The Basics of Design Sprints and Other Jargon

Posted 2 years ago by Graeme Fulton

Blog

Nowadays, more and more product teams are adopting Agile processes to manage projects and create more user-centric products. As a result, we've seen a bunch of new jargon pop up alongside the various design methodologies that emerged to help teams bring design to Agile development.

> “From Design Sprints to Agile UX, this is my attempt at breaking down industry jargon into something simpler.”

From Design Sprints to Agile UX, the aim has been to help teams bring design and development together to ensure Agile processes deliver an end product that looks and works just like the designer intended. All the different frameworks out there can get a bit overwhelming, so this is my attempt at breaking down the jargon into something more straightforward and understandable.

What we’ll cover:

We’ll cover the 4 methodologies outlined in the list below, looking at why each is used and how they improve collaboration with Agile development teams. For a summary of the collaboration challenges, check out my previous article. For now though, these are the frameworks and methodologies we’ll cover:

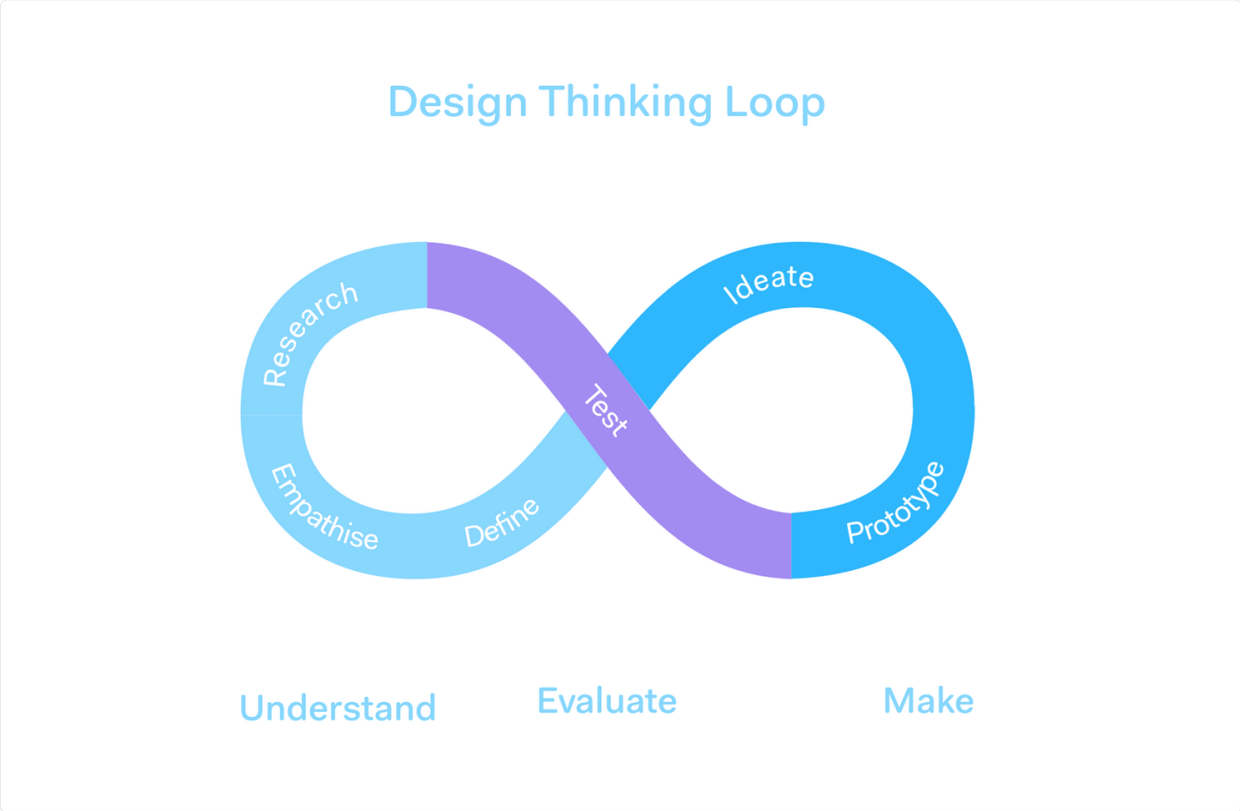
1. **Design Thinking**
2. **Lean UX**
3. **Agile UX**
4. **GV Design Sprints**

• • •

1. Design Thinking

What is it?

Design thinking is a framework to help organisations think more like designers through the use of elements from a designers toolkit such as *empathy and experimentation*. It blends what's technically feasible with what users actually want in a product, helping companies change the way their products are built.



Design Thinking Loop

Having your development team understand what design is by practicing it in the early stages of a project can help bridge the gap between design, development and other departments. The different steps involved can be seen in the diagram above, but for a more detailed overview check out IDEO's page.

> "Design Thinking helps make design more inclusive, providing the tools for anyone to think like a

designer."

Design Thinking and Agile collaboration

With so many teams embracing Agile for development, how can Design Thinking work alongside it? Here are 3 ways to go about it:

1. Prototype first

Prototyping ideas in Design Thinking workshops can also save Agile teams from developing features that aren't actually needed. Once the team understands their target user, they can start using feedback from that user to prioritise the features that will get delivered through Agile sprints. The most critical features can be focused on first and iterated upon quickly, whilst progress can be shared transparently with the rest of the team.

2. Focus on the Similarities

In an ideal world, teams using Design Thinking converge and collaborate really well. However, in truth, getting people from different departments to focus on Design Thinking exercises can sometimes prove a challenge, as you can get people who simply don't want to be there. There may be developers who just want to jump straight into coding and feel like the process is a waste of time. This is also suggested by Steve Perkins, Culture Strategist for Capital One, who writes that the main concern of working with Agile and Design Thinking can be the impetus to 'just start building', rather than focusing on the solution first.

