

## Mobile User Experience (UX) Design (intermediate)

Increase in mobile traffic has made mobile user experience one of the most important factors in the success of a product or website, and it means that the skills involved are absolutely essential for designers, marketers, and developers if they want to keep up with the times.

When you want to start designing a great user experience on a mobile device, using common sense design approaches or simply learning by doing won't be enough. With **61% of users unlikely to return to a mobile site they had trouble accessing** and, even worse, **40% choosing to visit a competitor's site instead**, such uneducated approaches could prove disastrous for your product or website. **A user's comprehension is 50% less on a mobile device, which means that content, navigation, and visual design elements must be twice as intuitive as they are on a desktop.** With hard facts like that, you must constantly keep in mind the point that little screens mean very little room for error. Designing for mobile means taking the rough of the risk with the smooth of the sheer profit potential, and your need to learn the skills involved in creating an easy-to-use mobile user interface and, consequently, a great mobile user experience is nothing less than vital.

### An introduction to Mobile User Experience Design

Much of what they already know about user research, usability testing, UI design, etc. remains true on the mobile platform. However, there are new developments in the UX design field and it is useful for an established UX designer to come to grips with these developments.

#### One Size Fits All? Definitely Not in Task-Oriented Design for Mobile

There is an ongoing assumption that responsive web design is the right approach to take for applications and websites for mobile environments. There's no doubt that mobile first combined with responsive design can bring excellent results in the right circumstances but there's an additional approach to consider: **task-oriented design**; which examines the user's requirements prior to deciding on a responsive approach to design and in many circumstances guides the design team away from responsive.

Responsive design is cost-effective; you begin with **the principle of mobile first** and design a single site or mobile app, which then scales to fit into other screen sizes. Mobile first tends to prevent the overload of the screen when the site is scaled to its smallest size. However, it is worth noting that this is not always the best way to design a site or app to be used across a wide-spectrum of devices.

**Task-Oriented Design** asks the question; what will the user do with this application or website on the specific device on which it will be used?

There are two main tasks for apps and websites:

**Tasks which involve consumption of information:** Think the Guardian or the BBC – these sites are all about providing content (of various types) to be consumed by the user. The requirement for interaction is minimal and whether they are delivered by a mobile website or a mobile app – it's likely that responsive design is going to be OK for sites that focus on information consumption.

**Tasks which involve increasing levels of user interaction:** Once you step away from information consumption most apps and websites require greater degrees of interactivity – this may mean that responsive design is not the right choice. **Adaptive design, where a different version of the site/app is built for each screen size may be a better choice.**

#### Examining the Usage Model for Task-Oriented Design

While there will always be some overlap between platforms in the way that users work – each platform tends to serve a discrete usage model.

**Desktops (and laptops)** – users tend to use these at a desk. They are sat and unmoving for reasonable time intervals. They perform complex tasks more frequently than on other platforms. They are more likely to be able to focus on a single task rather than multi-tasking. The traditional mouse/keyboard combination offers a very effective user interface to support this. The more controls and the more complex the task – the more likely it is that the user is going to sit down and use a desktop or laptop to get the job done.

**Tablets** – once again users tend to be stationary for the time they use a tablet. However, they're likely to use a tablet for a shorter-time period than a desktop/laptop and a typical usage interval might be 15-30 minutes. The keyboard is better designed for work than the smartphone keyboard but it's not as good as the traditional desktop keyboard either. **The more switching between virtual keyboards required – the more likely the user will abandon the tablet and the task or switch to using a desktop/laptop to perform it.** It's a good idea for an app to automatically switch keyboards when data input format is consistent.

**Smartphones** – phones are more likely to be used in very short intervals. They are often used for on the fly multi-tasking (checking Facebook, e-mail, appointments, etc. in the same period of use). The keyboard is much more limited and task complexity is likely to be much reduced when compared to even the tablet.

### Task Analysis For Task-Oriented Design

It's important that design teams conduct a thorough task analysis early in the design phase. This is a much neglected activity but it can help determine the way that a mobile app or website should be built as well as helping to inform the cognitive model, the UI design and the development team's use case creation.

