

UI Design

A How-To Guide for
Programmers

Mark Turner

Hello!



Mark Turner

Head of Research Software Engineering

Newcastle University



**Martin
LeBlanc**

CEO IconFinder

A user interface is like a
joke. If you have to explain
it, it's not very good.

Contents



First Principles

Things to know before you get started



Know Your Medium

Understanding the impact of your spacing, text, colour and context



Tools & Techniques

How to realise your design from concept to implementation



Good, Bad & Ugly

How to be inspired and learn from others

First Principles

Before even touching a mouse, keyboard, pencil or paper there are some things to keep in mind.







01

Start with user needs

Service design starts with identifying user needs. If you don't know what the user needs are, you won't build the right thing.

Do research, analyse data, talk to users. Don't make assumptions. Have empathy for users and remember that what they ask for isn't always what they need.



02

Do less

If we've found a way of doing something that works, we should make it reusable and shareable instead of reinventing the wheel every time.

This means building platforms and registers others can build upon, providing resources (like APIs) that others can use, and linking to the work of others. We should concentrate on the irreducible core.



03

Design with data

In most cases, we can learn from real world behaviour by looking at how existing services are used. Let data drive decision-making, not hunches or guesswork.

Keep doing that after taking your service live, prototyping and testing with users then iterating in response. Analytics should be built-in, always on and easy to read. They're an essential tool.



04

Do the hard work to make it simple

Making something look simple is easy. Making something simple to use is much harder (especially when the underlying systems are complex) but that's what we should be doing.

Don't take "It's always been that way" for an answer. It's usually more and harder work to make things simple, but it's the right thing to do.

Avoid quick fixes and the 'that'll do mentality'. It's a hard thing to avoid, especially if you've fallen out of love with the project but it makes life easier for both developers and users.



05

Iterate. Then iterate again

The best way to build good services is to start small and iterate wildly. Release Minimum Viable Products early, test them with actual users, move from Alpha to Beta to Live adding features, deleting things that don't work and making refinements based on feedback.

Iteration reduces risk. It makes big failures unlikely and turns small failures into lessons. If a prototype isn't working, don't be afraid to scrap it and start again.



06

This is for everyone

Accessible design is good design. Everything we build should be as inclusive, legible and readable as possible. If we have to sacrifice elegance — so be it.

We're building for needs, not audiences. We're designing for everybody, think about people from the start.



07

Understand context

We're not designing for a screen, we're designing for people. We need to think hard about the context in which they're using our services. Are they in a public place? Are they on a phone? Are they only really familiar with Facebook? Have they never used the web before?

The way people interact with the web is changing, you need to think about the situation in which a user may be interacting with your application - and what medium gives the best experience in that situation.



08

Build digital services, not websites

A service is something that helps people to do something. Our job is to uncover user needs and build the service that meets those needs. Of course, much of that will be pages on the web, but we're not here to build websites.

The digital world has to connect to the real world, so we have to think about all aspects of a service, and make sure they add up to something that meets user needs.



09

Be consistent, not uniform

We should use the same language and the same design patterns wherever possible. This helps people get familiar with our services, but when this isn't possible we should make sure our approach is consistent.

This covers a wide range of concepts from always positioning the menus in the same places on your pages to wording and naming conventions used throughout your application.



10

Make things open: it makes things better

We should share what we're doing whenever we can. With colleagues, with users, with the world. Share code, share designs, share ideas, share intentions, share failures. The more eyes there are on a service the better it gets – howlers are spotted, better alternatives are pointed out, the bar is raised.

Where possible make your code open and encourage others to contribute to common aims.



Know Your Medium

Understanding the impact of
your spacing, text, colours
and context

The Grid System

Seeing in Grids

Dynamic grids provide structure and work across devices

Articles

Compose to Vertical Rhythm

On the Web, vertical rhythm is contributed by three factors: font height and margin/padding. All of these factors must calculate with care in order that the rhythm is maintained.

04.Dec.2008

try
gn
focuses on the
tems of
ng, such as the
tion and root
as well as
uch as
the Fibonacci Series.

used separately
or in tandem.
04.Dec.2008

try
gn
focuses on the
tems of
ng, such as the
tion and root
as well as
uch as
the Fibonacci Series.