However, forcing people to use post-it notes and draw out storyboards won't influence an engineering culture. A better approach can be to focus on the similarities between Agile and Design Thinking to bring them together. For instance, Perkins also suggests that Design Thinking can be performed in an Agile manner, whilst at the same time, Agile can be performed with a Design Thinking mindset. This is possible because they both compliment each other's philosophy in a few areas – some of which are highlighted by **Shay Peleg**, Sr. Agile Coach at IBM:

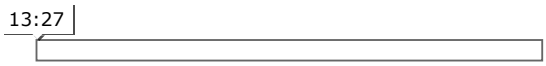
- The customer is at the centre of both.
- We're encouraged to prototype, and create a high quality representation of the final product without running through the entire development lifecycle.
- Both prevent you from falling in love with one final solution, as both are iterative, learning from previous cycles.
- Design Thinking promotes a multidisciplinary and diverse team to look at problems from various perspectives – this is similar to the concept of autonomous, self powered teams in Agile.

Check out the full list in Shay's article, *'Design Thinking = Agile?'*

3. Lead with design

Design Thinking may be a useful tool, but what happens to it once the workshops are over and the engineering begins? Empathy maps and personas can end up left behind and forgotten, as if they were just boxes to be checked off. Natasha Jen suggests this in "Design Thinking is Bullshit" 💩, questioning whether the practice actually effects end products:

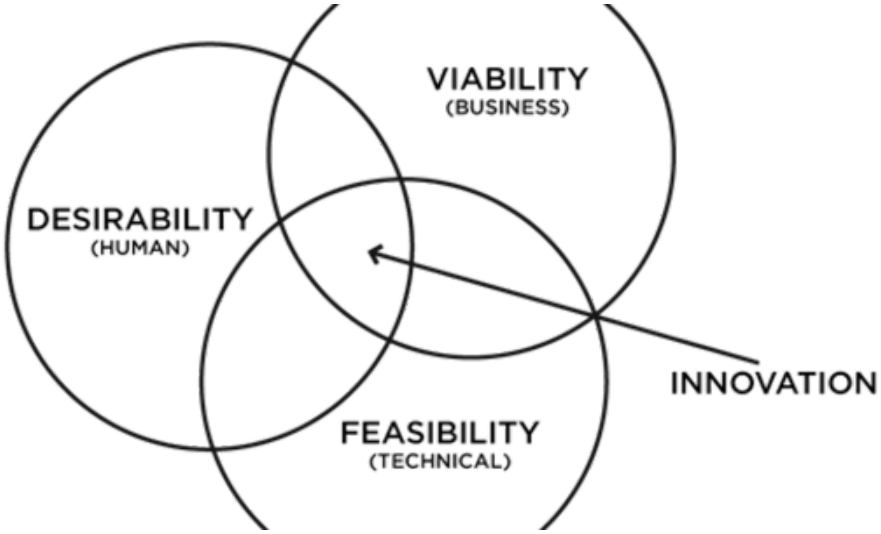
> "Design Thinking may be useful, but what happens once the workshops are over and the engineering begins?"



In my experience, these issues can emerge when there's an absence of skilled designers leading Design Thinking sessions, and consultants with less experience are used in place of designers to facilitate sessions. In these situations, the Design Thinking framework can end up being used as a golden path, that gets followed step by step. Despite this aspect, as a tool to spread design across an engineering lead business, there haven't been many better approaches.

Useful Resources:

MEDIUM MATT COOPER-WRIGHT



The blurring between Design Thinking and Agile – Front Line Interaction Design – Medium

"Hi Greg Firstly, totally agree with Juho, Kam and Peter 's thoughts. At IDEO Design Thinking lives in the strategic world where we use design methods to find the right question and begin to answer it. Agile is lives in the software world where once a question is asked teams iterate toward a solution.

Read the article on medium.com >

IDEO | DESIGN THINKING

Design Thinking | Design Thinking

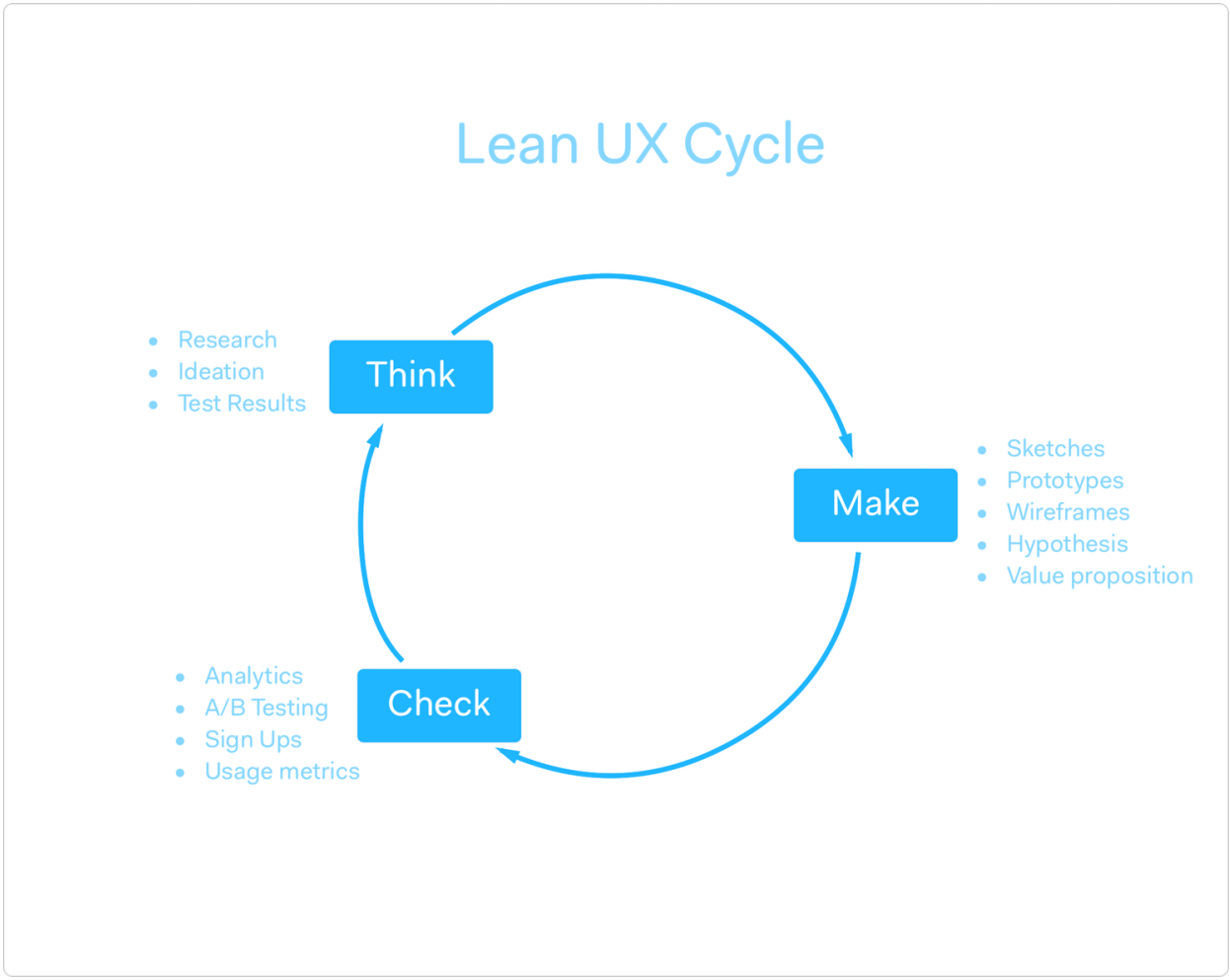
What is design thinking? Design thinking is a human-centered approach to innovation that draws from the designer's toolkit.