When you take the device into account you might find a process like making a flight reservation goes something like this:

- **Desktop/Laptop** – used for research. Comparing all the options. Examining the places the flights terminate in and checking hotel availability, sights to see, local transport, etc.

- **Tablet** – once the options are narrowed down on the desktop – the user might then want to review their final options on their tablet before booking those flights.

- **Smartphone** – may only be needed to review, forward or change a flight reservation or to check in before arriving at the airport. It may also be vital in the event that the flight is cancelled to be able to choose an alternative.

In this instance, a flight booking app, needs to offer quite different information and functionality to the same user on different devices. A responsive design simply won't get the job done – it would be better to build 3 versions of the same app or website and afford the user the choices and functionality they need to serve their needs.

Of course, this information only comes from user research – **you cannot assume that your users' needs will be different from device to device. They may not be.** However, **if they are markedly different when you start using a task-oriented approach it's a clear indicator that responsive design probably won't serve their needs effectively.**

### **The Take Away**

Task-oriented design asks the question; "Is responsive design the correct approach for my mobile application or website design?" In the case of information consumptive sites the answer is that responsive design is probably the right decision to make. In the case of interactive intensive sites/apps the answer may be more complex. Exploring what your users needs on each platform will help you make better design decisions resulting in better UX with the finished product.

### Responsive vs. Adaptive Design

Mobile web design requires a deliberate focus on users and their tasks. In mobile environments, usability becomes apparent quickly largely because users are using one hand and one eyeball.

**Responsive design:** device friendly design without having to code a new site for each device.

Some responsive design examples: <https://mediaqueri.es/>

A responsive website is simply a design that can stretch and rearrange itself based on the width of the browser rendering the site. In the past, servers would serve up whole different websites after detecting whether a user was viewing their site on a PC or a mobile device. These standalone, mobile-formatted sites were simplified versions of their parent sites, usually with limited functionality to increase page speeds over a slow connection.

When tablets started becoming more prominent, there was a need for websites that sat somewhere between a smart phone and a PC. Then, as smartphones became smarter, you had higher quality resolutions with different layout modes — which ultimately meant a lot to account for if you were developing individual sites for each device.

Responsive design solves the problem at a very low level by resizing and rearranging elements on the screen to fit the user's device. This is usually accomplished through a combination of fluid grids, flexible images and media queries.



Responsive design is a lot easier to implement and maintain as websites evolve with more functionality.

### Progressive enhancement:

Sometimes you will run across a website on your phone that simply does not work. Sometimes you can't navigate the dropdown menus. Other times, there is just too much content.

Progressive enhancement is the practice of catering to the lowest common denominator and selectively adding functionality based on a user's capabilities. Web developers have been using a similar approach for a while in order to ensure features in newer browsers (e.g. Chrome and Firefox) degrade nicely in older browsers (e.g. IE6).

## RESPONSIVE DESIGN + PROGRESSIVE ENHANCEMENT = ADAPTIVE DESIGN

Html 5 + Aria (accessible Rich Internet Applications Specs) allow to build accessible designs.

### ADAPTIVE DESIGN

Levels of Adaptive Design:

1. Text of a Page (Copy well written)
2. HTML semantics (Tags H1, strong, quote)
3. Audio-Visual (CSS-images, audio, video)
4. Interactive JavaScript, Flash, Java
Adaptive Design Possible CSS3

Adaptive design adapts to the context of the use, not just the size of the device. This is a device friendly design that caters to what you want a user to do based on their device.

Ask yourself: how can I enhance this mobile design? Use peer pressure for example (only 2 rooms like this left, 54 people are looking at this flight just now...etc)

Mobile **Call to Action**: some scenario: you want the user to easily bypass a legal agreement on a screen, you want your users to share their purchase on social media in two places: once they have checked out and purchase the product / while they are viewing the product details.

A separate mobile “m.dot” website / from the same website for desktop.

A responsive design usually doesn't drastically change the site's design.