04.Dec.2008

Incremental leading

In editorial design, there is

Graph Paper by Konigi

This graph paper is made

The Typographic Grid

We consider this to be the

InDesign 11x17 Grid System (12)

An Adobe InDesign file with a

UX Magazine

A well designed collaborative site, with a very nice grid structure, that focuses on user experience.

02.Dec.2008

Doane Paper Utility Notebook

A portable notebook

Ace Jet 170

AisleOne
Athletics

BBDK
Blanka
Build
Corporate Risk Watch
David Airey

Dirty Mouse
Experimenta
Experimental Jetset
Form Fifty Five

Grafik Magazine
Grain Edit
Graphic Hug
Helvetica Film

stem is an aid, not a guarantee. a number of possible uses and each man look for a solution appropriate to his style. But one must learn how to use the art that requires practice." Peter-Brockmann

Negative Space

Sometimes it's more about the empty space than the components



You've got mail

Get a smarter all-in-one email platform to start growing your business the right way

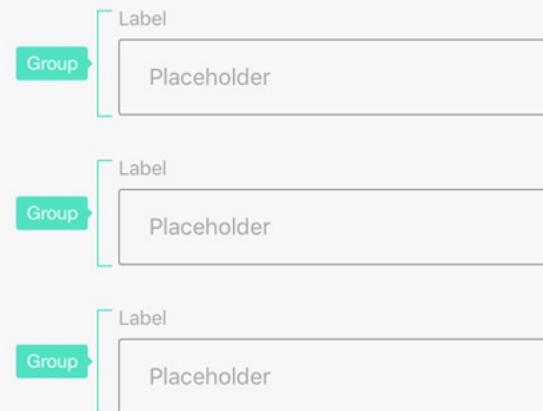
Start now



You've got mail

Get a smarter all-in-one email platform to start growing your business the right way

Start now



Headers & Copy

Use the right font for the right purpose in the right moment

BIG = Important

Medium = Standard copy

Small = Supplemental Info

Tiny = Important legal stuff

Helping to Read

Use biology to keep your users
comfortable and engaged

Black on White

Almost Black on Almost White

Colour is Powerful

It can completely change a user's expectations

The importance of

COLOUR



MOON



DEATH
STAR

stephenwildish.co.uk

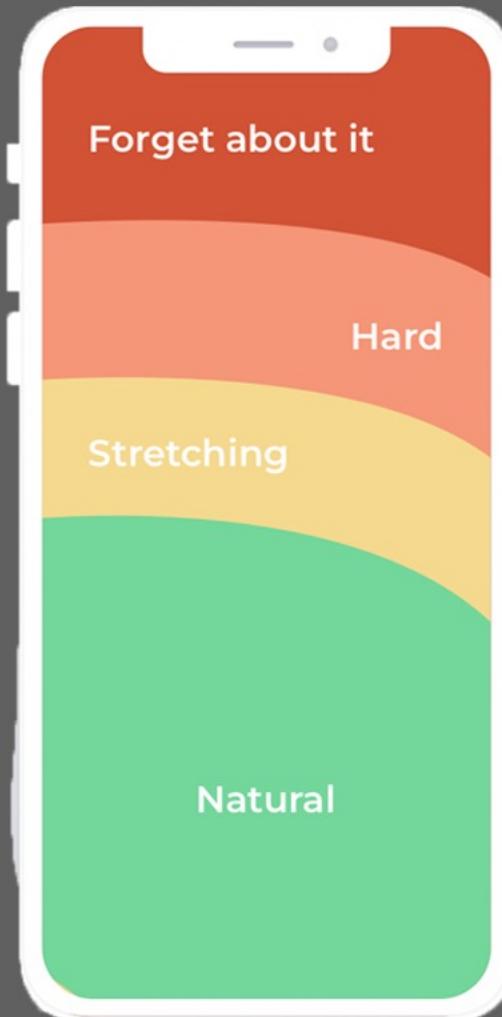
Products of Culture

Colour is culturally significant, which makes understanding your users important.