Read the article on designthinking.ideo.com >

2. Lean UX

What is it?

Lean UX is a set of principles based on the agile methods of the 'Lean Startup', where much of the focus is brought to the present. Instead of throwing design deliverables over the wall, teams accept that a final design can't be created up front, and believe the answers will emerge as hypothesises are tested through incremental Minimal Viable Products (MVPs). For the sake of understanding, it can be pictured as this continuous 'build, measure and learn' loop, or a think, make check cycle:



Lean UX Cycle

Like Design Thinking, Lean UX isn't a process where each tool has to be applied in a specific order – it's a process that helps you reduce waste, and work together to build a customer centric solution. For a more in-depth overview of Lean UX, check out

SAFe's article, who outline all the main principles such as 'The Lean UX Process', 'Outcome Hypothesis', and the 'Minimal Marketable Feature'.

Lean UX and Agile

Lean UX works well with Agile because they both operate iteratively, enabling teams to apply learnings from each cycle to the next iteration. As Interaction Design Foundation put it:

> *"The nature of Agile development is to work in rapid, iterative cycles and Lean UX mimics these cycles to ensure that data generated can be used in each iteration."*

Here are 3 ways to go about using Lean UX alongside an Agile Development team:

1. Extend the role of the designer

Lean UX brings all the different departments in an organisation together, extending the role of the designer past a simple hand off point, and towards understanding what's actually required when it comes to implementing a design.

This encouragement for designers to understand how developers work not only improves collaboration but can also reduce waste. An understanding of the technical constraints in which designs are built can lead to more realistic and feasible designs that can actually be implemented by engineers.

2. Empower everyone to design

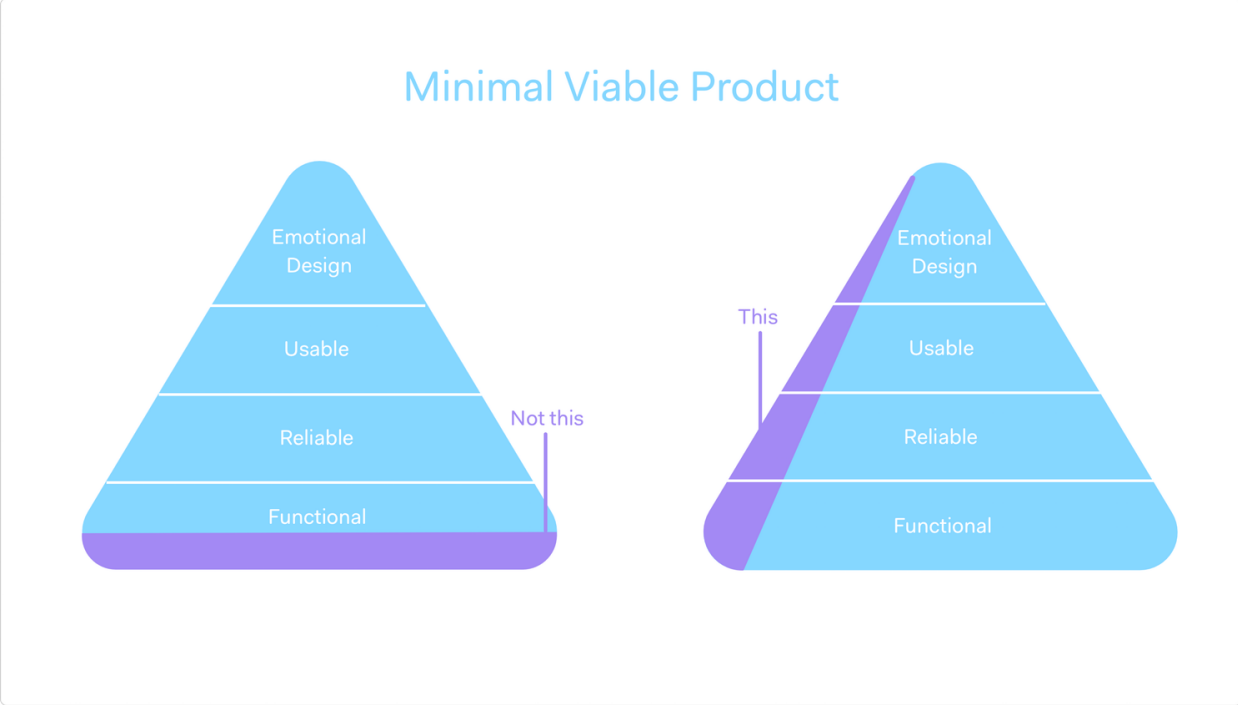
Just like designers working with code, the converse is true for developers, who get involved in design where possible and increase their understanding of UX. As highlighted by the Interaction Design Foundation, this sharing of design responsibilities not only makes sure we research, design and build altogether, but also