### What is Responsive Design?

**Responsive design** is a way of developing web properties so that the device they are used on determines the way that the site will be displayed. It's normally done using the principle “mobile first” – i.e. the experience is defined on mobile platforms such as smartphones and tablets and then scaled up to larger screens.

The need for this approach is being driven by the understanding that more and more people are accessing the mobile web and in many markets now – smartphone access to the net is much greater than traditional PC access. Mobile devices offer different capabilities to desktop/laptop environments too, such as touchscreens, GPS data, accelerometers, etc.

Finally, **mobile forces designers to think simply**. The experience on mobile is very limited thanks to the screen real estate. Sidebars, adverts, social media inputs, etc. are difficult, if not impossible to accommodate, so the mobile experience needs to be an exercise in simplicity with complexity only being allowed when the screen real estate allows for it.

## The Three Major Principles of Responsive Design

There are three main principles that drive responsive design. Of course, there are other principles that come into play in certain designs but these are three that bind all responsive sites:

- Fluid Grid Systems
- Fluid Image Use
- Media Queries

### Fluid Grid System

In the days before the internet there was print. In print they determined the size of what would be displayed (and where) in absolute measures. When the internet arrived this trend was continued and websites were defined in terms of pixel sizes.

For the responsive web this absolute size won't work. So we use the idea of relative sizes rather than absolute ones.

Let's take a look at an example of how this works, using a mathematical formula:

**Target size / context = relative size**

So let's say you're working on a website with a wrapper that is designed to display the site at a maximum wide of 960 pixels and the device uses a maximum browser window width of 1280 pixels.

The 960 pixels is our target size. The context is the 1280 pixels.

So:  $960/1280 = 75\%$

This principle will also apply to any child elements within your wrapper. So let's say that you're using a two column approach to your layout. The columns are intended to be of equal size with a margin of 20 pixels between them (making each column 470 pixels wide). We use the same formula again for the columns and the margin:

$470/960 = 48.9\%$  (for each column)

$20/960 = 2.2\%$  (for the margin)

In your CSS script you then take these percentages and apply them to the appropriate properties (width, margin, etc.)

### Fluid Images

The easiest way to handle fluid images (**images that scale to fit their container**) is using the CSS command:

```
img { max-width: 100%;}
```

This tells the browser that the image should be a maximum 100% of its pixel value and that it should scale according to its container. The idea is that this prevents an image from being stretched when the container becomes larger than the image – and thus eliminates degradation of the image – and ensures that it will shrink to the container when needed (this also maintains the original aspect ratio of the image).

You may find that you want to use multiple pixel densities as there are now substantial differences in the pixel density of certain screens (particularly on phones and tablets but also on certain laptops and desktops). To make this work – you can set the image to be larger than the container (typically twice as wide). The use of **SVG files** can make this easier as they are the smallest image files and can usually be scaled to any resolution.

For more complexity (lots of images) the responsive images organization offers the following approach:

You can declare multiple sources for an image using the picture element:

```
<picture>
<source media="(min-width: 40em)" srcset="big.jpg 1x, big-hd.jpg 2x">
<source srcset="small.jpg 1x, small-hd.jpg 2x">

</picture>
```

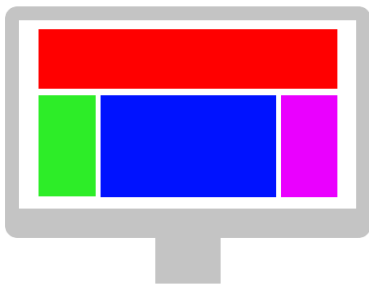
Combined with the srcset and sizes attributes to provide the list of image sources and sizes to allow a browser to select the best image:

```

```

## Media Queries

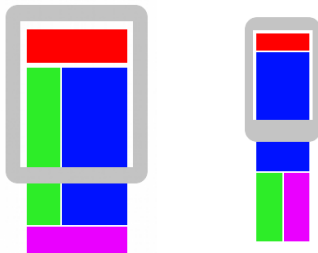
Media queries are designed to alter the layout of the site when certain conditions are met. For example, the two column approach detailed above might not be practical in the screen real estate of a smartphone.



