



CODECLAN

CODECLAN'S CRASH COURSE

USER EXPERIENCE DESIGN

Hello and welcome.

Introduction — who am I?

WHAT ARE WE GOING TO LEARN TODAY?

ROUGH ORDER OF EVENTS

- ▶ What is UX? How does it work?
- ▶ UX and web developers — why?
- ▶ Design principles
- ▶ Proto-personas
- ▶ User needs
- ▶ User journeys
- ▶ Interface design principles
- ▶ Interface sketching with prioritisation
- ▶ Group work
- ▶ Summary

These are all the things we will be talking about today. We will do some individual work and also group work.



UX CRASH COURSE FOR DEVELOPERS

PART 1/2

OBJECTIVES

BY THE END OF THE FIRST PART YOU SHOULD...

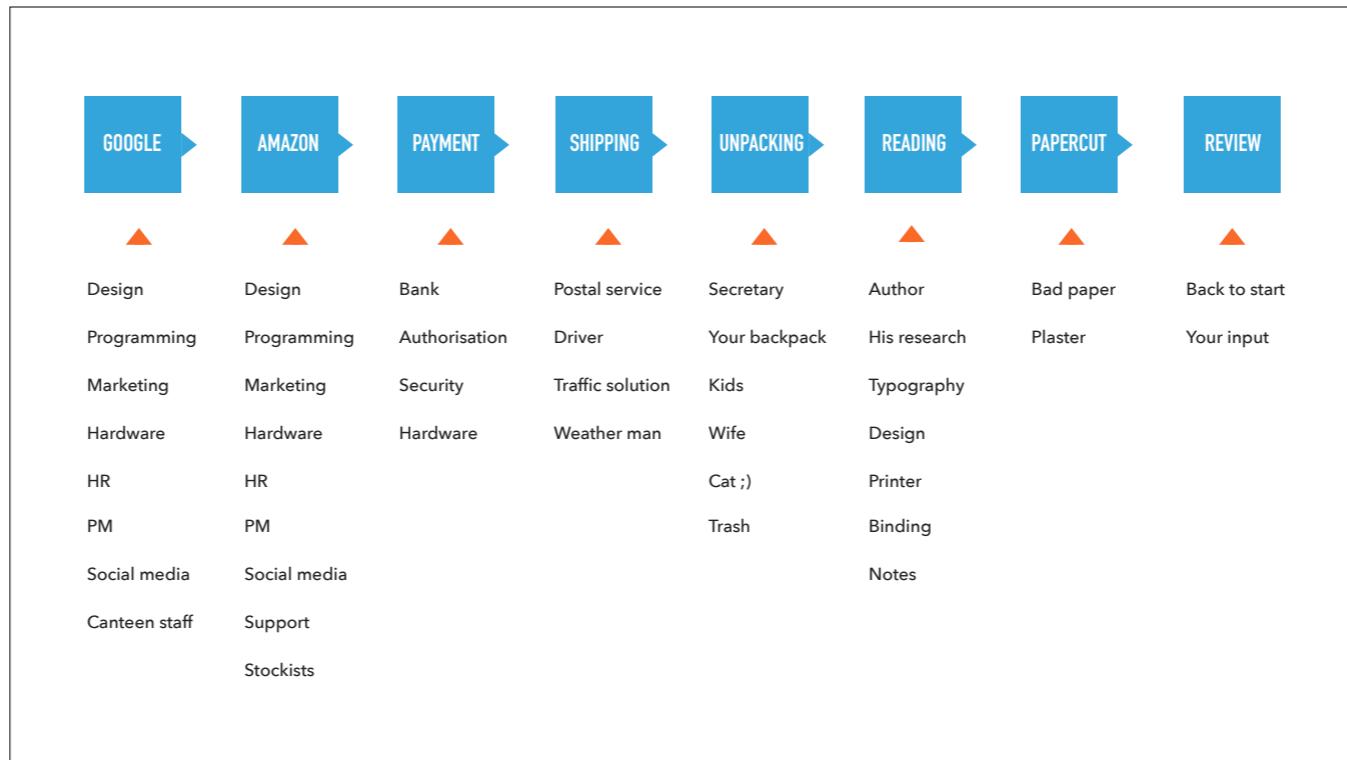
- ▶ Get a good overview of UX, its history, purpose and methods
- ▶ Learn about the importance of UX and its connection with typical WebDev work
- ▶ Be able to position yourself within the software development lifecycle



USER EXPERIENCE DESIGN

WHAT IS IT?

Let's talk about what UX is first. To understand what is it, we need to get our heads around an experience. What does it constitute of?



Design is not just what we see. Consider this experience of buying a book on Amazon — how many professions and stages are involved?

SOFTWARE EVERYWHERE, WE'RE ALL DESIGNERS

It's not only what happens in the browser.

The process that we're going through as developers (and other designers) it's a product in itself.

What is a product? It's an effect of the work of all these parts. It can be a website, or an app, or even a physical product that has a digital component to it (like an appliance or a gadget).

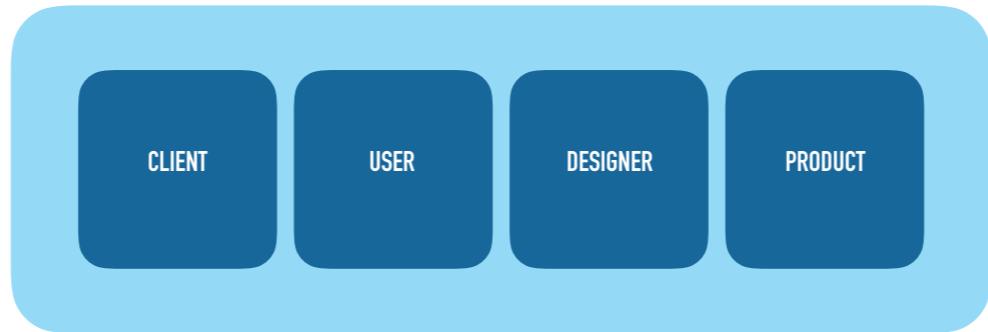
Think of it this way: you're using your skills to work with others. To develop, to research, to document.
Would you rather see them work together with you and have a good experience or struggle?

THE DEFINITION

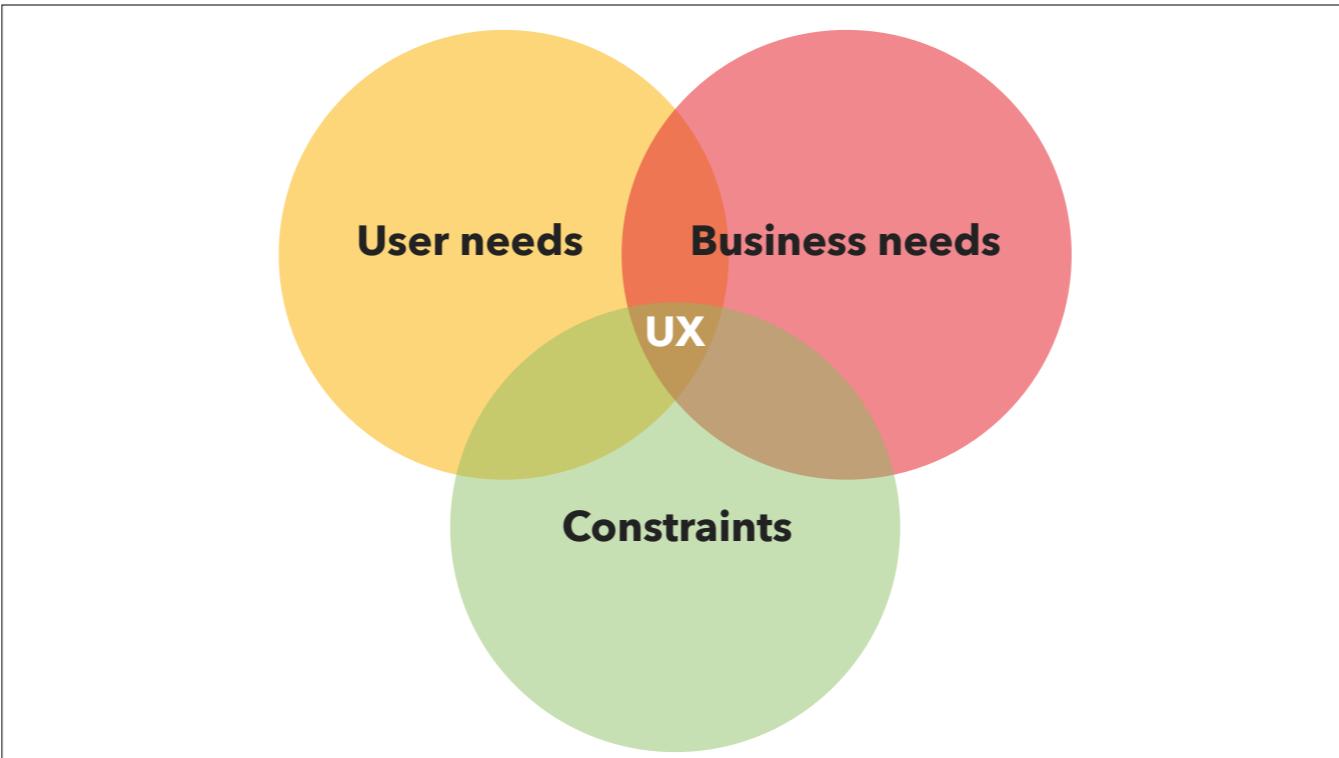
- ▶ User experience design is a design discipline that focuses on delivering products that meet organisational and user needs. It derives from psychology, HCI, informatics and information design.
- ▶ It is a set of methods that designers employ in order to understand requirements and build products that work.
- ▶ It is by its definition a collaborative, engaging and efficient way of attacking design problems. Plus, it's fun. Seriously.