ensures there's no single source of design that may lead to a bottleneck for development:

> *there's no "bottleneck" created by having a single UX design resource trying to get the whole job done in tight timescales by themselves.*

By having departments feedback to each other more often in Lean UX, everyone involved with the project works more collaboratively from inception to implementation. This doesn't mean different teams won't have their own space though – there are still times to get things done individually as a designer or developer.

3. Fail fast with MVPs



Minimal Viable Product, adapted from Jussi Pasanen

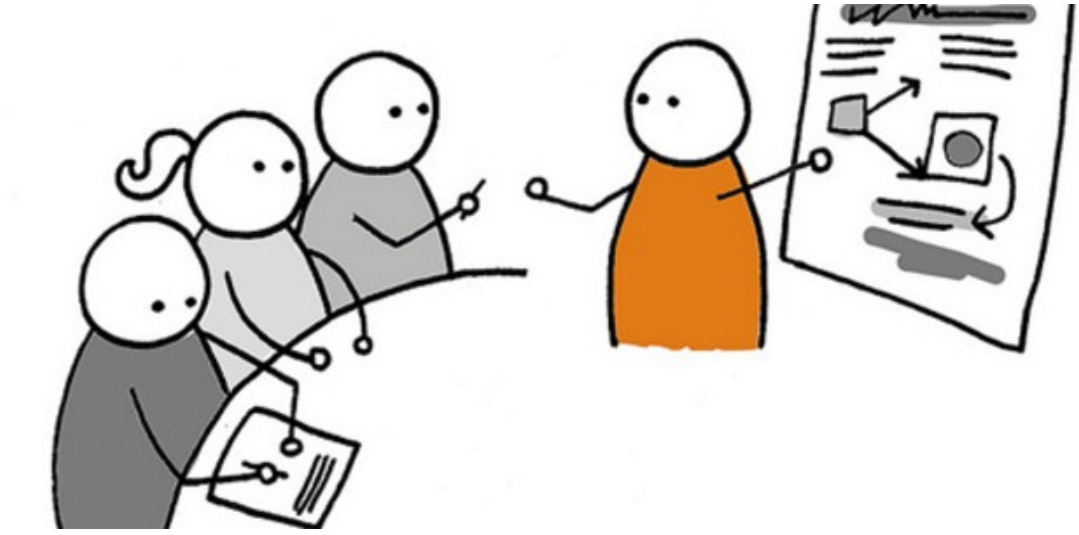
A core concept of Lean UX is the Minimal Viable Product (MVP), which is the simplest and cheapest working version of a product. The use of MVPs enable product teams to throw features away that fail the initial hypothesis, enabling more experimentation – which is also key to agile.

> “MVPs allow us to remain customer-centric, save us from marrying ideas, and ensures there are no ‘sacred cows’.”

MVPs allow us to remain customer centric, saving us from marrying ideas, and ensuring there’s no “sacred cows”.

Useful Resources:

SMASHING MAGAZINE



The Lean UX Manifesto: Principle-Driven Design

My colleague Ajay and I have been working at incorporating lean UX at the enterprise level for over two years. In studying it, I find that there’s a temptation to lay down rules, and if the rules aren’t followed... well, then, you can’t call it lean UX.

Read the article on smashingmagazine.com >

SCALEDAGILEFRAMEWORK

Lean UX

What if we found ourselves building something that nobody wanted? In that case, what did it matter if we did it on time and on budget? [1] -Eric Ries Lean User Experience (Lean UX) design is a mindset, a culture, and a process that embraces Lean-Agile methods.