The idea is that you can then use a media query to specify when the screen real estate should be rearranged. You will find that this works best with a “mobile first” approach where you define what you want on mobile and then scale up from there.

The CSS might look like this:

```
@media screen and (min-width: 480px) { /* ..larger screen sizes here.. */ }
```



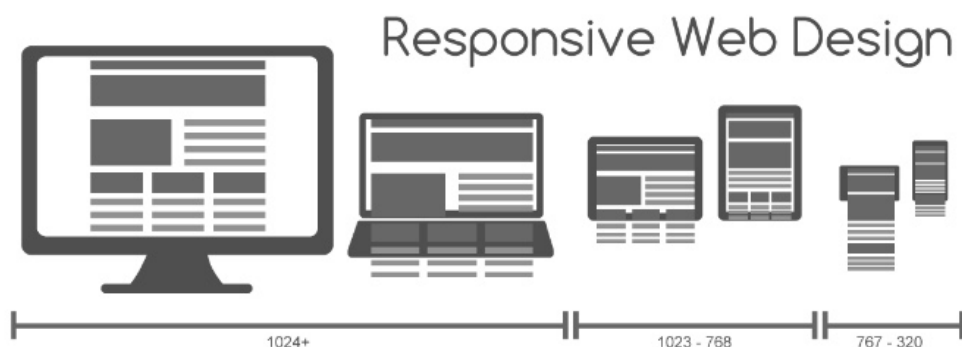
You could then define a series of these CSS commands for each “breakpoint”. That’s when each layout becomes broken on a specific screen size. You’ll need to test content to see where breakpoints occur and plan them. Eventually, you may find you can predict breakpoints based on the screen resolution of a device.

## One More Thing – Frameworks

It can be useful to adopt an existing framework for responsive design rather than “reinventing the wheel”. For example; **Bootstrap is one of the most popular “mobile first” frameworks available online** and you can download it from their website for nothing.

## The Take Away

Responsive web design takes a bit of practice to get right. The three core principles described in this piece will get you a long way towards getting your responsive designs right. Like it or not – catering for multiple devices is here to stay and designers need to be able to do this at the drop of a hat.



**Responsive designs** respond to changes in browser width by adjusting the placement of design elements to fit in the available space.

A responsive website shows content based on the available browser space. If you open a responsive site on the desktop and then change the size of the browser window, the content will move dynamically to arrange itself (at least in theory) optimally for the browser window. **On mobile phones, this process is automatic**; the site checks for the available space and then presents itself in the ideal arrangement.

Responsive design is straightforward. Because it is fluid, it means that users can access your online world and enjoy as much of it on their handheld device as they would on a massive monitor. For this to be true, **responsive design requires a very good conceptualization of the site and a deep knowledge of the needs and wants of the end users!**

**Adaptive Web Design** was introduced in 2011 by web designer Aaron Gustafson in his book, Adaptive Web Design: Crafting Rich Experiences With Progressive Enhancement. It is also known as **progressive enhancement of a website**.

Where responsive design relies on changing the design pattern to fit the real estate available to it, **adaptive design has multiple fixed layout sizes**. When the site detects the available space, it selects the layout most appropriate for the screen. So, when you open a browser on the desktop, the site chooses the best layout for that desktop screen; **resizing the browser has no impact on the design**.

In adaptive design, it's normal to develop six designs for the six most common screen widths; 320, 480, 760, 960, 1200, and 1600 pixels.

### **Standalone Mobile Design = m.dot**

There is also the option to create a mobile-only website (these are usually denoted in the URL bar of a browser using the "m." prefix). This option was once an excellent approach. Designers would create sites for mobile devices, attuning the elements and layout for a dedicated format. Google delivered search engine rankings to mobile sites, but today the same preferences are given to adaptive and responsive sites.

The big drawback of creating a separate site (rather than using different designs or employing a changeable design) is that it requires an awful lot more maintenance in order to keep the two versions of a website homogenous. With no particular incentive to do this, the mobile-only design has fallen out of favour in recent times. It seems unlikely that it will make a comeback anytime soon.

