep less. Square and Verifone take much longer to turn on, so they sleep _less_frequently.

The point here is that if you infrequently make sales—like an antique furniture store—you might find yourself waiting 22 seconds for *every single payment*.

2. Auto power-on

Often you can look at the gaming industry for cues on how to build better products. For example; one technique used to keep the player entertained while a new area is loading, is to put them in a lift.

The player walks into the lift, presses a button and waits. Unaware that outside those four walls, the game is frantically loading elements for the next scene.

When the game has loaded completely, the lift doors open, and the player is in a new world. As far as they're concerned, this was a realistic simulation of a lift, not a means of hiding a loading process.

You'll see why this is relevant if we delve into the coffee shop example a little deeper. This below would be a typical coffee shop process:

- ☕

  Serving the customer

  i.e., taking order, brewing coffee and handing over drink.

- 📱

  Typing an amount into your keypad

  i.e., inside the application.

- 💳

  Grab / point to the card reader

  i.e., ask the customer to tap their card

- 🎉

  Taps card

  i.e., transaction complete.

It's mundane and works well. But if—like in my scenario—you only realise that the reader is turned off *when you go to make a sale*, then there's an extra step:

- 💳

  Grab / point to the card reader

  i.e., ask the customer to tap their card

- 😳

  Turn card reader on and wait

  i.e., small talk while it turns on.

- 🎉

  Taps card

  i.e., transaction complete.

Impressively, not only do iZettle and SumUp turn on within a few seconds, they do something really smart: **they automatically turn on when you initiate a sale through the app**.

In the period of time between initiating the payment, and grabbing the card reader, it will have turned itself on and probably be ready to use.

In other words; you'll never need to remember to turn your iZettle and SumUp devices on—you might never even notice that they've turned themselves off.

The lift technique hides the loading process of a game, and similarly, turning the devices on automatically right **before** the user needs them, means that they can power on silently before the user is paying attention.

By comparison, Square and Verifone have physical buttons that you need to press on the device—so you'll feel every second of that powering on time, because you're aware of the process that is going on.

3. The problem with small cables

Ideally, you'd charge these devices every night, and then start each day with a full battery. But, it's inevitable that you'll forget every so often.

In those instances, you'll probably need to use the device **while** it's charging, which begs the question; how long is the included charging cable?

Length of the charging cable (cm)

40

82

105

151

170

Length (cm)

The average bar counter is about 100cm tall. For a device that's intended to be placed on a counter, surely you'd make the cable long enough to reach one?

To help you better visualise how short 40cm is, I've included the generic iPhone cable (far left) in the image below.

iPhone, Square, iZettle, SumUp, Verifone

This may not immediately seem like a UX problem, but it is.

If your SumUp device ran out of power during a busy period, you'd have to ask your customers to bend down and use their card 40cm away from a power outlet.

Is this why SumUp also sell a 'charging dock' with a 1m cable? Probably. Could they have included a 1m cable in the box as standard? Yes.

4. The magic of bluetooth

Aside from Verifone, these devices all connect to an app via Bluetooth. This can be an almost zero-click connection process, as shown by SumUp below:

You tap to add a card reader, and it then—somehow—**wakes up your device**, syncs it with your app, installs any updates and it's ready to use.

It worked flawlessly, and the simplicity is a triumph.

Credit where it's due, Square's is very similar to SumUp's.

But by comparison, iZettle's version feels slightly laboured—you need to select **which card reader** you want to connect, ***before*** it'll start searching for any.

All four of the iZettle devices have bluetooth, and the *type* of reader even appears on the 'available readers' screen—so it wouldn't be confusing even if you *did* find multiple types of reader available nearby.

I can't see why they'd use this device-selection screen, unless it's actually just masking some technical limitation.

5. The instruction manual

A study was published in the 1980s, but it's perhaps even more accurate today than it was back then: most **people don't read instruction manuals**.

You see, there are two subconscious assumptions that often make reading these paper booklets feel like a waste of time.

- ⚡

  Assuming the fastest route

  i.e., people assume that they'll guess the fastest route, in less time than it'd take to read the instructions and then do those steps.

- 🧠

Assuming it'll be similar to something else

i.e., they assume that this product will work like other apps that they already know how to use.