DIFFERENT FOCUS

Four components of the product development process



These are four components of the development process.



UX processes only work well if they take into considerations business (organisational needs) and technology constraints.

WHAT IS A CONSTRAINT?

- ▶ It's a restriction put on the system (or any of its components) that makes delivery of a feature set difficult or impossible. It then requires designers to rethink their strategy. It can be caused by technology, policy or limitations that users face.
- ▶ Examples:
 - ▶ Mobile connectivity unavailable for Google Maps in far Highlands
 - ▶ Insecurity of sending sensitive data via email
 - ▶ Inability to ask for passwords over the phone
 - ▶ Inability of those, who can only use voice for navigating physical controls of the keyboard

CONSTRAINT IMPLEMENTATION PLAN

Topic	Possible effect of constraint on product	Solution
Hardware and software platforms	System crashing if not enough RAM memory.	Optimise code, increase system memory
Performance requirements	Data unavailable when no wifi signal present	Cache data offline
Persistent storage and transactions	Payment failure if card declined from the authorisation centre	Provide other methods of payment
Usability	If passwords too complex – logging in would be a problem	Allow for natural, sentence-long password
Budgets	We'd have to use HTML framework (dev unavailable). Low accessibility.	Choose accessible framework or increase budget
Time limitations	Work has to be done in three weeks, quality might suffer. Not enough QA.	Increase the project time by a week.



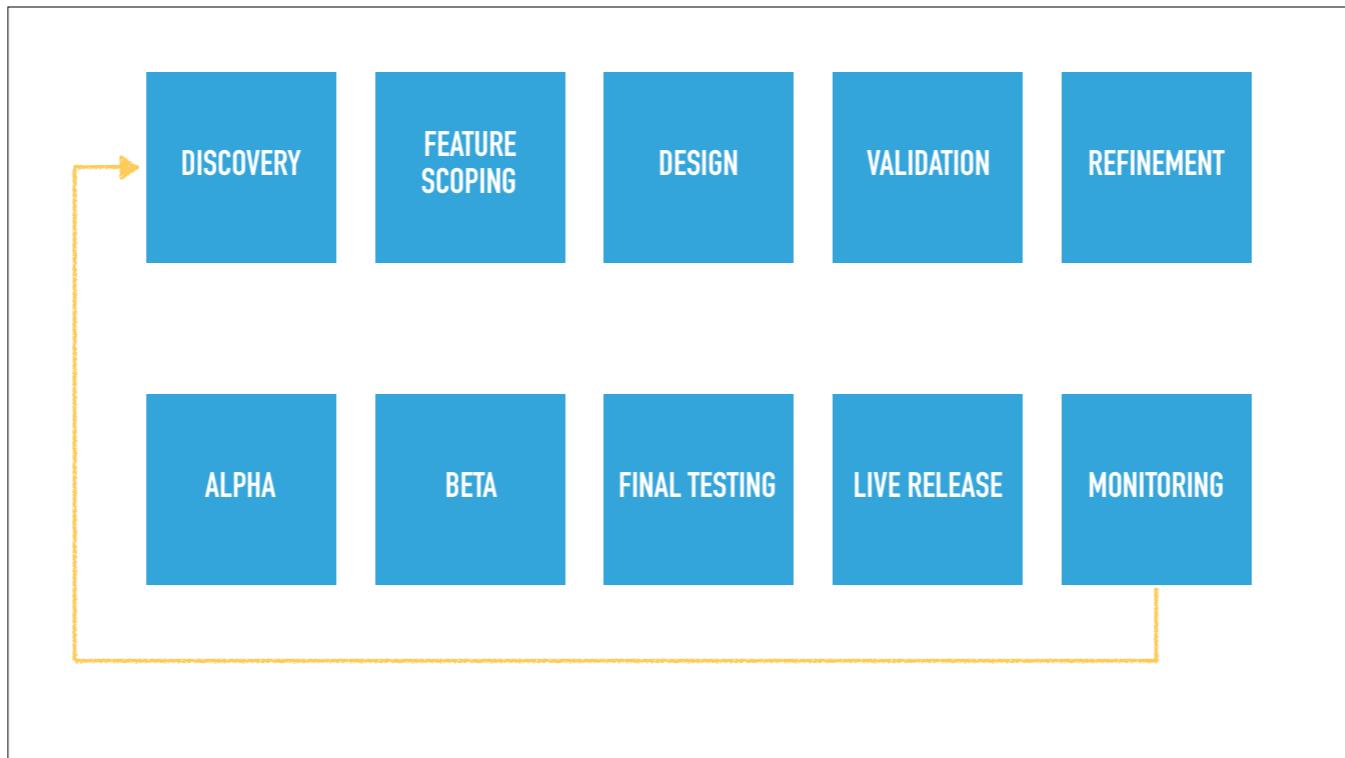
USER EXPERIENCE DESIGN

METHODS

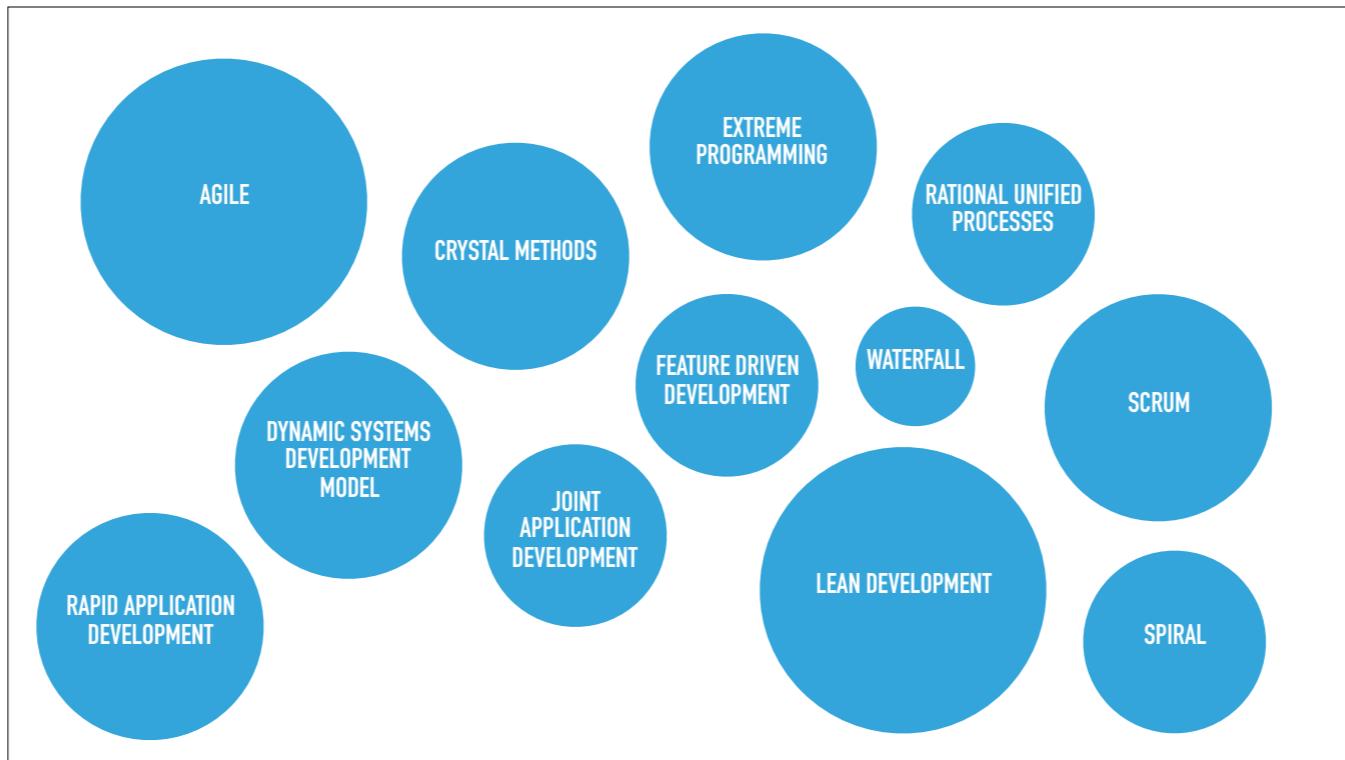
We know what UX is. But how does it work?

UX design applies a number of practical methods into the process that surfaces good products. Let's talk about these.

In order to do that, we have to look at the process itself. How is software made?



This is the typical user-centred design process. Now, you might have heard of different software development methodologies.



The range of development methodologies is vast.

UX works independently from any software development method. It fits them all.

These are some of the software development methodologies, with Agile, Lean, Scrum and Waterfall being probably the most widely adapted.

Some of them are based on actions that are well planned and documented, and in some others there's a more hands-on approach to doing things.

Some are heavy on testing small parts of functionality – like Lean – and some are more focused on rapid development in shorter sprints, which are parts of a project.