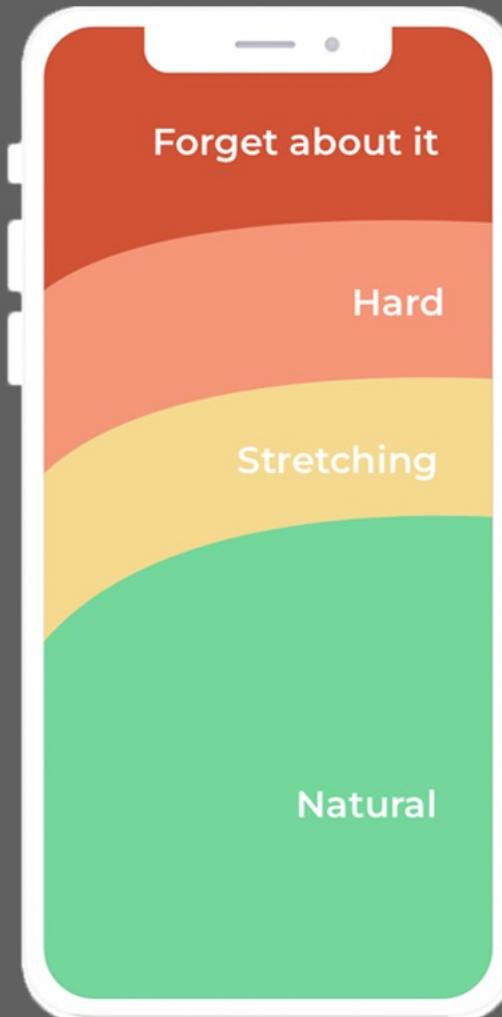


Context is Everything

What else is going on when the user is interacting with your application?



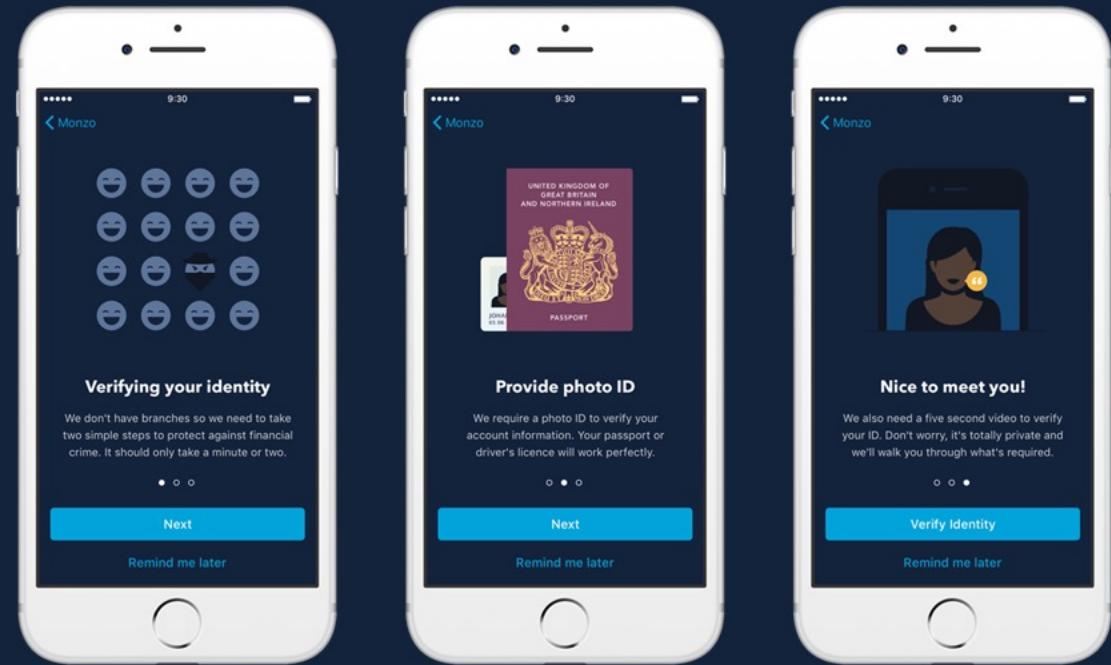
LEFT



RIGHT

Good Assumptions

Sometimes you know what a user is trying to do. Use that to your advantage.