So, as a general design principle, your onboarding needs to help guide this impatient user through the journey, assuming that they never bothered reading the instructions.

Which is why I feel like iZettle, Square and SumUp **all** dropped the ball here—connecting your card reader wasn't obvious, there was no onboarding and I genuinely had to go back and check the instruction manual for each of them.

Sure, iZettle and SumUp will prompt you to add one, but only **after** you've attempted to make a card payment—despite it being obvious that you've not connected a card reader yet.

It's a total misfire—it relies on people being curious enough to click on an action that they **know** they shouldn't be able to do.

This is highlighted in the case studies themselves, which also include some of my favourite memes yet.

Conclusion

Aside from the pitiful charging cable, SumUp has certainly been the easiest device to connect.

But **all** the entry-level devices—except Verifone—have an achilles heel; they need to be connected to an app to work. And if your phone or tablet runs out of power, then they're entirely useless.

That is unless you purchase either Square's (£240) or SumUp's (£99) more expensive devices, and iZettle don't even offer any standalone devices right now.

However infrequently you'd forget to charge your card reader overnight, well you actually have to remember to charge *two* devices.

This creates a trade-off:

- ☝️

  1. Standalone device

  i.e., no dependency on a second device.

- ✌️

  2. App and device

  i.e., you can then have the benefits that come with the application layer.

And don't forget, this application layer opens up a whole world of possibilities, and is one of the main reasons why Square, iZettle and SumUp have done so well in the first place.

But what about if you could get the best of both worlds?

This is exactly what Visa are trying to achieve with their tap-to-phone technology, which allows you to accept card payments through the NFC chip in your mobile phone.

And this is a step in the right direction for a few reasons:

1. Lowers barrier to entry. Theoretically, services could charge no up-front fees for someone wanting to take payments.

2. Faster set-up. You could be set-up and able to take payments within minutes, rather than waiting for the device to arrive.

3. Very flexible. Say you were at a weekend event and your card reader died—you could use this as a great back-up plan.

4. Immediately scalable. Do you need 10 more devices for a weekend event? No problem, just ask your staff to bring their mobile phones.

But this technology will not replace physical card readers—at least not immediately,

This is for a number of reasons, but perhaps the most obvious is that these devices can **physically read a card**—i.e., the old fashioned way.

In the long-term, perhaps a 100% contactless payment option won't be a hinderance, but right now it is. Right now we need to be able to insert and swipe our credit cards.

So if one of these companies has partnered with Visa in the near future to offer 'virtual card readers', you've heard it here first.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Opening an account

Oct 29, 2020 • 20 minutes

#UX of Payments

-

Making a payment

Nov 12, 2020 • 19 minutes

#UX of Payments

-

Building a store

Nov 19, 2020 • 21 minutes

#UX of Payments

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

#UX of Payments • 20 minute read

Opening an account

-
- 29 Oct 20
- •
- 20 mins read

By Peter Ramsey

29 October 20

You will have used one of these four payment systems, even if you don't recognise their names. Their purpose is simple: to allow merchants to accept card payments.

But getting a device like this hasn't always been simple. It often required multi-year contracts, and purchasing—or leasing—all the required hardware could cost thousands of dollars each year.

Then, with the growth of cloud computing, Square (2009), iZettle (2010) and SumUp (2012) challenged the payment duopoly that Verifone and Ingenico had built.

I've signed up to all of these devices, and have forensically benchmarked, analysed and compared the experience of each one.

If you're wondering why Ingenico isn't included—they never bothered to return my sales calls or emails. But luckily, their offering is so similar to Verifone, that Verifone can be a proxy for both of the 'incumbent payment systems'.

Right then, let's get stuck in.

> ***Summary:*** *The challenger providers were cheaper, simpler, faster and more efficient. And not just a little bit—considerably so.*

Key takeaways

- 💨

  The negative effects of 'dead air'.

- ⚓

  How SumUp uses payment as an anchor.

- 🧠

  How iZettle helps create 'flow'.

- 🚧

  Building for the subconscious.

- 📪

  Why a delivery company may be letting you down.

- 📦

  A symptom of inefficiency.

Case studies

This chapter only has **3 case studies**, because Verifone's process was entirely manual. Each case study shows:

Select a presentation to watch

There are **3** case studies available, you're currently watching 'iZettle'.