**Modern web design gives us three options to use:** Responsive, Adaptive, and Standalone Design, although standalone has fallen into disuse.

**Responsive Design** lets designers show content based on the browser space available. This allows consistency between what a site shows on a desktop and what it shows on a handheld device. Responsive design is the "traditional" option and remains the more popular approach to date.

#### **Pros**

Uniform & seamless = good UX.

Abundance of templates to use.

**SEO** friendly (Search Engine Optimisation)

Often easier to implement

#### **Cons**

Less screen size design control.

Elements can move around



Advertisements lost on screen.  
Longer mobile download times.

**Adaptive Design** is more involved in that the designer has several fixed layout sizes. It offers an alternative to the “one-size-stretches-to-all” approach.

### Pros

Allows designers to build the best UX for the appropriate device.  
Mobile devices can sense their user’s environment.  
Designers can optimize advertisements based on user data from smart devices.

### Cons

Labor-intensive to create – most adaptive designs are retrofitting traditional sites to make them more accessible.  
Tablets and netbooks can have trouble with site configuration tending to be smartphone- or desktop-oriented.  
Challenging to SEO — Search engines have trouble appreciating identical content on multiple sites.

**Choosing between a responsive and an adaptive design takes careful consideration.** While it might be prudent to stick to a responsive design for the sake of expedience (saving cost, improving SEO, and keeping users content with a seamless experience between devices), it’s crucial to check the pros and cons of both designs in full. Adaptive design can tune in more to users’ varying needs in the field; thus, it’s vital to keep a finger on the pulse of change.

Think about your product or service. Does it access users in a specific setting? What is it about their behavior that you can use to keep them informed and engaged? Remember, it’s not just mobile devices that are getting smarter. In our homes and offices, we have more than traditional desktops. Now, a wide range of smart devices sense and react to the environment, from clocks and heaters to a host of devices that comprise the “Internet of Things”.

**The Context of Mobile Interaction:** Context is critical to mobile design; understanding the big picture of a user’s interaction with a device enables designers to create better user experiences on that device.

**Mobile Interaction and Useful Heuristics:** This model is based very much on the traditional understanding of context including:

- Culture – economic, religious, manners, legal, social, etc.
- Environmental – the noise, the light, the space in which something is used, privacy, etc.
- The Activity – are they walking, driving, working, etc?
- The Goals that the user has – their status, social interaction, entertainment, etc.
- The Attention Span the user has available – continuous (full/partial) or intermittent (full/partial)
- The Tasks the user wants to carry out – make calls, send messages, etc.
- The Device on which the user operates – OS, hardware, capabilities, etc.
- The Connection available to the user – speed, reliability, etc.

**The Proposed Heuristics:** ten heuristics for the mobile design environment:

- Mobile interactions must be user-driven. Mobiles are personal devices and content must be wanted and welcomed by the user and requested by them. A key consideration is non-verbal interaction with mobile devices.
- Mobile derives from previous user models. The past use of desktop computing and even landline telephony will drive expectations even as new possibilities open. Design must take this into account.

- Mobile requires ease of use. Smaller screens, different interactions, etc. demand ease of use be placed at the forefront of mobile design.
- Mobile users do not welcome constant disruption. Design must fit into the user's life and not constantly distract from that experience.
- Mobile users are in constant contact with their devices. This opens the opportunity for interactions of varying intensity and duration. Design needs to consider that not every interaction with a mobile device will be similar.
- Mobile users will expect continuous user experiences beyond the device. There will be a need for user experiences to link to desktops, GPS software, and other services and this should be considered in the design phase.
- Mobile interactions may be smaller parts of greater user objectives. Searching for an address is not a goal in its own right – getting to the address is. The easier the interaction with the device the more likely the user is to reach that greater objective.
- Mobile users will trust peer-to-peer marketing the most. This means new marketing opportunities that lay within the social sphere. Blending the social and marketing possibilities will require careful design consideration.
- Mobile users will expect GPS to enhance their experiences. GPS capabilities will deliver the ability to interact with the world beyond the mobile.
- Mobile may lead to a “dumb terminal” with further processing carried out elsewhere in the network.