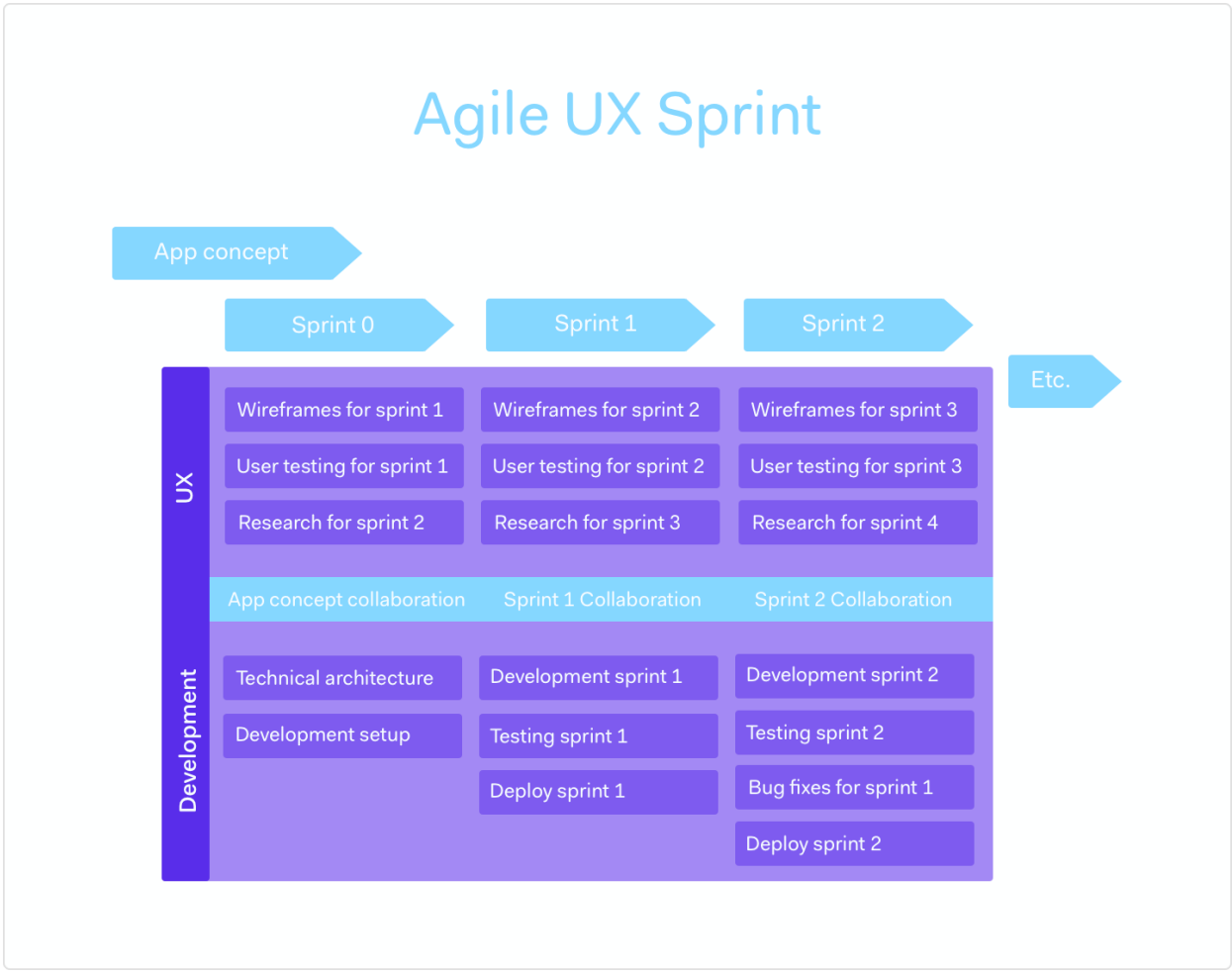
Read the article on scaledagileframework.com >

. . .

3. Agile UX

What is it?

Agile UX builds upon the concepts of Agile development by adding elements from User Experience design into the process. In contrast to Lean UX bringing departments together, Agile UX focuses on bringing design *into* the Agile process. This means design is actually carried out using sprints, scrums and retrospectives – the same framework a development team will be using. An example of an Agile UX alongside development may look like this:



Example design sprint, adapted from White Space

Agile UX is iterative, so can prevent big upfront designs being handed over to development teams. Therefore, designers are less likely to go off on their own, and design in isolation to feedback from the rest of the team.

> "Agile UX is iterative, so can prevent big upfront designs being handed over to development teams."

There are a few ways to go about Agile UX, with 2 of the most effective as follows:

1. Design within sprints

Design can be carried out in the very same Sprint as development work, so that design and development are always working on the same thing. This is also known as 'Just in Time' agile UX, as the designs are created in tandem with what's being developed. Using this approach can strengthin the relationship within teams, as high levels of collaboration are needed. Conversely though, the constant communication needed can slow development speed. Derek Larson frames this well:

> *It's like two bridge construction teams building from opposite shores; constant communication is critical to make sure when they meet in the middle they aren't yards apart.*

This is also why many product teams operate with UX working 1 sprint ahead of development instead. Let's now look at this approach:

2. Work ahead of developers

Loranger suggests that UX working ahead of development sprints is preferred by many teams, as it ensures for more time to carry out tasks such as research and user testing. This extra time enables designers to test even more assumptions, and have clear designs ready to pass to development for *their* next sprint. In return, developers can feed back working code for designers to test with users.

More Benefits of Agile UX

Overall, Agile UX shares many of the benefits of Lean UX, plus a few of its own:

- Agile UX is incremental, preventing big up front designs handed to development teams
- Both design and development are working with the same methodology, so can work closer together and have a shared understanding.
- By staying one sprint ahead of development, product decisions can be made ahead of implementation, guiding the software development process.

- Designers can better integrate user feedback in each sprint, as they work closer to the rest of the team.

Whilst there are clearly many benefits, one major drawback is that the quality of user research can often be sacrificed in order for design to stay ahead of each Agile Sprint cycle.

Useful Resources:

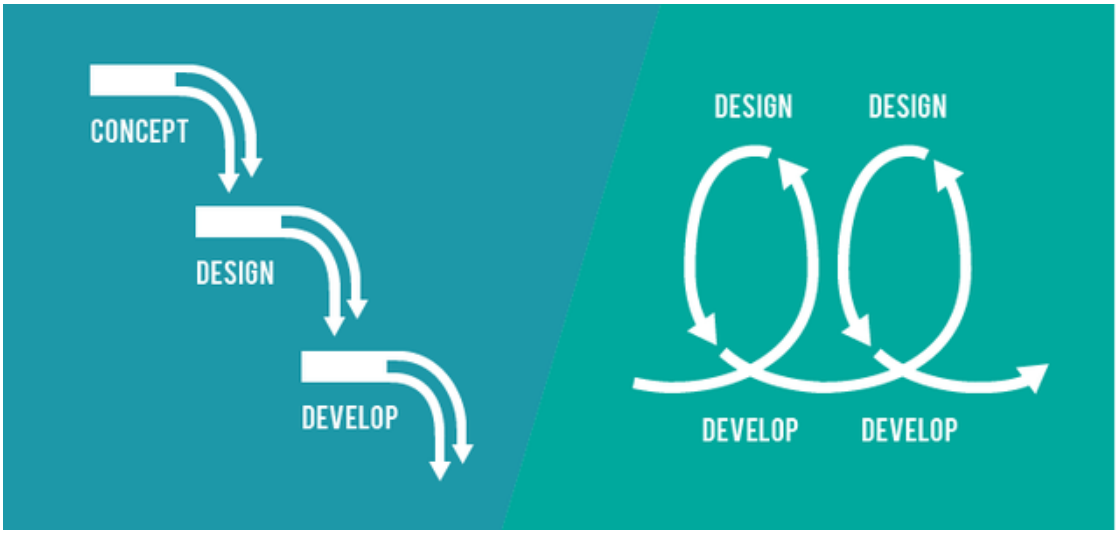
NN NIELSEN NORMAN GROUP

Infusing UX to Agile Development Processes

New research shows UX making strides in infusing user-centered design approaches into Agile development processes.

Read the article on nngroup.com >

MEDIUM LUCA LONGO

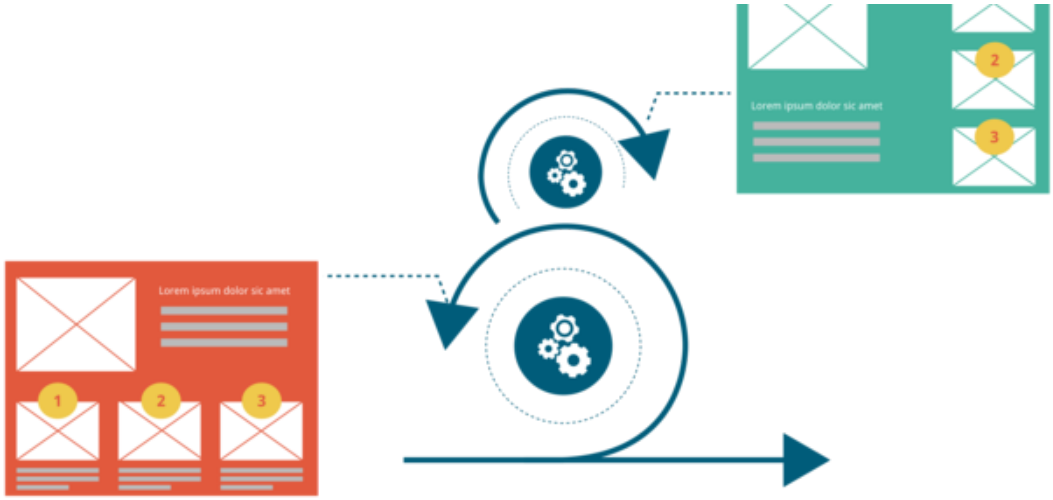


How Providing UX in Agile: Design Lead and Product Owner

By Luca Longo Today, I'll talk about my experience working as UX Lead in an agile company. Just to make sure, I'm not going to talk about using agile during your design process, Google already published an excellent document about that.

Read the article on uxplanet.org >

MANIFESTO JIM BOWES



How to create an Agile UX design process – Manifesto

Agile and UX seem like natural bedfellows but many organisations who've successfully got their development teams to adopt Agile methodologies are struggling to integrate UX design into the process. The principles of Agile were set out by software developers for software developers.