There are **3** case studies available, you're currently watching 'Square'.

There are **3** case studies available, you're currently watching 'Sum Up'.

iZettle

iZettle

Square

Square

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 95

Close

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 94

Close

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 51

Close

Benchmarking the experience

It says a lot about the state of an industry when the incumbents have made it intentionally difficult to compare prices with their competition.

Unlike with Square, SumUp and iZettle, there's no 'Verifone store' where you can purchase a reader. Instead, you need to go through a sales rep to get a quote.

After shipping and taxes, how much did the entry-level readers actually cost me?

Cost of entry level card reader

£22.30

£34.00

£34.00

£79.00

GBP

Square's entry-level device was 34% cheaper than iZettle and SumUp, and a massive 71% cheaper than Verifone.

Or, to make it less abstract; imagine if Verifone had a 70% off sale—it'd still be more expensive than Square.

But, you should be aware that these devices aren't perfectly comparable. For example, Square's device doesn't have a screen, and none of the entry-level challenger devices will print paper receipts.

This comparison is of the minimum spend required to get *something* set-up.

After I'd bought all these devices, I then had to wait for them to arrive.

Number of days for the reader to arrive

1

2

2

5

Number of working days

Okay, so the challengers were faster and cheaper. But what about the actual ***experience*** of signing up?

The experience of signing up

The above benchmark set the tone: Verifone is more expensive and took longer to arrive, but ultimately they're just two metrics.

The rest of this chapter explores what the user experience of actually joining each service is like.

1. Dead air

I have a confession; claiming that Verifone took 5 working days to arrive is misleading. It actually took **10 working days**.

This is because the Verifone sign-up process is entirely manual. You can't sign up online, but instead you go back and forth with a sales rep over email.

And no, it wasn't even a Docusign, it was literally a PDF that I had to annotate my signature into, using Mac Preview.

The full process went like this:

- 👋

  Enquiring on their website

  Hey, I need this.

- 🙇‍♀️

  Discussing requirements with sales rep

Answered 34 questions in an email, and sent photo ID.

- 📝

  Complete contract and Direct Debit

  Then return via email.

- 📦

  Device arrives in the post

  5 working days later.

If the above process doesn't sound frustrating to you, consider that they'd only reply during business hours, and this elapsed two weekends.

That's 4 days of dead air. 4 days where I could have just signed up to iZettle, Square or SumUp.

To clarify, all 3 of the challengers had entirely digital processes. Within 5 minutes you could have any of these accounts, and an email confirming that your device will be with you in a few days.

Sure, if Wetherspoons approaches you and asks for 10,000 terminals, then a sales rep would be helpful to iron out logistical issues. But for most small businesses, that kind of high-touch process is crippling.

2. Payment as an anchor

There's a technique often used in sign-up processes; **creating anchors of investment**.

In short, an anchor refers to a point in the process that, once passed, the user would be incredibly unlikely to give up on what they were doing.

Anchors help keep bored users engaged.

These anchors are very subtle, and often unintentional. But here are 3 common types that you've probably experienced before:

- 💵

  Payment as an anchor

  i.e., once you've paid for Netflix, it wouldn't matter how long their sign-up process was, you'd still use it.

- ⏰

  Time as an anchor

  i.e., you've already spent 45 minutes answering questions on a form, so you'd have a lot to lose by quitting.

- 🏋

  Effort as an anchor

  i.e., you've already done the hard/boring part of a process, so you view future tasks as being easy.

This creates a dilemma: what's the most effective anchor, and how do you get your users to reach that as soon as possible?

In this instance, one of these anchors is obvious. Let's look at **how long into the process** (as a percentage) you are, when you're asked to **pay for your card reader**.

When do you pay for the card reader (% of process)

18%

66%

99%

N/A

Clicks as a % of total clicks required

There are clearly two approaches:

1. Pay early. SumUp get the user to pay for the device almost immediately (you actually select your reader after 4 clicks).

2. Pay late. iZettle and Square use 'effort as an anchor', and paying for your reader is basically the final step.

And then you have Verifone, who are essentially anchor-less—drifting around on the ocean during weekends.

3. Inducing a state of 'flow'

The mantra "_time flies when you're having fun_" may be true, but why does time also fly when I'm sat answering emails?