You don't have to remember this all – not at all. You will be working with some of these, but so far what's important to understand is that there's a space for UX in all of them.

You can read the descriptions here:

<http://www.itinfo.am/eng/software-development-methodologies/>

DISCOVERY METHODS

- ▶ Target group definition: proto-personas
- ▶ User needs discovery
- ▶ User journey mapping

DESIGN METHODS

- ▶ Interface sketching
- ▶ Interface wireframing and prototyping

Here at CodeClan we're trying to make you into web developers. Perhaps some of you will become user experience designers, but that's not known at this stage.

That's why we want to show you a number of methods that will help you pushing your work forward.

Let's look at these methods in bit more detail.



UX FOR DEVELOPERS

WHY IS IT IMPORTANT?

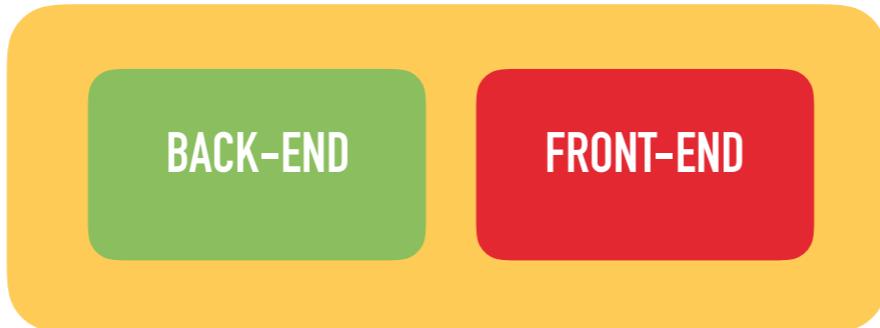
This is all good — but how does UX work for developers? Why is it important to them? To you?

IMPORTANCE OF USER EXPERIENCE DESIGN

- ▶ The question of 'how' vs. 'why'
- ▶ Products that answer real needs — no need to spend time developing useless features
- ▶ More robustness and future-proof code
- ▶ Greater customer satisfaction directly translated to profit
- ▶ Fantastic collaborative experience
- ▶ Iterative process that makes us work for better result all the time
- ▶ Less bad products on the market, less information overload and environmental impact

TRADITIONAL WAY OF APPROACHING DEVELOPMENT

Product



Back-end is everything that you do. But nowadays you will be often crossing into the front-end territory, traditionally occupied by visual designers and interface designers, with content writers, marketing people and so on.



Nowadays it looks different. Developers are often required to work both in front-end and back-end roles. With JavaScript we can do full stack development now. Roles are mixing. You'll even find UX designers doing parts of coding work when wireframing, or marketing professionals coding JavaScript for Google Analytics.

Illustration: Codecondo

THE GIST IS:
IN ORDER TO DEVELOP GOOD
SOFTWARE YOU NEED TO CARE
ABOUT UX. AT LEAST A BIT.

That's what it's down to.



UX CRASH COURSE FOR DEVELOPERS

PART 2/2

OBJECTIVES

BY THE END OF THE SECOND PART YOU SHOULD...

- ▶ Be able to follow basic design guidelines
- ▶ Apply the following methods into your practice:
 - ▶ Proto-persona profiling
 - ▶ User needs and journey analysis
 - ▶ Sketching
- ▶ Experience discussed UX methodologies in a group activity setting.



GOOD FRAMEWORK

WHAT MAKES PRODUCTS WORK?

There is a set of standards that all developers and other designers should think about when developing products. After all, we need to produce the best stuff. How can we ensure that this is the case?

Well, there are some design principles to follow.

This list is based on industry standards, but you can look into things like [GDS Digital Principles](#) or [Inclusive Design Principles](#) to learn more.

Keep it simple (KISS)

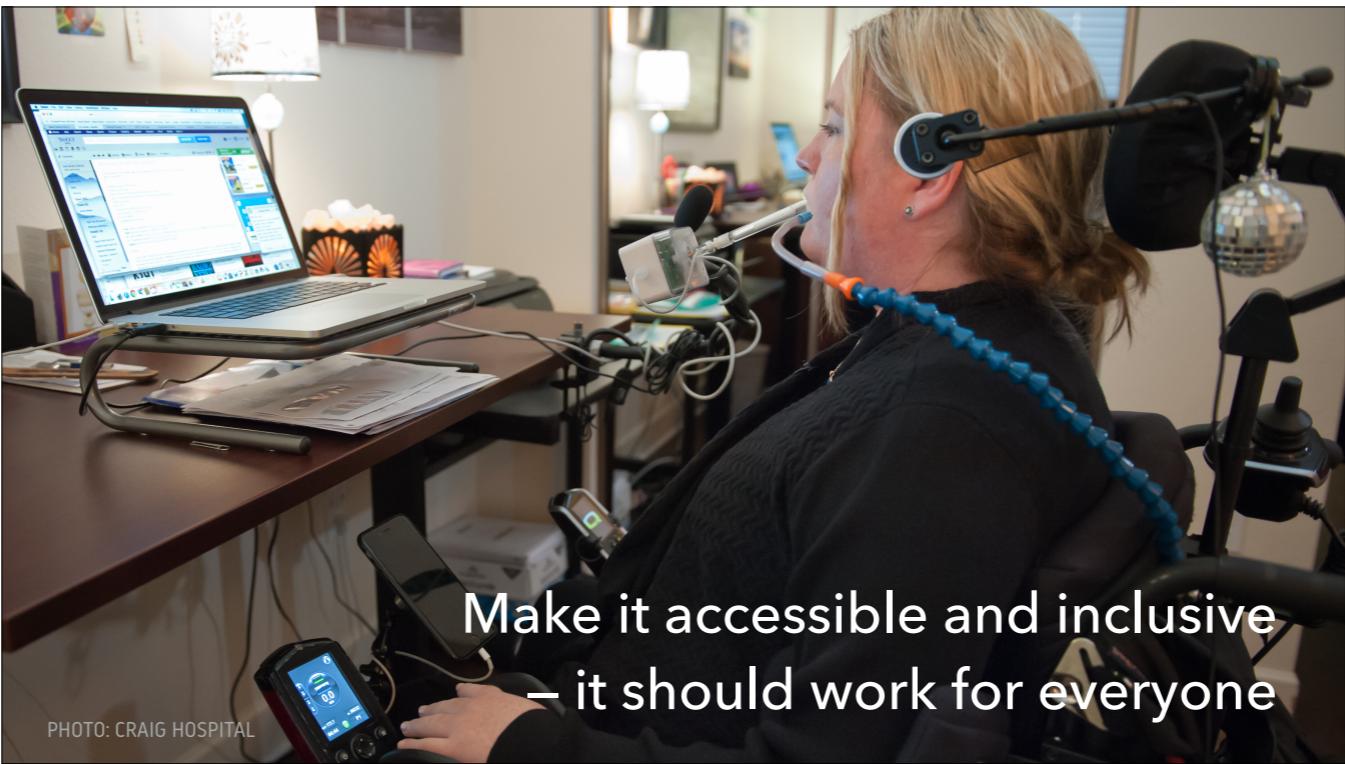


PHOTO: WIKIPEDIA

Rule 1: Simplicity.

Our work has to be simple and work for the end user. They should not struggle.

This means that we **have to do the hard work to make it simple**. It's not users who need to struggle — if we have to on the way to the product, so be it.



Make it accessible and inclusive
– it should work for everyone

PHOTO: CRAIG HOSPITAL

Rule 2: accessibility and inclusivity

Our products have to be built in a way that accommodates for different needs.



Rule 3: Design with data. Don't assume things! Talk to your users. Find rationale for your design decisions.