Increasingly the lines are blurring on where devices are being used and how they're being used in unison. You should see devices as location agnostic and instead associate them with purpose—I want to check (mobile), I want to manage (desktop), I want to immerse (tablet). This shift away from objective context toward subjective context will reshape the way we design experiences across and between devices, to better support user goals and ultimately mimic analog tools woven into our physical spaces.”

The **mobile context** is one of shorter interactions “checking” where you might dip in and dip out of a social network, seek an address or scan your e-mail but don't want to do anything particularly complex.

The **tablet**, is mainly a leisure device (though it has its own enterprise context too) and provides a chance to immerse in an experience without becoming overly interactive with the device.

Finally, the traditional **desktop/laptop** platform is where people manager their overall experiences on(and indeed, off)line.

This contextual model is based around the user's intentions rather than their physical location and while there may be some shift between levels on each device; the main intent of each platform is fairly clear.

### **Making Use of the Crowd – Social Proof and the User Experience**

Social proof (sometimes referred to as informational social influence) is a psychological concept. It refers to the tendency of human beings to follow the actions of others when making decisions and placing weight on those actions to assume “the correct decision”. It's a concept that can be used in product design for the Internet and mobile web to help drive user decisions in the direction that a business wants them to go.

**Social proof demonstrates that we are likely to converge on a course of action when others provide that action for us.**

**Social proof is used for two reasons in user experience design:**

**To deliver credibility.** If other people find a source useful or credible – we are more likely to believe that source may useful or credible for ourselves. An indication of this belief in other people that is displayed within the site or product – boosts credibility for other users.



**To promote adoption and/or acceptance.** Volumes of people subscribing to a Facebook page or Twitter feed can encourage others to do the same. Large amounts of people doing something is a psychological indicator to people that they should do the same thing.

### **Usability Testing**

If you're using social proof to try and increase your credibility – you want to conduct some survey questions that enable you to gauge user confidence.

You can also use usability testing to ensure that the volumes of social proof that you're providing aren't overwhelming users. Too much information can be worse than none at all. Users are happy to be convinced by social proof but there's only so much data that they can process easily.

You might also want to determine whether users even notice your social proof. After all, position is important and it's quite possible that your users won't benefit from social proof if they don't even notice it's taking place. **Eyetracking** software as part of usability testing can help you decide whether users are noticing social proof and whether they're paying any real amount of attention to it if they do notice it.

### **Load Time Testing**

Social proof, particularly when it comes from social media or other plugins, can slow down the performance of your website or application. You want to keep measuring load times because poor load times can cause abandonment even before the user has had a chance to interact with your content.

### **Social Proof and the Mobile Web**

Social proof is harder to integrate into the mobile user experience. Average user ratings are popular in app stores, for example, to indicate the value of a particular app and some brief summary reviews are also displayed. However, it's vital to keep an eye on the available screen real estate and be certain that you're not overwhelming users when it comes to delivering social proof on mobile. Usability testing is a must.

### **Prototyping for Mobile Apps**

The design phase for mobile applications should include a prototyping stage. It is at this point that users can "play" with your ideas and concepts and give you valuable feedback that shapes the final designs before you begin development. This can save time and money in development and create products that offer significantly better user experiences than ones that move from concept to production with no evaluative stages in between.

Prototyping is the act of creating a model of a product so that it can be tested by users before you expend valuable development time on creating the actual product. In the sense we are using it prototypes encompass everything from simple sketches of the product interface right through to dynamic interactive computer models of the product and stopping at wireframes on the way as an interim prototype.

There are numerous software tools on the market that allow designers to quickly and easily create prototypes. In fact, some of these packages are so easy to use that they may eliminate the use of sketches in your prototyping procedures. Why hand draw when you can create screens, with a single click, which mirror the mobile platform you are working on and then you just drag and drop functionality in to them?