Well, partly, that can be attributed to being in a **state of flow**, which is basically just a period of total focus.

Inducing your users into a state of flow is the holy grail for software companies.

And although it's not easy, or even predictable, you can incubate a suitable environment for flow with a few considerations:

- 🚦

The task has the correct difficulty

i.e., not too easy, and not too hard.

- ⛰️

  A sense of progression

  i.e., progress makes the user feel like they're good at the task.

- 🎯

  A lack of distractions

  i.e., let the user get on with it.

Ignoring Verifone, which of these services created the best environment for flow?

iZettle

Square

Sum Up

To me, it's clear that iZettle is by far the best here. The case studies go into detail, but in short; it's more consistent, fewer distractions and has a real sense of progress.

4. Signposting

The subconscious mind is a wonderful thing. It processes subtleties in our environment, and silently helps us navigate the world.

For example, if you were to approach the door shown below, you'd subconsciously decide that this is a push door, and not a pull one.

Push and pull doors *look* different to each other—a design choice intended to encourage your brain to engage in autopilot. And it works.

This technique is used in the digital world too. There are elements and components which you may not pay much attention to, but act as signposts for your subconscious.

Think about this; why—during a form, or sign up process—do you sometimes see pages like the one below, between sections?

It's brilliant and subtle—these pages act as a bookend. They tell you that a section has finished, and that the **context** around the questions will now change.

This gives the user a chance to reorient themselves, and refocus their attention on the new context.

Or to clarify the iZettle example above: “you’ve answered everything about yourself, so from now, all the questions will be about **your company**“.

Nobody else utilised this technique, and their experiences jump around between asking about your personal details, to your billing address, to your registered business address.

5. Choice of delivery company

Before Amazon ran their own delivery service, they’d use third parties companies.

One of these was so complained about—particularly referring to damaged or missing items—that it made mainstream media in the U.K.

I, like many others, blamed Amazon. I knew it wasn’t their fault directly, but I blamed them for their choice of partners.

It’s important to remember that the user experience of a product or service transcends pixels on a screen. And, like Amazon learned, it certainly includes any services provided by third parties.

As a company shipping physical products, **your choice of parter is not trivial**. You’re trusting them to do a great job. If they fail, you fail.

So I was interested to see which delivery company each service opted for.

iZettle

Square

Square

Verifone

DPD

DPD

Yodel

TNT

Tracking ✅

Tracking ✅

Tracking ✅

No tracking ❌

Fortunately for this comparison, MoneySavingExpert.com ran a survey of nearly 10,000 consumers in the U.K., earlier this year, and then ranked the major delivery companies from best to worst.

Did SumUp and Verifone use the two worst-voted delivery companies because they didn’t do enough research on them? Or because they were cheaper?

If using one of the least-favourable delivery companies wasn't bad enough, Verifone didn't even pass along a tracking number, so my parcel just randomly arrived one day, while I was out.

Had I not installed a doorbell camera, I'd have missed the delivery, and had to wait another day at least.

Sending a parcel that requires a signature, without any warning or tracking number, is a form of passive torture.

6. A symptom of inefficiency

Finally, I've got an even more abstract UX consideration: how efficient is their packaging.

Aside from the obvious environmental issues—which consumers are generally becoming more aware of, and frustrated by—inefficient packaging is a **symptom of a broader inefficiency**.

iZettle, Square and SumUp all came in small boxes, and the experience felt similar to opening the box of an iPhone.

But then my Verifone device arrived.

I know it looks big, but sometimes images can be deceiving, so I measured their volume. Square came in at (580cm³), compared to Verifone's (15,300cm³).

Let that sink in; the Verifone box is more than 26x larger.

Sure, the Verifone device is bigger and it comes with more stuff—which I'll be discussing in the next chapter—but the size of the box is **entirely unjustified**.

But why does this matter? Well, it's all about first impressions.

Looking at the image above, do you see a company that has relentlessly obsessed over optimising their processes?

Do you see a company that is constantly trying to reduce their shipping costs, and therefore make their devices more affordable for the end user?

Conclusion

Let's get the obvious point out of the way: the sign up experience of Verifone is clearly worse than all three of the challenger providers.

But, which of the 3 was best?

In my opinion iZettle strikes the best balance of focus and simplicity. It was organised and felt great.