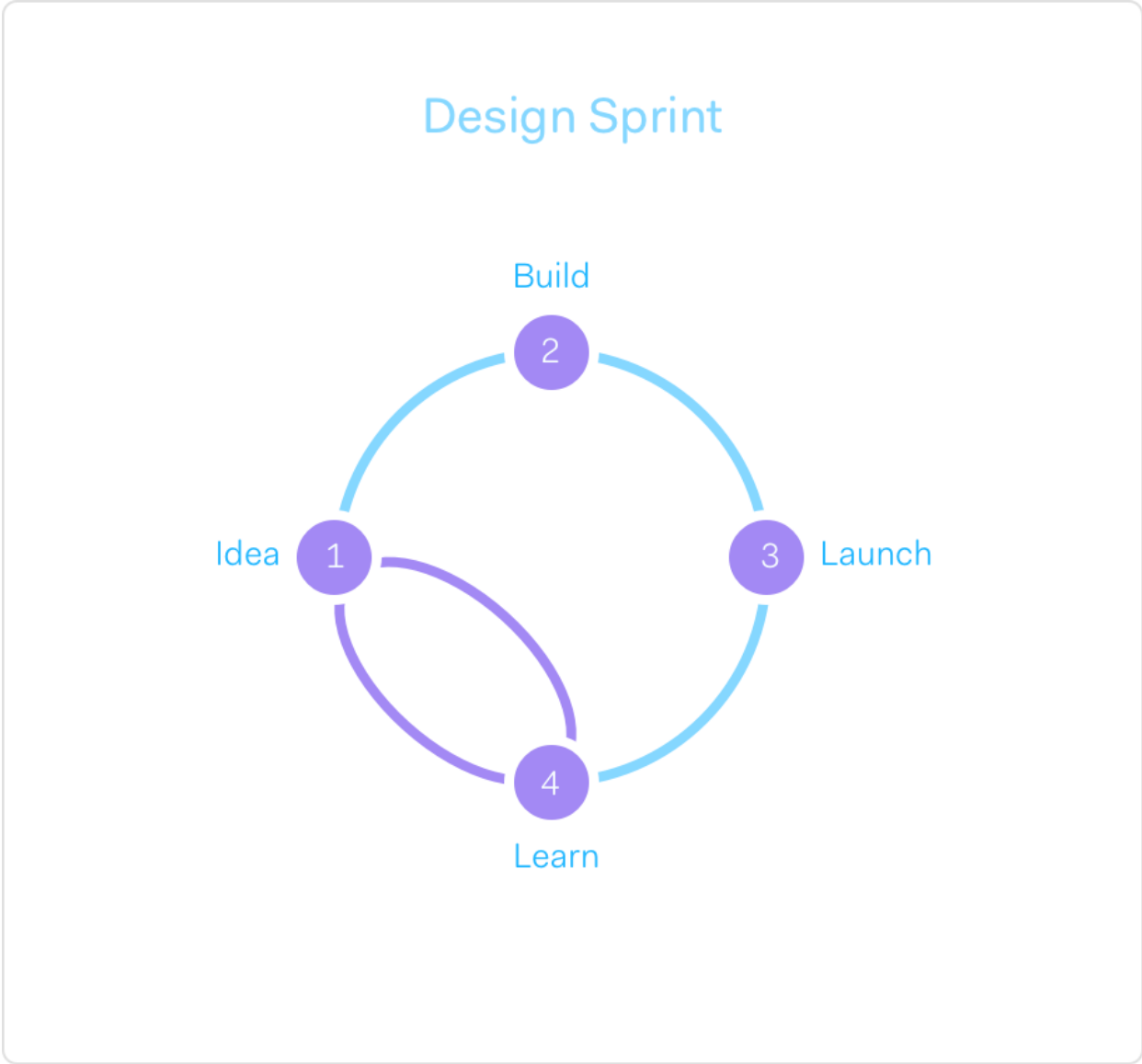
Read the article on manifesto.co.uk >

4. GV Design Sprints

What is it?

The Design Sprint by Google Ventures can be seen as a combination of the three approaches above: Design Thinking, Agile, and Lean. Let's refer to Google's own definition for clarity:

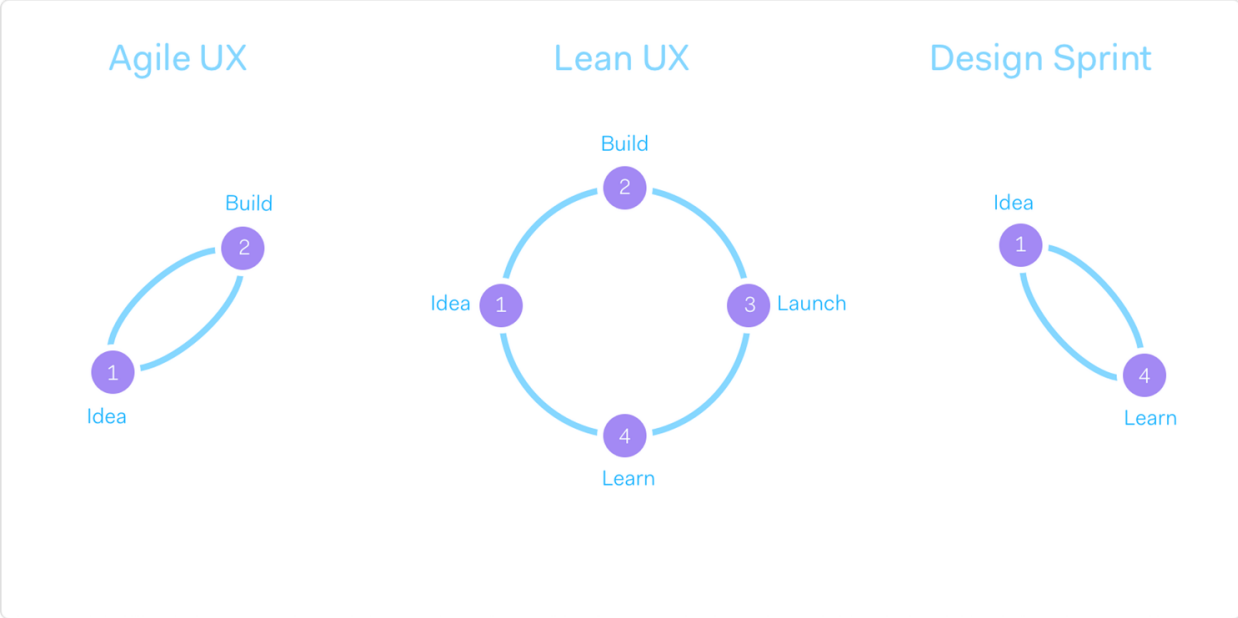
> *The sprint is a five-day process for answering critical business questions through design, prototyping, and testing ideas with customers. Developed at GV, it's a "greatest hits" of business strategy, innovation, behavior science, Design Thinking, and more—packaged into a battle-tested process that any team can use.*



GV Sprint, adapted from GV.com

Focus on Innovation

As highlighted in the definition, and shown in the diagram above, Design Sprints can be a powerful tool for innovation. Where *Lean UX* tries to address everything from idea to delivery of a feature, Design Sprints focus primarily on the idea and understanding of the problem. William Deng, author in UX Mastery, helps us visualise the differences between Design Sprints, Agile and Lean. Here's our adaptation of his diagram:



Agile UX, Lean UX and Design Sprint – adapted from UX Mastery

> "Design Sprints focus primarily on the idea and understanding of the problem."

Innovate faster

The aim is to learn *without* building and launching features, as you would with an MVP (in Lean UX). This approach therefore gives chance for entire product teams to iterate and test ideas rapidly together within just 1 working week. This is especially useful in larger organisations and corporations who can sometimes be slower to move. Design Sprints get everyone together and can be a great solution to generate momentum quickly and early on in a project.

The GV Sprint also provides activities and resources for every single day in the week, so teams know exactly what they're doing, and are able to quickly test ideas. In contrast to Design Thinking, everything in a Design Sprint is set out and structured, enabling teams to move quickly through a design process. For an overview of the differences between the two, check this article out.