Understand context

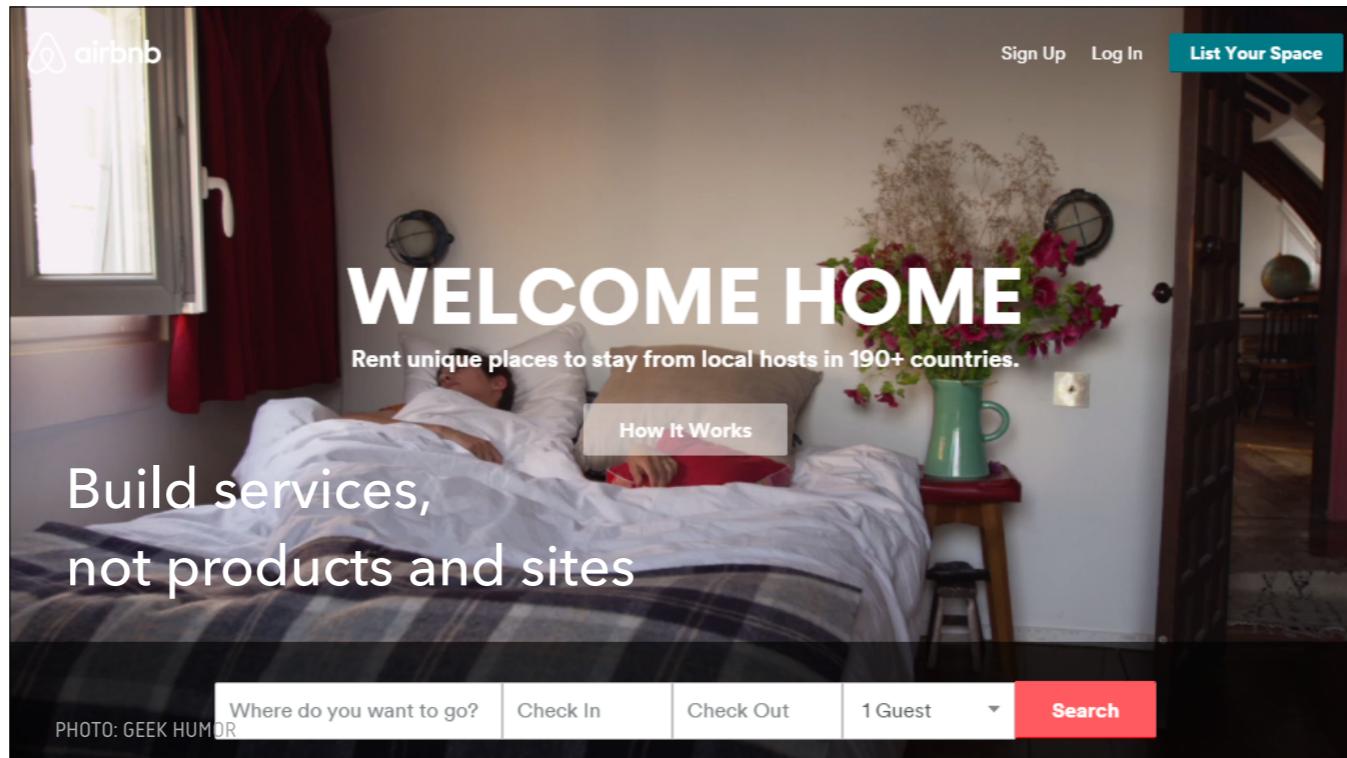


PHOTO: GEEK HUMOR

Rule 4: Understand context

Where are your users? What do they do? What could make their life and work difficult? How are they keen to respond to your program?

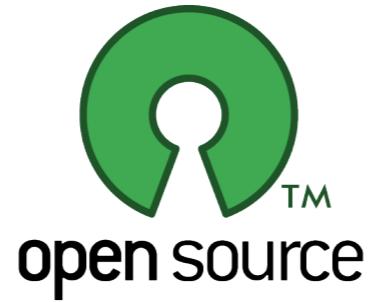
This is the real life incognito mode ;)



Rule 5: Build digital services, not websites

The website is just the start. We need to focus on the entirety of delivery to make sure our products are truly exemplary. And your work can help this happen.

Keep things open, share, contribute.



Rule 6: contribute with your work

Allow others to use it and use work of others. Contribute, develop together, keep it open, evolve.

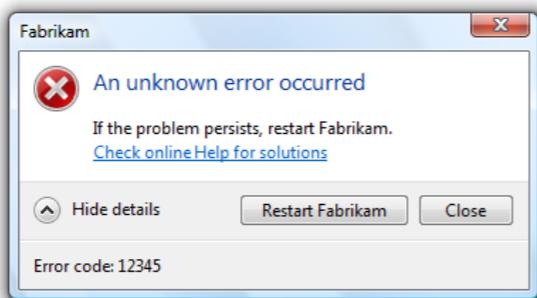
Keep your users happy and informed.



Rule 7: Keep your users happy and informed.

Make sure that your product communicates with the user, guides them and allows them to work efficiently and with no stress.

Allow users to recover from errors.



Rule 7: Keep your users happy and informed.

Make sure that your product communicates with the user, guides them and allows them to work efficiently and with no stress.



Rule 8: Test, iterate, then test, then iterate...

Always test your products and refine them after understanding the results. The only real validation to a product's success is user satisfaction. Let me repeat that

IT'S ONLY GOOD IF IT'S
USABLE, FRIENDLY AND
PRACTICAL.

Got it? :)



LET'S GET OUR HANDS DIRTY

PRACTICAL UX

It's time for us to do some work. The rest of the day we'll spend on finding our way to a digital product.

WHAT ARE WE GOING TO DO?

- ▶ Understand our target group and produce proto-personas
- ▶ Identify their user needs
- ▶ Draw key user journey(s)
- ▶ Sketch interfaces
- ▶ Discuss and talk about other considerations

These are all the things we will be talking about today.



PRACTICAL UX

USERS, THEIR PROFILES AND NEEDS

Let's look at what happens when we approach a design problem.

USERS, THEIR PROFILES AND NEEDS

IMAGINE THE FOLLOWING

Your friendly sandwich shop wants to open deliveries online. As a developer, you have offered to help with the website (they will pay with sandwiches).

What do we know?

What would you immediately think of when you see this? How would you approach the problem?

```
<div class="page" id="page">
  <!-- Begin .header -->
  <header class="header cf" role="banner">
    <a href="#"></a>      <a href="#search-form">
      <a href="#nav" class="nav-toggle nav-toggle-menu icon-menu"><span class="is-vishidden">Menu</span></a>
    <nav id="nav" class="nav">
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav><!--end .nav-->
    <form action="#" method="post" class="inline-form search-form">
      <fieldset>
        <legend class="is-vishidden">Search</legend>
        <label for="search-field" class="is-vishidden">Search</label>
        <input type="search" placeholder="Search" id="search-field" class="search-field" />
        <button class="search-submit">
          <span class="icon-search" aria-hidden="true"></span>
          <span class="is-vishidden">Search</span>
        </button>
      </fieldset>
    </form> </header>
  <!-- End .header -->    <div role="main">
    <div class="block block-hero">
      <a href="http://www.fillerati.com" class="inner">
        <div class="b-thumb">
```

You might be thinking any of these things. Sandwiches. Websites. Shopping cart, money, browsers.

Most likely you're thinking of code, though. You're thinking of databases. Interfaces. Ruby, HTML, JavaScript, you name it.



But... How about the users? What do they want?

Each of them might want a different sandwich. Some may want to pay by card, some — by cash. Some users might be using the website from their mobile on the way home from work (or to work), and some from their computer in the office. Perhaps some will be allergic to certain foods. Their requirements will be very different.

In order to build a new product — and to make sure that it meets needs of users and the business — we need to understand the users first.

But how can we do it if we don't really know who they are?



Colin
Aspiring guitarist

Behaviours

- Lives with two flatmates, so can't be loud
- Plays with his band every week
- Likes learning from youtube videos
- Always on a lookout for new techniques
- Goes to concerts as often as he can, to look at other players
- Records his rehearsals so can listen back and improve

Demographic information

- 28 years old, employed in a guitar shop
- Lives in Glasgow
- Studied at music school a long time ago
- Earns 16k/year

Needs and goals

- Wants to get better at music theory
- Needs to learn new styles as has many ideas for new music
- Would like to pay to a tutor but doesn't have much money
- Would like to share his successes in learning with similar people
- Wants to meet new people through music

Let's look at how we can describe these people. Meet Colin, a proto-persona of an aspiring guitarist.

Let's talk about proto-personas in detail now.

- * Based on assumptions - it's ok as will be validated
- * Drawings or photos with faces
- * Demographics, behaviours, needs and goals

Your turn. Design a proto-persona — client of a sandwich shop.

Let's discuss. What was difficult about this?

After we have proto-persona profiles, we can think of their user needs.

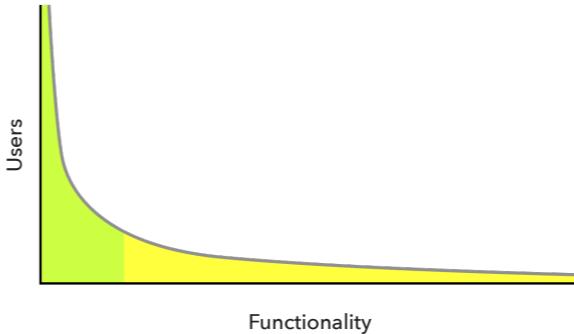
AS A... person who added the sandwich to the shopping cart
I WANT TO... pay with my card
SO THAT... I don't need to worry about paying the delivery person.

User needs — they are arising from the goals section of proto-personas. That's good for the starters. This method of gathering information about them is based on **Agile User Stories**.

In the real world, you would want to enrich them with data coming from sources such as Google Analytics and ground them in behavioural research.

LONG TAIL THEORY

- ▶ Don't build things that will only be used by small number of users
- ▶ Concentrate on functionality that delivers for most needs first
- ▶ Only then proceed with the extras.



The long tail theory was formulated in regards to content, but it does work well in connection with software, too.



PRACTICAL UX

USER JOURNEYS

We've established earlier on that everything we go through in the internet is a journey of sorts. In order to learn how to plan for the best, we shall learn how to establish an user journey.

WHY MAP USER JOURNEYS?