But, compared to the experience of *using* these devices every day, the sign up processes is a small blip in your overall experience.

The next 3 chapters will really put these devices to the test.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

The card reader

Nov 05, 2020 • 21 minutes

#UX of Payments

-

Making a payment

Nov 12, 2020 • 19 minutes

#UX of Payments

-

Building a store

Nov 19, 2020 • 21 minutes

#UX of Payments

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking

#Payments • 18 minute read

How Klarna Works

-
- 17 Sep 20
- •
- 18 mins read

By Peter Ramsey

17 September 20

Klarna is one of the biggest 'buy-now, pay-later' services in the world, and you'll probably have seen their logo in a few familiar places, like ASOS, Topshop and Wayfair.

The idea is that you can buy something from one of their participating retailers, receive the item, and pay for it through Klarna later on.

And it seems to be working well. As I write this, Klarna is a $10 billion company, and they claim to process more than a million transactions every day.

So, you can imagine how meaningful a 1% improvement in their customer retention would be. Even 0.1% would be noticeable.

Well, let's explore the UX mistakes that Klarna are making, and ultimately how they could improve their product fairly easily.

Summary

- 🎯

  Recognising event-driven behaviour.

- 🍯

  How not to ask 'are you sure?'.

- ⛳

  The irony of saying something is 'easy'.

- 🗺️

  Always include a way back.

- 💬

  Avoid content echos.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 107

Close

5 UX takeaways

1. Event-driven behaviour

Sometimes—as I'm making my morning coffee—I'll realise that we've run out of milk. So I rush down to my local shop, power-walking through the isles at 7am with a bottle of milk in hand.

It's a very small shop, so the cashier will have seen me pull up, and use the most efficient path to go from the door, to the milk, to the till—and have Apple Pay up and ready before they open their mouth:

> *"Do you have a Tesco clubcard?"*
>
> *"_No._" (slightly out of breath)*
>
> *"Would you like to sign up **now**?"*

Sure, it may be company policy to always ask, but it's a stupid company policy.

It's also a perfect example of an issue that happens a lot when designing software: people misunderstanding, or not considering what is driving a user's behaviour.

I've overly-simplified this, but consider that there are only two types of behaviour:

- 🎯

  1. Event-driven behaviour

  People have a clear purpose or goal.

- 🤷‍♀️

  2. Casual browsing

  Open to being tempted by offers and new products.

Klarna have a version of this when paying for your *first* item, after you click on a CTA that is very event-drive: "*I'm ready to pay*".

As this is your first experience of Klarna, you're probably anxious to see if there will be any additional fees. You want to make the payment as quickly as possible, and be reassured that you didn't make a mistake trusting Klarna in the first place.

Well, after clicking on that CTA, you're immediately asked to invest your time into creating better future shopping experiences. This is their 'clubcard' moment.

You click "I'm ready to pay"...

Introducing: temptation

The point here isn't that Klarna shouldn't be asking users to invest their time—it's that right now, in this moment, the user is focused on a task.

And if you're trying to get a user's attention, to ultimately invest time into something irrelevant, then you'll have much more success doing it ***after*** they've completed that task.

Or rather: if Klarna asked this same question after the payment had been made, I'd be confident that they'd increase the engagement they received.

2. "Are you sure"?

Imagine if my Tesco scenario actually went like this:

> *"Do you have a Tesco clubcard?"*
>
> *"_No._"*
>
> *"Would you like to sign up now?"*
>
> *"No thanks."*
>
> *"_**Are you sure?****"_*
>
> *"Yes"*
>
> *"**But a**re you sure?****"*

Let's be honest, this would be insane, yet it's exactly what Klarna do. Actually, they ask if you're sure **four times** in a row.

Skip

Yes, skip

Skip

Yes, skip

There are two problems with what Klarna are doing here:

- 👌

  1. Offering no new information

  Just asking the same question again isn't convincing—you need to entice the user into changing their mind with some new information, or a benefit that they may not have correctly valued before.

- 🙄

  2. Making you skip each step individually

  If you want to skip the first part of the personalisation process, you probably also want to skip all the following parts too.