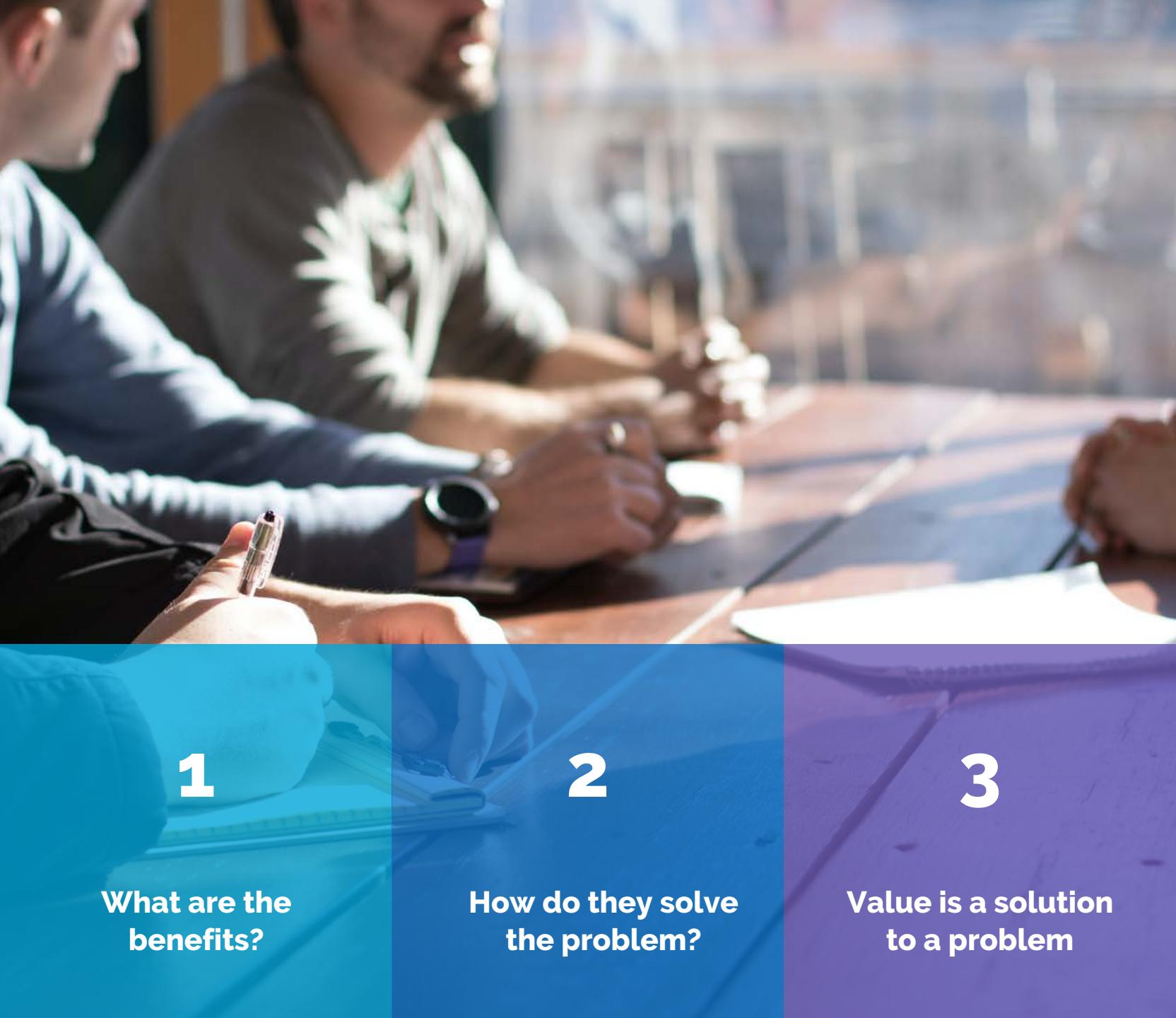
Tools & Techniques

How to realise your design from conception to implementation



Value Proposition

Try to make your UI the answer to the question “What problem does my users trying to solve”. Users don’t think in terms of clever features or novel coding, they think in terms of problems and how to solve them.



1

What are the benefits?