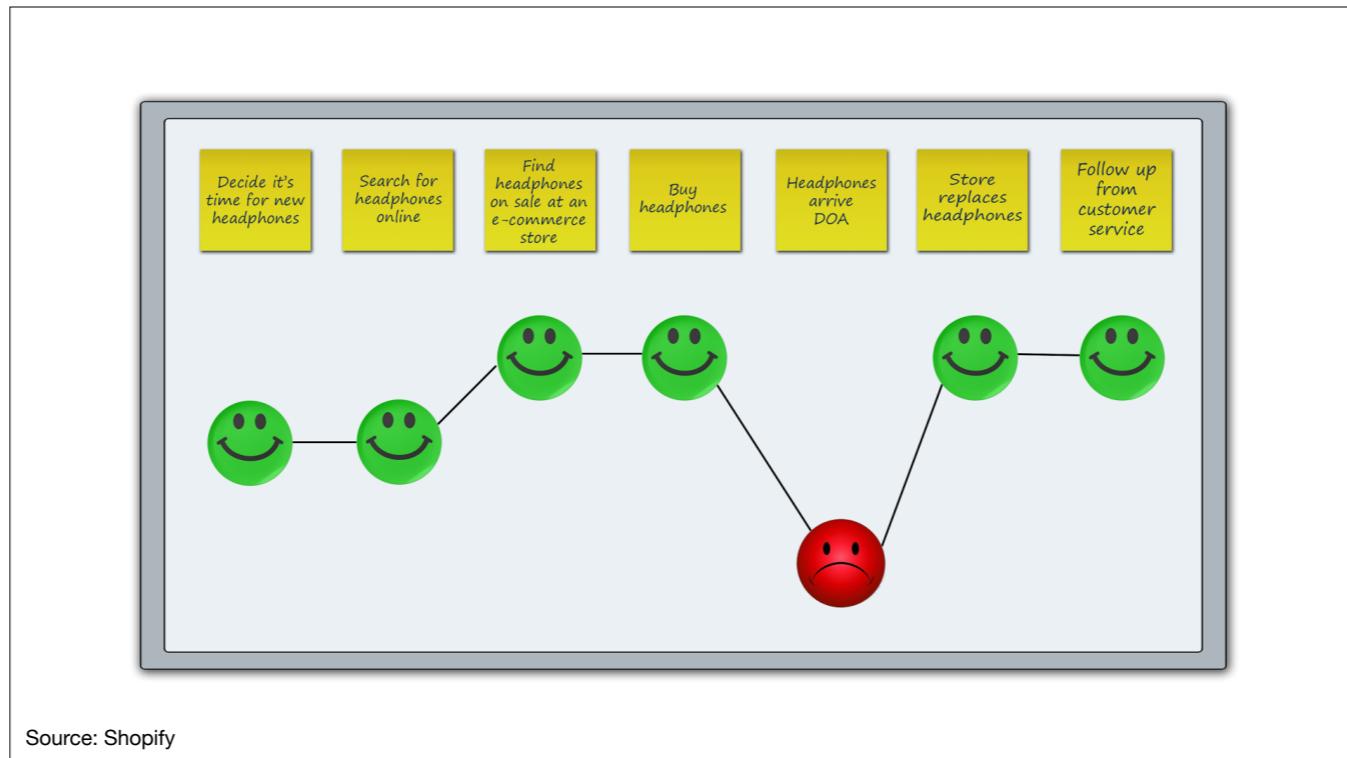
- ▶ To get better understanding of all steps involved in reaching a goal
- ▶ To narrow down requirements for the back-end and the front-end
- ▶ To identify potential pain points
- ▶ To empathise with the user a bit more
- ▶ To validate our initial persona work
- ▶ To identify new user needs.

There are some particular benefits to user journeys.

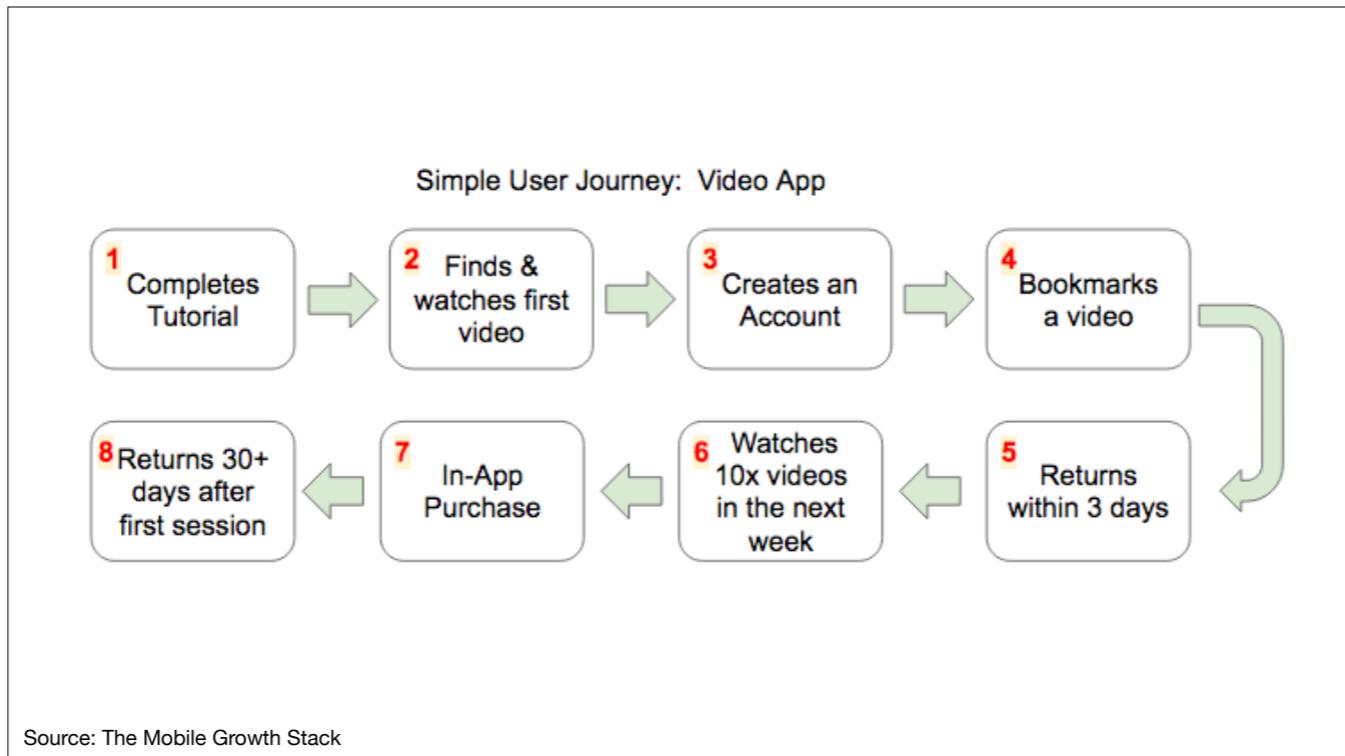
We know that real, big user journey maps can be incredibly detailed, outlining not only steps and actions, but also emotions and many other aspects of what happens.

You will be most likely building journeys like that with your new colleagues, when you get to new jobs.

Here at CodeClan we're going to do something less involved, but really helpful.



A very simple user journey, together with emotional state.



User journey mapping allows us to find out about the way to completing goals. Whilst normal user journey maps are quite sophisticated and can take a long time to produce, here at CodeClan we will be looking at a simplified version that's more applicable to what developers do.

	Record	Download to computer	Import to editing software	Edit the video	Export	Upload	Share
ACTIONS	Charge battery Buy blank tapes	Find cable Transfer to PC	Choose to install app Import footage	Add crossfades, music Tinker for hours	Choose file format, codec Wait for export - zzzz	Select hosting service Upload file	Send out chain mail Link to myspace, bebo, etc.
QUESTIONS	Do I have blank tapes? Batteries charged? Spare battery packed?	Why is it taking so long?	Which app is best? Is my PC compatible?		What's a CODEC? What settings are best?	Where can I put this file?	
HAPPY MOMENTS	Video quality is amazing! Love the long zoom				Labour of love - looks great		Hooray, friends like it!
PAIN POINTS	So much preparation!	Can't find goddam cable! out of disk space. Huh?	Hard to pick the right app	Takes way too long	It came out all blocky!	Too many decisions!	Youtube is 10 mins max Doesn't play on mum's PC Too much work overall
OPPORTUNITIES	Use flash storage not tape Allow AA batteries	Build an integrated USB cable	Preload editing app free on camera	Streamlined editor - less control but quicker!	Video recorded as web ready	Jargon free export options Share as one click feature of editor	Jargon free export options Ensure a joined up 'end to end' experience

Source: Harry Brignull @ Medium

This is a more developed journey, outlining actions, steps, questions, happy moments and pain points with opportunities.

It is providing quite a lot of information and allows us to understand what's really going on.

And this is what we will try to do. :)

User journey			
Your name _____		Date _____	
User action		User action	
1	Start	2	
System response		System response	
<hr/>			
User action		User action	
5		6	
System response		System response	
<hr/>			
User action		User action	
7		8	End
System response		System response	

Our simplified template consists of user action and user response.

How to work with this?

1. Pick a journey that you want to illustrate. For instance, we might want to purchase a sandwich for delivery.
2. Fill in all steps.
3. Think of potential problems that might arise.