It's actually quite helpful to use my analogy again:

It may be convincing enough if the Tesco employee responded with new information, for example:

> *"Are you sure? All I need is your email address and you'll get a £2 bonus instantly."*

In conversation, this is mostly instinctive. If someone says no to your offer, you immediately know that you need to entice them a little more. Even children understand this.

But in software, human instinct needs to be designed. It rarely happens by coincidence.

3. The irony of telling people how easy something is

Telling someone how easy something is—without quantifying it in any way—is in short; useless.

There's a deep irony too: companies believe their own products are easy to use, so they tell people how easy they are to use. In doing so, adding stuff for the user to read, making it take longer.

The key is to not just say something is easy, but explain why it's easy. Otherwise it's nothing more than padding.

For example:

❌ **Useless comments**

✅ **Good alternatives**

It's really easy.

All you need is an email address—that's how easy it is.

It's really quick to do.

90% of our users complete this process within 75 seconds.

Klarna have a great example of this: telling the user **"It's a breeze"**.

If this seems tiny and insignificant, that's because it is. But it's also easily solvable, and a missed opportunity to say something that's actually interesting.

Great product design is a balancing act of being as concise as possible, whilst also being as convincing as possible.

4. People will enter their details incorrectly

You've probably heard the quote: *nothing is certain, except death and taxes*.

Really though, this needs to be rewritten for 2020:

> *"Nothing is certain, except death, taxes,* ***and that customers will enter their personal details incorrectly on your form****".*

When designing software, in particular forms, you need to make sure that users can go back and edit their data—without having to just start again—because they will get it wrong some of the time.

Which is why verifying your phone number with Klarna is so frustrating—you can't go back. You have to request a new email, get a new link, and start again.

Enter number

Realise it's wrong

Can't go back

This is probably an oversight, but it’s such a major inconvenience when it does happen.

5. Content echos

I’m a big fan of helpful text: the small—usually grey—secondary line of text that gives context to a title.

But sometimes people push it too far, and really there’s nothing relevant extra to say. For example:

It’s a harmless line of text. The designer probably wasn’t overly-happy with putting it in, and knew it was a waste of time, but it made the aesthetic design work better.

But here’s the thing: subconsciously people will start to notice that some of the content isn’t very helpful, and so they stop paying attention to it.

The risk of the content echo is not the echo itself, but that it teaches the user that sometimes the helpful text isn’t actually helpful at all.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

Buying a card on Moonpig

Dec 11, 2020 • 18 minutes

#Purchasing

-

Buying a domain on GoDaddy

Mar 05, 2020 • 7 minutes

#Domains

-

How creating an event on Eventbrite works

Feb 21, 2020 • 4 minutes

#Purchasing

-

Signing up to Cash App

2 months ago • 20 minutes

#Payments

#Investing • 15 minute read

How Coinbase works

-
- 19 Aug 20
- •
- 15 mins read

By Peter Ramsey

19 August 20

From a user experience perspective, buying, storing, trading and selling cryptocurrency—like Bitcoin—is still a pretty rough experience.

There's a steep learning curve, terminology to understand and a regular supply of horror stories about hackers stealing cryptocurrency to make the experience uncomfortable for most.

Coinbase is on a mission to change that.

They want to make buying cryptocurrency feel safer and simpler, in the hope that it entices new investors, who would otherwise be put off by the idea of managing their own digital wallet.

But how could the user experience be better? Is it good enough?

Summary

- 🐎

  Avoiding the ‘get started’ trap.

- 🤦

  Why you need a feedback architect.

- 👨

  Be aware of the header.

- **Usability tip:**

  The mobile experience is much better in fullscreen.

  Go fullscreen

  You can navigate the slides using your keyboard.

  Yeah, understood

-
-
-

- **Subscribe to the newsletter and never miss a future case study.**

  Free UX newsletter trusted by product-lovers.

  Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Or, share this article and help the mission

Copy

Link Copied!

Slide 1 of 95

Close

3 UX takeaways

1. The 'get started' trap

The Coinbase app really doesn't have any onboarding. But, they do have something that I'll refer to as the 'get started' trap.

A 'get started' trap works like this:

- 👮‍♀️

  1. Lack of onboarding

  The user is left wondering what to do.

- ☝️

  2. Tempting action that promises 'help'

  A CTA that says something like 'get started' or 'learn more'.