Learn through prototypes

A key component of the sprint is the creation of a prototype, which is one of the best ways to gather data and test ideas out. Essentially, this helps reduce as much time as possible that may be spent by engineering working on ideas that may be thrown away. As Google put it:

> *"The sprint gives you a superpower: You can fast-forward into the future to see your finished product and customer reactions, before making any expensive commitments."*

Useful Resources:

DESIGNSPRINTKIT

Design Sprint Kit

A design sprint is a five-phase framework that helps answer critical business questions through rapid prototyping and user testing. Sprints let your team reach clearly defined goals and deliverables and gain key learnings, quickly. The process helps spark innovation, encourage user-centered thinking, align your team under a shared vision, and get you to product launch faster.

Read this on designsprintkit.withgoogle.com >

UXMASTERY WILLIAM DENG

Had enough of agile sprints? Time for design sprints – UX Mastery

By now, everyone in the technology industry has either heard of, or worked in agile sprints. But what about design sprints? Popularised by GV, Google's venture capital arm, design sprints are an effective way to rapidly test and prototype ideas in the space of a week.

Read the article on uxmastery.com >

. . .

Overall, find what works for your team

From my experience working between design and development, I've found that in the best teams, 'designer *versus* developer' doesn't exist – it's more like 'designer *and* developer', with many people even doing both. Therefore, the success at which designs are delivered as intended comes down to the people in your team and how they work. Hiring the right type of developer who has an appreciation for design, or designers who value development can be the key.

> "In the best teams, 'designer versus developer' doesn't exist – it's more like 'designer and

developer'"

Try combining methodologies

After considering all of the frameworks outlined above, it's fair to say that there's no single best approach for improving collaboration between engineering and design. Instead, sometimes a combination of a few may frameworks could for your team.

For example, you might start with a Design Sprint to get everyone aligned from the offset of a project. The resulting Design Sprint prototypes can then be carried forward to provide a clearer direction for agile sprints.

This can improve certain weaknesses in the agile process too. For instance, a major criticism of agile has been that it 'has no brain', meaning that Agile can *seem* to be performing successfully since it's delivering features. The problem is, they may not be the right ones. Design Thinking and Design Sprints can help provide that brain to agile.

The prototype is the new spec

As seen throughout, the prototype plays a key role in pretty much every approach for bringing design and development teams together, and it's therefore become an indispensable tool for collaboration.

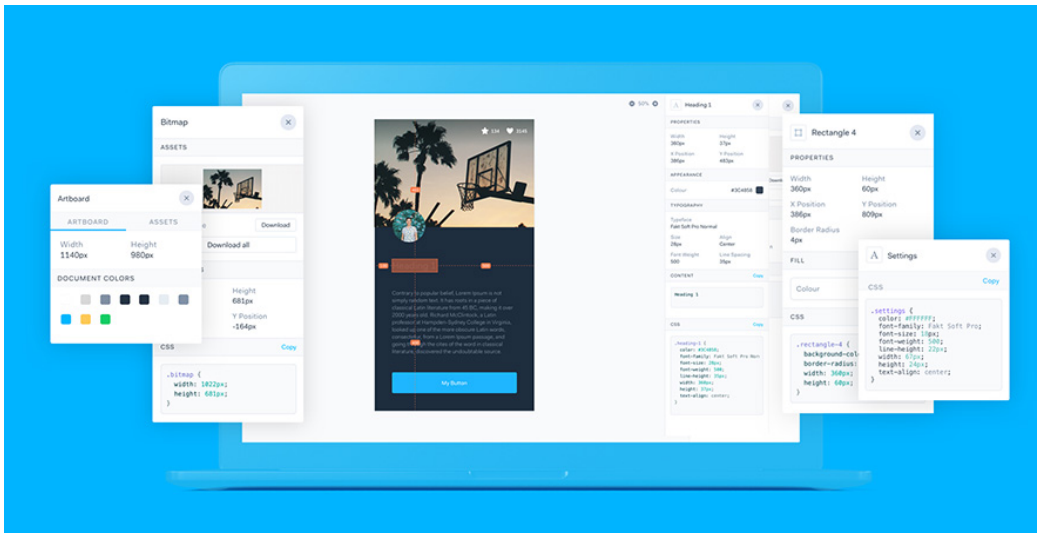
"Gone are the days of documentation – prototypes have become interactive design specs that people can experience."

In Design Sprints and Design Thinking, prototyping gives everybody the chance to influence the goal of the project and also provides a shared vision for teams to align around. With this common understanding, there becomes less of a need for lengthy documentation, as the prototype becomes the documentation itself.

It's much easier to understand a complex interaction through experiencing it via a prototype than it is from reading a long description. This is why we've made it possible to embed Marvel prototypes directly in Dropbox Paper documents, enabling teams to get more value out of the tools they're using.

Reducing the number of deliverables can also break down walls, keeping design and development working together even beyond the date a product is shipped. Marvel integrations can help you here too, as you can even embed Marvel prototypes into JIRA or Confluence for your development team to refer to.

If you're not already using Marvel, give it a go today – be sure to check out our latest feature, Handoff, and see how we can help you bridge the gap between design and development:



Turn designs and prototypes into code, specs and assets.

Learn more

Graeme Fulton

Designer, Writer and Developer at Marvel! Say hello on Twitter.

Related Posts

Blog

The Quest to Find the Happiest Notes App

René Galindo

For some reason, I've always been obsessed with note-taking apps. It took me some time to understand why, but looking back (literally at notes from the past), I've noticed that they have essentially become idea vaults. Note-taking apps is where we store our ideas, hence their...

Read more

Blog

What is Usability Testing?

 Kitty Wong

Since the early 90's, the terms 'Usability' and 'User Experience (UX)' have been talked about non-stop in the world of design, and are often referred to as 'must haves' in the design process for products and services on multiple platforms. Where this couldn't be more true, it's hard to...

Read more

Blog

A Handy User Experience Testing Checklist for Fast Paced Designers

 Naomi Francis

With usability testing software on the rise, there is no reason why a designer can not run user tests quickly and effectively when needed. With the introduction of our own ux testing software, we're on a mission to create the resources you need to run user testing sessions...

Read more