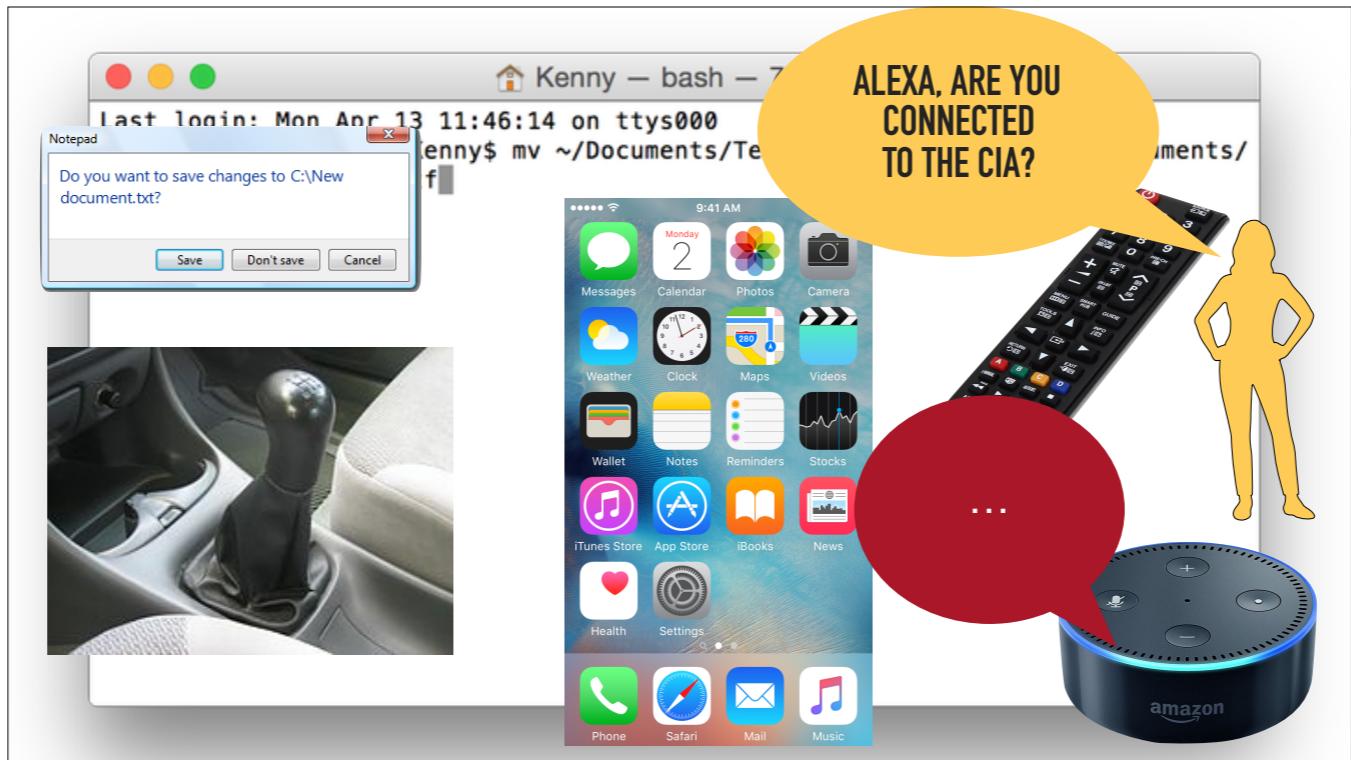


PRACTICAL UX

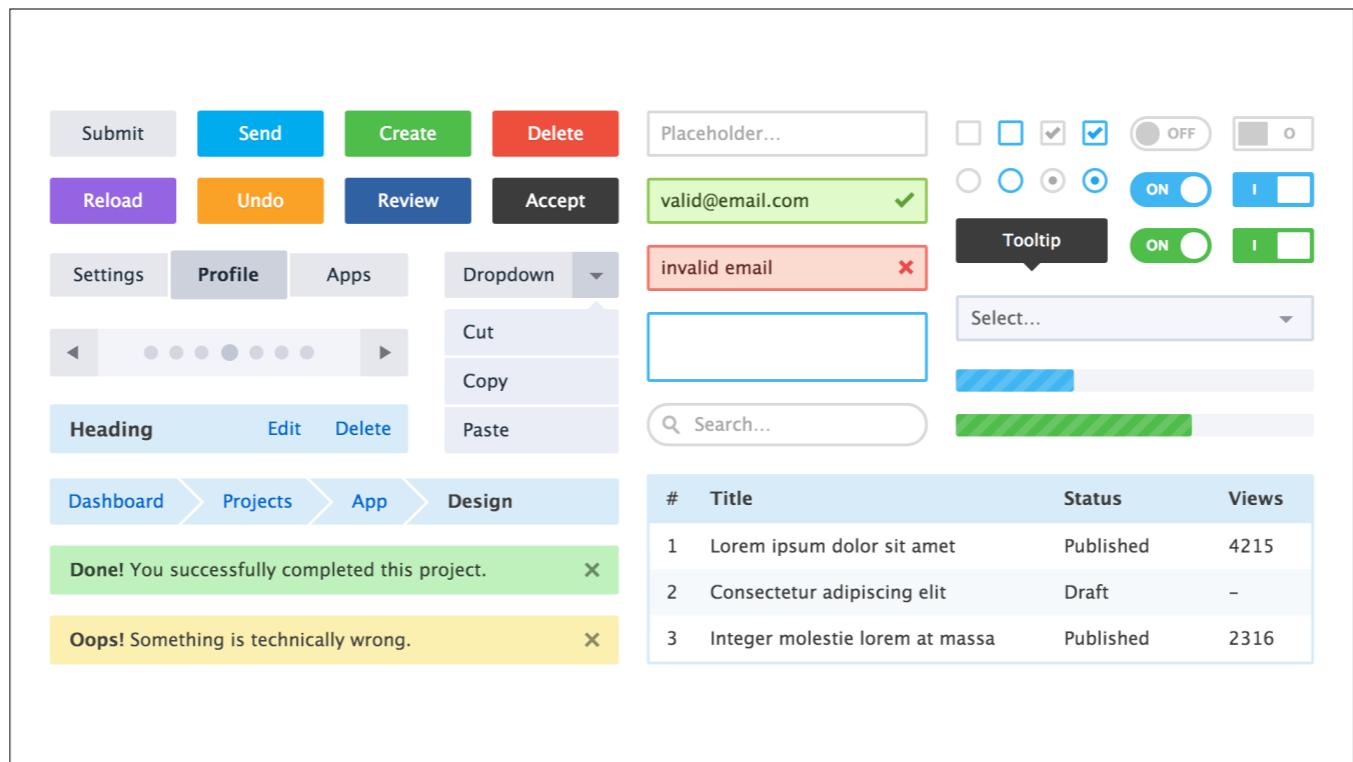
ON INTERFACES

Ok, so we now have proto-personas, we also have user needs documented and we have user journeys.

It'll be the time to put some interface sketches together. Let's look at how might we approach that.



What's an interface? We use these to interact with technology. Naturally, it's not only reserved for the computer.



When it comes to web, we all know what interface elements we can use when building websites. Name them :)

This comes from [CSSFlow](#).

Let's have a little quiz.

"I want to choose a country."

– says the user

What interface element would you use to choose a country?

The image displays two side-by-side screenshots of a user interface for selecting a country.

Left Screenshot (Desktop):

- A form with fields for Address Line 1, Address Line 2, City, State/Province/Region, ZIP, and Country.
- The Country field dropdown is open, showing a list of countries. The entry "United States" is highlighted with a blue selection bar and has a checkmark icon to its left.
- Below the dropdown is a search bar containing the text "united".
- Autocomplete suggestions listed below the search bar include United States, United Kingdom, Tanzania, United Republic of, United Arab Emirates, and United States Minor Outlying Islands.

Right Screenshot (Mobile):

- A title "Select Country" at the top.
- A search bar with the placeholder "Search..." and a magnifying glass icon.
- A list of countries with their flags and names. The entry "United States" is highlighted with a blue selection bar and has a checkmark icon to its left.
- The list includes: Afghanistan, Aland Islands, Albania, Algeria, American Samoa, Andorra, Angola, Anguilla, and Antarctica.

Yup - you would use a list.

"We need our clients to choose multiple different toppings."

– says the pizzeria chef

What element would you choose to allow a choice of pizza toppings?

Veggie Toppings:

- | | | |
|---|--|--|
| <input type="checkbox"/> Red Onions | <input type="checkbox"/> Red Onions | <input type="checkbox"/> Mushrooms |
| <input type="checkbox"/> Green Peppers | <input type="checkbox"/> Sliced Tomatoes | <input type="checkbox"/> Green Olives |
| <input type="checkbox"/> Fresh Garlic | <input type="checkbox"/> Pineapple | <input type="checkbox"/> Sun Dried Tomatoes |
| <input type="checkbox"/> Jalapeno Peppers | <input type="checkbox"/> Broccoli | <input type="checkbox"/> Green Chilli Pepper |

Meat Toppings:

- | | | |
|--|---|--------------------------------------|
| <input type="checkbox"/> Pepperoni | <input type="checkbox"/> Grilled Chicken ** | <input type="checkbox"/> Ham |
| <input type="checkbox"/> Bacon | <input type="checkbox"/> Salami | <input type="checkbox"/> Hot Sausage |
| <input type="checkbox"/> Italian Sausage | <input type="checkbox"/> Anchovies | <input type="checkbox"/> Ground Beef |

"I would like to compare these cameras."

– says the user

What would you use?