- 🤯

  3. Pushes you in the deep end

  The CTA doesn't offer guidance at all, and just starts an action. i.e., making a payment or choosing a monthly subscription.

In effect, this is where something promises a helping hand, but is nothing more than a shortcut to doing what the *company* wants you to do.

After signing up to Coinbase—and verifying your ID—you're greeted with this very tempting 'Get started' button.

But clicking it doesn't offer any help at all. Instead, it immediately asks you how much Bitcoin you'd like to buy. And, it defaults to Bitcoin—removing the choice about which cryptocurrency you're interested in.

This doesn't help you get started with Coinbase, and the CTA is nothing more than an illusion—an onboarding trojan horse.

The key takeaway here is that Coinbase's users probably *do* want to buy some cryptocurrency—that's why they signed up.

But they also want to **understand how Coinbase works**, and to be sure that it's the platform for them, before entering their card details.

If Coinbase is going to keep capturing new users, they'll need to appeal to people who haven't already decided that they're ready to purchase Bitcoin.

2. Feedback architecture

In the physical world, you almost always get feedback which lets you know if something worked or not—without any additional effort from that item's creators.

For example, if you used a faulty vending machine, which didn't dispense your drink, you'd know pretty quickly. You wouldn't need to see an error message—the lack of drink ***is*** your error message.

But with software, someone needs to be the architect of this feedback. These error messages need to be created and implemented.

When these moments haven't been implemented properly, you end up with a confusing and broken experience. This is exactly what happens when adding a new card to your account on Coinbase.

Click 'add card':

Then:

You add a card, see a spinner and then get put on the homepage—without any attempt to tell you if adding your card was successful or not.

I don't need to explain *why* this is bad UX, but rather I can point to a few reasons why this type of mistake is so common:

- 📋

  1. Volume of possible states

  The number of possible outcomes for a medium-sized SAAS product is enormous, and it grows with every new feature.

- 🔬

  2. Inadequate testing

  Maybe you can add your card in 4 different places, but only 3 of these journeys were tested at the last deployment.

- 🤷

3. Unclear responsibility

This types of UX mistake will often slip through ‘job role’ cracks. Is it the designer’s job? The product manager’s? The developer’s?

3. No surprises, please

As a designer, I always aspired to create something unique. Perhaps a new homepage layout that nobody had seen before (and that didn’t just copy Stripe).

But, the truth is: designing the best experience isn’t about being unique, it’s about being easy.

And guess what feels really easy to use? Things that feel **familiar**.

As an example; which of the two designs below do you feel is easier to use?

Option A:

Option B:

Objectively, I *could* put forward a half-reasonable argument why the second option is okay. (It’s not better, but I’ve used an extreme example on purpose).

The point is, we’re so used to keyboards being at the bottom of the page, that any deviation from this design would be confusing, a shock and very difficult for people to come to terms with.

These rules are often unspoken, and mostly subconscious. But, here’s one that I think you’ll appreciate: almost every app screen can be split into two sections:

- 🥳

  1. The header

  This is where predictable stuff goes. Like the company logo, back button, page title, breadcrumbs and the device settings (battery , signal etc.).

- 🍯

  2. The actual page

  This is the bit that I should pay most attention to.

This isn’t true all the time, but it’s true often enough that when people use apps they don’t typically even read what’s in the header.

Which is where Coinbase make a mistake: the only way you’d know that you were buying Bitcoin on this page, is if you read the page title—which is **in the header**.

The natural first-viewing of this screen is messy, because people try to work out what they’re doing from the body of the content, and eventually work their way up.

I’m not suggesting that you can’t put things in the header—not at all—rather, that the main body of the page should also make it clear what you’re doing.

Researched and written By Peter Ramsey

Join others who get regular Built for Mars content like this delivered into their inbox.

Learn more about the newsletter

Leave this field blank

You're now subscribed to the Built for Mars newsletter!

Email me future studies (privacy).

Email

Subscribe

Similar Case Studies

-

How Robinhood handled a crisis

Feb 01, 2021 • 13 minutes

#Investing

-

How Freetrade works

Jul 21, 2020 • 19 minutes

#Investing

-

Selling an NFT on OpenSea

6 months ago • 14 minutes

#NFT

-

Monzo vs Starling: Overdrafts

17 days ago • 18 minutes

#Banking