2

How do they solve the problem?

3

Value is a solution to a problem

User Interviews

How do you know what problems your users have if you don't ask them? Gleaning insight from users is hard but not impossible. This can be done before you start designing, during the process as you iterate, and at the end.



1

**Focus on tasks
and problems**

2

**Crowdsource
input from users**

3

**Monitor to check
user input**

Journey Maps

If your application performs a series of functions, draw out the flow a user takes to complete each task. Thinking in this way will highlight bottlenecks, inefficiencies and discrete concepts within the app.



1

**Focus on
individual tasks**

2

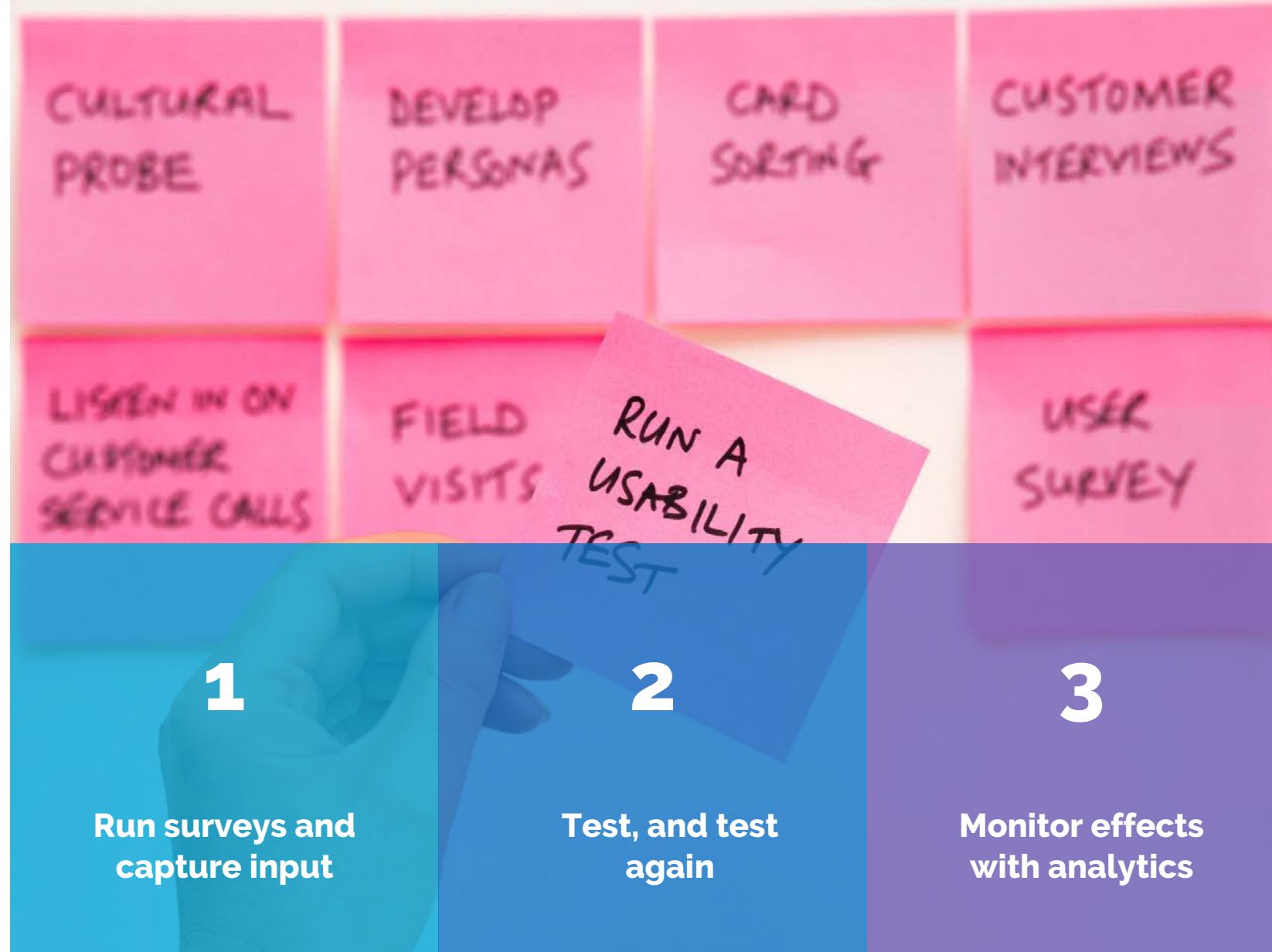
**How do they fit
together?**

3

**Communicate
throughout**

User Testing

With larger user bases companies will engage with different types of testing. Testing replaces interviews to scale getting insight into your users. Testing can take the form of surveys, A/B testing or application analytics and much more.