Canon PROFESSIONAL PORTABILITY							
	EOS M3	EOS M10	PowerShot G1 X Mark II	PowerShot G3 X	PowerShot G5 X	PowerShot G7 X	PowerShot G9 X
Optical Zoom	-	-	5x	25x	4.2x	4.2x	3x
Wireless Connectivity							
Resolution	24.2 Megapixels	18.0 Megapixels	12.8 Megapixels (3:2) 13.1 Megapixels (4:3)	20.2 Megapixels	20.2 Megapixels	20.2 Megapixels (3:2)	20.2 Megapixels (3:2)
Image Sensor	CMOS APS-C	CMOS APS-C	CMOS 1.5"	CMOS 1.0"	CMOS 1.0"	CMOS 1.0"	CMOS 1.0"
Image processor							
ISO Range	ISO 100–12800*	ISO 100–12800*	ISO 100–12800*	ISO 125–12800*	ISO 125–12800*	ISO 125–12800*	ISO 125–12800*
Video Recording	1080p HD / MP4 format	1080p HD / MP4 format	1080p HD / MP4 format	1080p HD / MP4 format			
LCD Size/Type	3.0" Tilt, Capacitive Touch	3.0" Tilt, Capacitive Touch	3.0" Tilt, Capacitive Touch	3.2" Tilt, Capacitive Touch	3.0" Vari-Angle, Capacitive Touch	3.0" Tilt, Capacitive Touch	3.0" Capacitive Touch
Lens (in 35mm format)	-	-	24–120 mm f/2.0–f/3.9	24–600 mm f/2.8–f/5.6	24–100 mm, f/1.8–f/2.8	24–100 mm, f/1.8–f/2.8	28–84 mm, f/2.0–f/4.9
Shutter Speed	30–1/4000 sec.	30–1/4000 sec.	60–1/4000 sec.	30–1/2000 sec.	30–1/2000 sec.	250–1/2000 sec.	30–1/2000 sec.
Recording Media	SD/SDHC/SDXC Memory Cards (UHS-I compatible)	SD/SDHC/SDXC Memory Cards (UHS-I compatible)	SD/SDHC/SDXC Memory Cards (UHS-I compatible)	SD/SDHC/SDXC Memory Cards (UHS-I compatible)			
Viewfinder	Optional Electronic Viewfinder EVF-DC1	-	Optional Electronic Viewfinder EVF-DC1	Optional Electronic Viewfinder EVF-DC1	Built-in Electronic Viewfinder	-	-
RAW Mode	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Power Sources	Battery Pack LP-E17 AC Adapter Kit ACK-E17	Battery Pack LP-E12 AC Adapter Kit ACK-E12	Battery Pack NB-12L AC Adapter Kit ACK-DC100	Battery Pack NB-10L AC Adapter Kit ACK-DC80	Battery Pack NB-13L AC Adapter Kit ACK-DC110	Battery Pack NB-13L AC Adapter Kit ACK-DC110	Battery Pack NB-13L AC Adapter Kit ACK-DC110
Dimensions (W x H x D)	4.37 x 2.68 x 1.75 in. 110.9 x 68.0 x 44.4 mm	4.25 x 2.62 x 1.38 in. 108.0 x 66.6 x 35.0 mm	4.58 x 2.91 x 2.61 in. 116.3 x 74.0 x 66.2 mm	4.85 x 3.01 x 4.15 in. 123.3 x 76.5 x 105.3mm	4.43 x 3.01 x 1.74 in. 112.4 x 76.4 x 44.2mm	4.06 x 2.38 x 1.59 in. 103.0 x 60.4 x 40.4 mm	3.86 x 2.28 x 1.21 in. 98.0 x 57.9 x 30.8mm
Weight*	Approx. 12.9 oz. / 366 g	Approx. 10.6 oz. / 301 g	Approx. 19.5 oz. / 553 g	Approx. 25.92 oz. / 733 g	Approx. 13.3 oz. / 377 g	Approx. 10.7 oz. / 304 g	Approx. 7.37 oz. / 209 g

A table.

*"Before making a change, users have to log in.
We need their username and password."*

– says your colleague, the developer

What would you use?

 Installer is trying to install new software.
Type your password to allow this.

Name:

Password:

Login

Email

Password

Keep me logged in

[Forgot password?](#)

Do not have an account? [Register here](#)



WEB INTERFACE DESIGN PRINCIPLES

- ▶ Keep it simple
- ▶ Use familiar interface patterns
- ▶ Make sure that content is readable and meaningful
- ▶ Report errors and allow users to correct them
- ▶ Write semantic HTML and good CSS so it works everywhere
- ▶ Use typography and colour wisely
- ▶ Try sticking to the ‘one action per page’ rule

MOBILE INTERFACE DESIGN PRINCIPLES

- ▶ Make it thumb-friendly ('design for the thumb')
- ▶ Remember about the closed loop — always come back to the home screen
- ▶ Make text large enough (12pts minimum)
- ▶ Don't stick too many elements on one screen
- ▶ Test it on different devices (applies to Android)
- ▶ Use animation to indicate action and changes
- ▶ Keep controls in one place
- ▶ Remember about the context

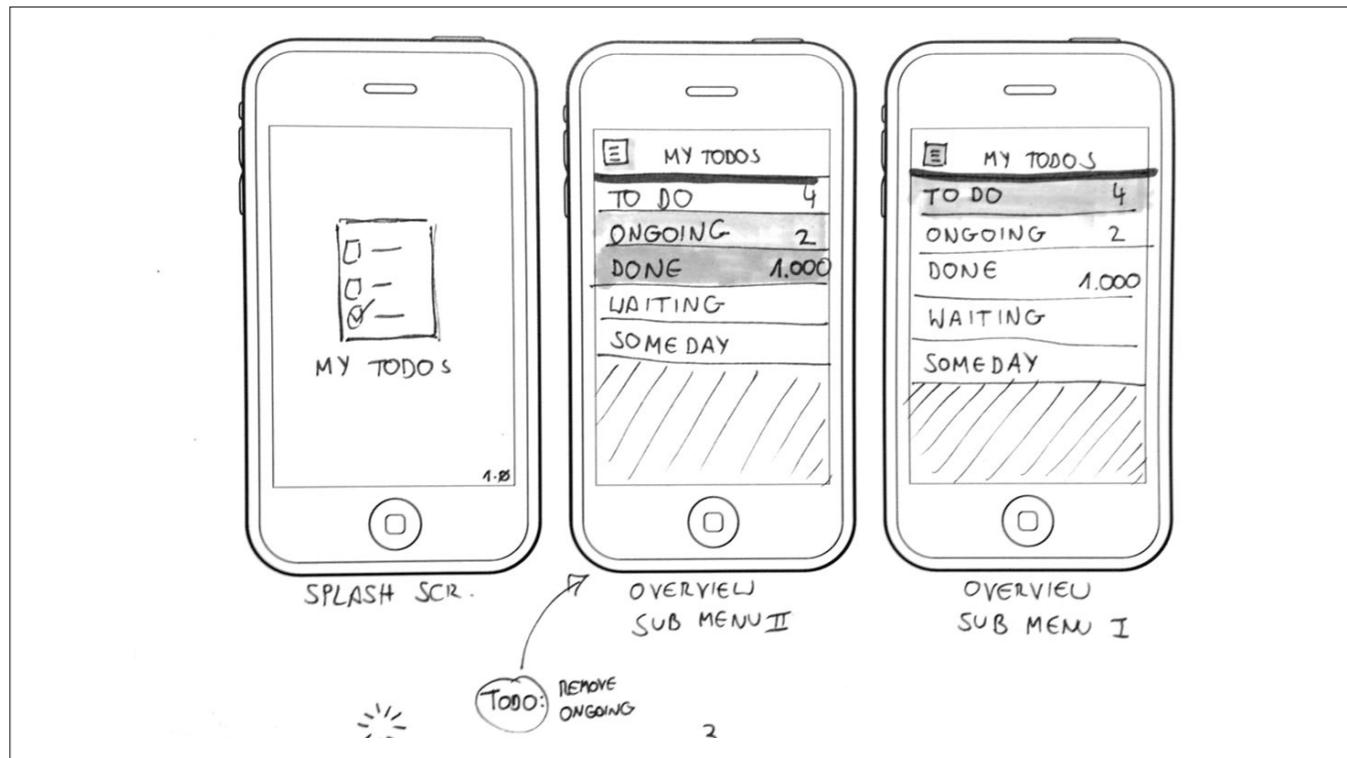
There are some additional points to remember when designing for the mobile

HOW TO APPROACH INTERFACE DESIGN?

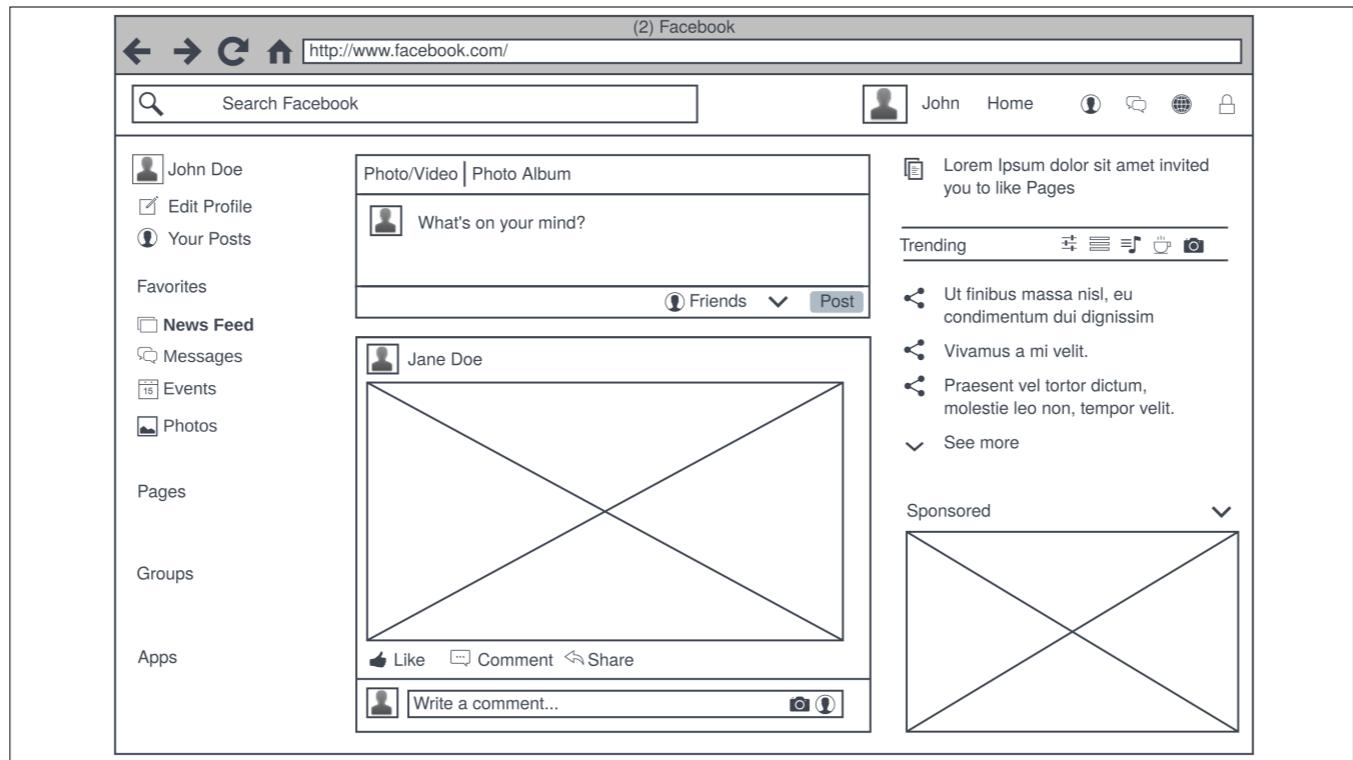
- ▶ Identify functionality
- ▶ Prioritise it
- ▶ Draw it out on paper/wireframe/prototype
- ▶ Test it
- ▶ Build it — but first, improve it. Iterate!

How do we go about designing an interface?

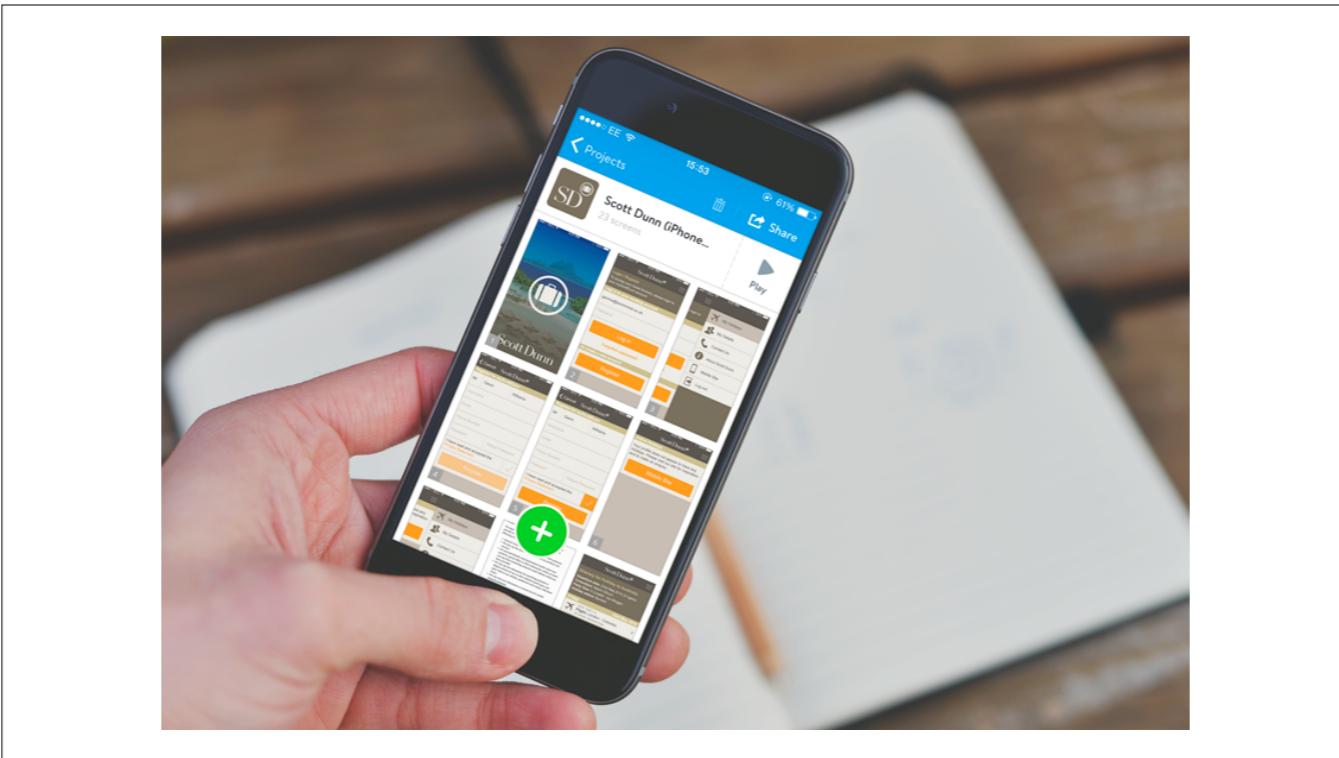
Let's look at our sandwich shop again.



This is a hand sketch. We can use these at the beginning, as these are easiest to discard.
The point is to get something together that supports our identified user needs and journeys.
This we can then quickly test by showing it to someone.



This is a wireframe.



This is a prototype. (Image by [Ecommnet](#))

HOW WOULD YOU GET FEEDBACK ON THIS?

- ▶ Test your prototypes
- ▶ Ask in a non-leading way:
 - ▶ Instead of asking “do you like it?” ask “how does it come across to you?”
- ▶ Talk to multiple people
- ▶ Fix issues on the basis of findings

Whilst proper way of obtaining feedback requires a lot of work, there are some things that could be done quickly.
Pay attention to how you’re asking questions, though.

THE SANDWICH SHOP

Your friendly sandwich shop wants to open deliveries online. As a developer, you have offered to help with the website (they will pay with sandwiches).

WHAT MIGHT USERS WANT TO DO?

1. See the sandwich
2. Read more about it
3. Check allergens
4. Check for nutrients
5. Buy

Imagine, that we're sketching just one page, allowing the user to buy a sandwich.

Why don't we sketch this together. Let's prioritise first.



WHAT DID WE LEARN SO FAR?

SUMMARY

We're doing great. We can now do a couple of things and it will be worth looking at them again just to make sure we're on the same ball.

SUMMARY

HOW WOULD YOU APPROACH A DEVELOPMENT PROJECT?

1. Understand your target group and prepare (proto)personas
2. Identify their user needs
3. Identify goals and draw user journeys
4. Sketch an interface and prototype it out
5. Test
6. Refine
7. Start coding!

How would you approach this?

SUMMARY

WHAT SHOULD WE REMEMBER ABOUT WHEN DESIGNING?

1. Keep it simple — build only things that would matter to users
2. Keep it consistent and use patterns that people are familiar with
3. Use typography and colour
4. Use common interface elements
5. Think of the context
6. Anything else? ;)

How would you approach this?



PRACTICE MAKES PERFECT

GROUP WORK TIME!

Since you know a few methods for designing good software, we should put it into practice.

You will be doing this work in pairs. If you prefer to do it as a group of three, that's fine, too.

We will see if you can apply all things learned today in a practical way.

GROUP WORK TIME!

CHOOSE A PRODUCT

- ▶ New rival to Spotify — for streaming the music on the go
- ▶ Mobile diary app for storing your daily thoughts
- ▶ Online healthy food store
- ▶ An app for cat and dog owners that can keep record of animals' health

Choose one of these products. Ready?

GROUP WORK TIME

WORK TOGETHER AND...

- ▶ Identify core target groups and build two different proto-personas
- ▶ Identify user needs per one of them
- ▶ Choose a key user journey. Go through it and identify potential problems.
- ▶ Sketch one screen of your web app/system

Is this clear? I will be helping on the way, of course. Let's have X minutes.

HOW DID IT GO?

Let's discuss what you've made.



THAT'S US

FINAL WORD

REMEMBER

- ▶ Start with your target group and produce proto-personas
- ▶ Take user and business needs into consideration
- ▶ Always test your work if you can
- ▶ Build only when you are happy with the tested prototype
- ▶ Document user needs and keep it somewhere — review the list periodically

QUESTIONS?

SLACK: #UX

WOJTEK.KUTYLA@CODECLAN.COM

FIND ME ON LINKEDIN AND TWITTER

Do you have any questions?

Thank you.