Wire Frames

Wireframes are the single most time saving investment you can make. They force you into understanding your whole application before you start coding. Fixing mistakes on paper is much faster than fixing mistakes in code.



Balsamiq



Moqups

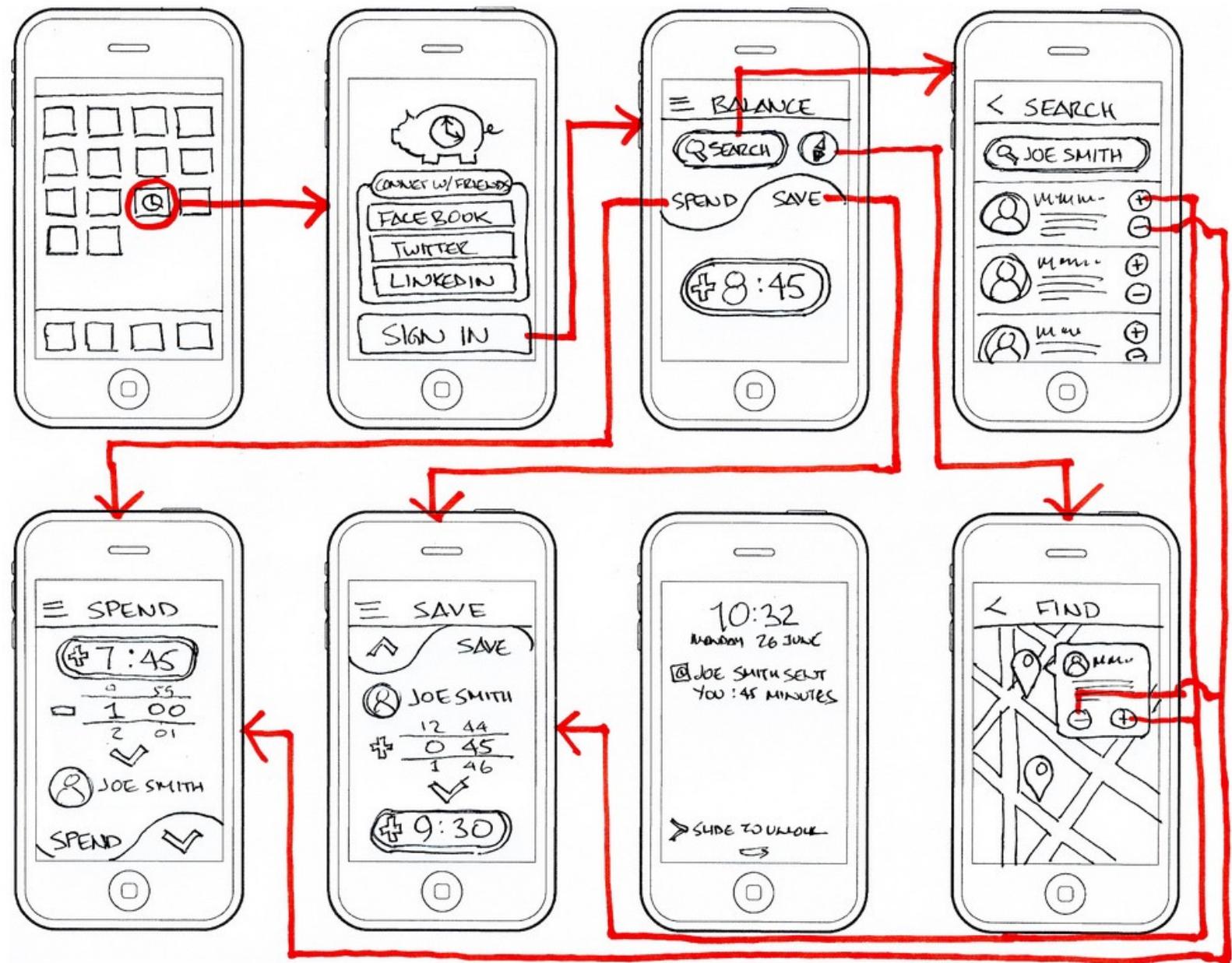


Wireframe CC



Why Wireframes?

Wireframes help you make sense of the list of requirements by conceptualising each part of the interface. It helps spot bottlenecks, disconnected features and poor UX before sinking too much time into the project. Reconnect with pen and paper, the first wireframes are done on whiteboards and notepads.



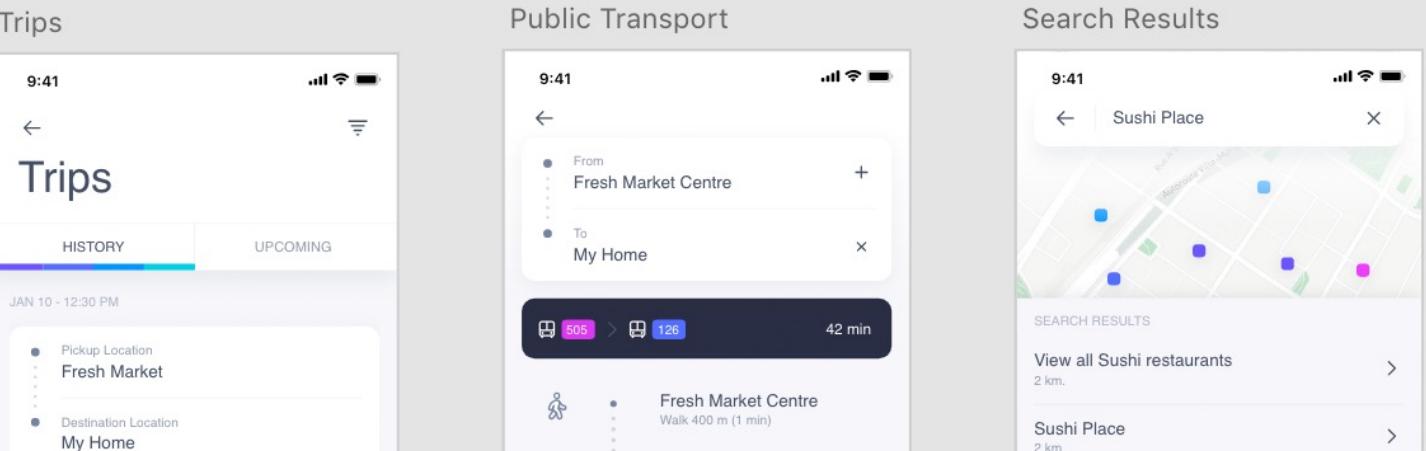
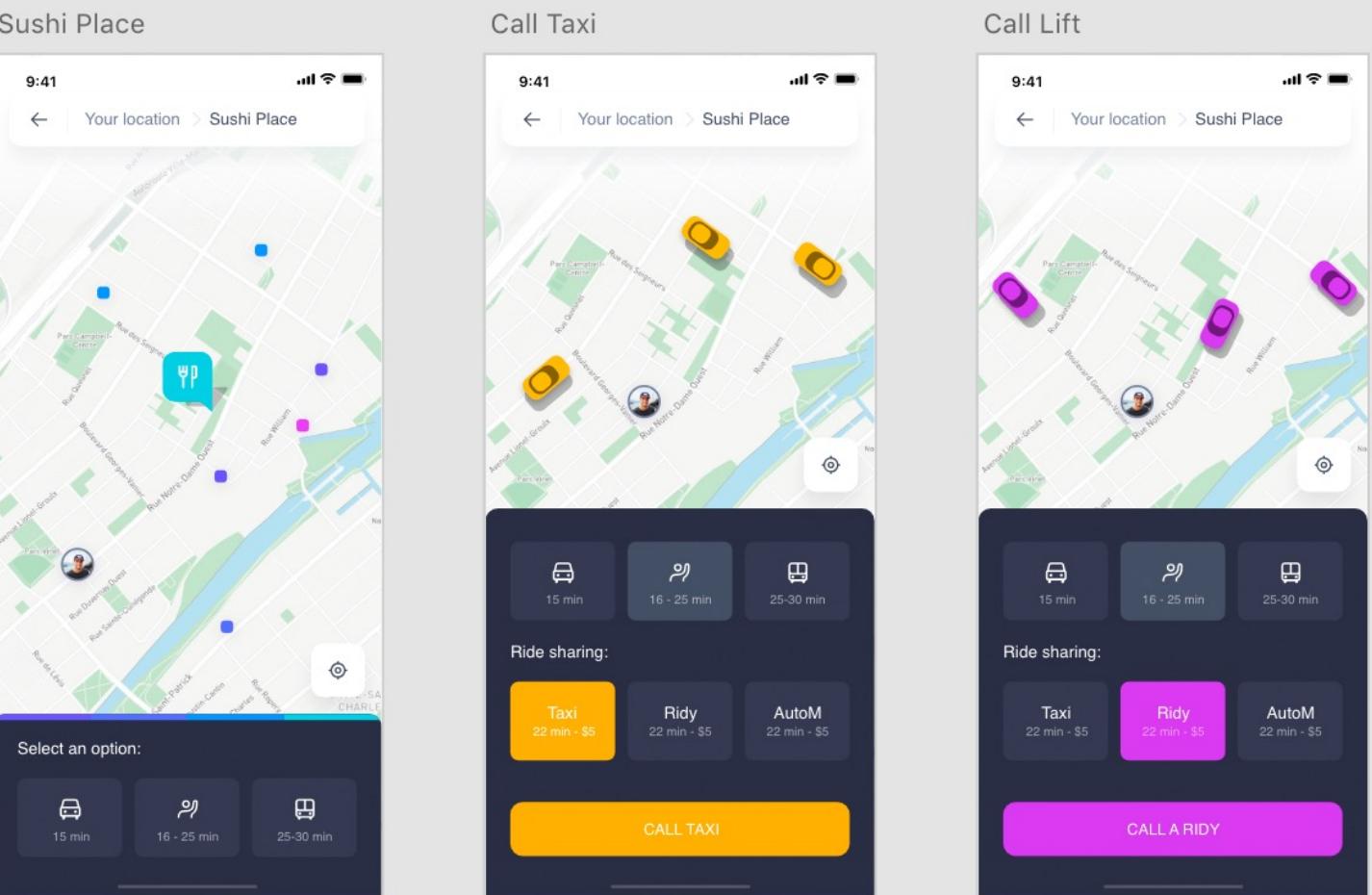
Rapid Prototyping

The next level of prototype is essentially wireframes + interaction. Prototypes are digital versions of your screens, often with early versions of assets such as fonts and colours, that are clickable and navigable. This is a stage where you start to involve users in the design process.



Wireframe to Prototype

Prototypes are a more immersive and interactive version of your wireframes. They make it easier for users to experience what the app will be like and start to introduce aspects such as colour and multimedia.



Colour Palettes

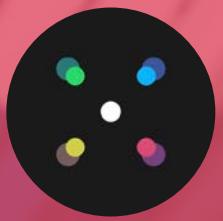
What colours go together? How do people put colour palettes together? Luckily, colour transcends UI design and is an aspect of all design – steal from fashion, interior design, cinematography and nature.



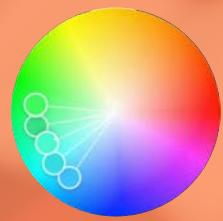
Coolors



ColorMind



Adobe Colors



Font Libraries

Font libraries help you use fonts that are not part of the standard web-safe offerings that come pre-installed on your OS. By bundling the font files with the code, you can give users the same experience across devices and vendors.



Google Fonts



Font Squirrel



Icon Finder





Good, Bad & Ugly

How to be inspired and
learn from others

The Takeaway

1. Use principles to frame your work
2. Understand the power of medium
3. Use tools to help plan your design
4. Don't be afraid to copy from others
5. All design is an exercise in restraint