With that in mind we're going to take a quick look at 5 of the most popular tools, today, for creating prototypes for mobile app design.

**Balsamiq** (to sketch and create wireframes) **Justinmind** (great prototyping tool which also includes simulations for things like gesture control, tap and hold, swipe, etc.) **Moqups** (free and easy, but only produces wireframes though; if you want interactivity – you'll need to choose a different platform for your prototypes.), **Proto.io** (purely mobile prototyping platform. It allows you to quickly develop prototypes (and simulations) that reflect the final product. There's no coding required and it can be accessed in almost any browser).

# The Elements of Mobile UX

## A User-Centred Approach to Mobile Design and a 5 Stage Process for You to Use

Taking a user-centred approach to mobile (and other) design can help you keep in mind the outcomes that you intend rather than wasting time on unrewarding designs.

The mobile web is a description intended to distinguish accessing the internet on a tablet or a smartphone as opposed to a regular PC or laptop

There are advantages to delivering a mobile web experience:

- There is the opportunity to cater for the specific users' needs at the right moment and in the right place.
- The mobile web can be accessed in places where the Internet is not easily accessible from other devices.
- Development for the mobile web can be cost effective and even be cheaper than standard websites.
- There is potential to reach a much larger user base (there are more smartphone owners than desktop and laptop owners).
- There is the opportunity to reach a much wider geographic area (in developing nations smartphones are often the only way for a user to access the internet).

## **The User-Centred Mobile Design Approach = UCD**

There are **5 stages of the cycle** (all development is assumed to be cyclical with products going through multiple iterations over a lifetime).

You **assess the situation** as it is at the moment (maybe that's "we don't have a mobile website" or maybe it's "our mobile app is not performing as we'd hoped it would", etc.).

Then you work out what it is that your **users need** from you. Once that's done, you sit down and **prioritize features for your mobile platform** and then you work on **designing** those features while ensuring that you've put "mobile first" and finally, **you review and refine** the design.

You need to think about whether a fully functional mobile app or website is really what you need – or is it possible that your users only need a small subset of functionality on the move and that they'll do most of their work at a desktop?

Is it possible that your user base is less interested in a mobile web experience and more interested in an enhanced experience with the products and platforms you already offer?

## **Understanding Your User**

Before you go rushing out to do any design or feature prioritization it's important to understand your users better and in a mobile context. You need to know things like:

- How do they prefer to access the internet?
- How much time do they spend interacting with your site at the moment?
- How much time do they spend online using a mobile access point?
- What features would be critical for providing a mobile experience?
- What are they frustrated with in your offering at the moment that could be done better via mobile?
- What devices are they using to access the mobile web?

If you have more than one user persona – you'll need to answer these questions about each user group

Don't forget that too much complexity early in the product lifecycle can be a major drawback. Prioritizing needs also means not being afraid to hold back ideas to future releases. **A great minimum viable product can often be better than an overly complex one.**

Josh Clark, the author of Tapworthy- Designing Great iPhone Apps, offers **three categories for mobile web access**:

**Microtasking**: When the user interacts with their device for brief but frenzied periods of activity

**Local**: When the user wants to know what's going on around them

**Bored**: When the user has nothing better to do and is looking to be entertained or otherwise diverted

Keeping these categories in mind can make it much easier to design for the user's needs and focus on what makes mobile different from other access platforms.

Sketch and prototype in early iteration phases. Make sure to test them with users. Get feedback and iterate rapidly.

Don't forget to ensure your site is compliant with W3C standards.

**Then go back to the beginning and iterate.** It's why the user-centred design process for mobile is cyclical. Well, just like all user-centred design processes!

### **Mobile First and the Power of Mobile Computing**

Where do you start? Do you begin with the desktop experience and scale down? Or do you begin with the smartphone and then scale up? The concept of "mobile first" suggests that you might want to begin with the smartphone.

**Mobile first**, is the concept that you should design for the mobile platform first and then scale the experience up to other platforms.

#### 3 reasons for designing for mobile first:

##### **Mobile has Exploded:**

Billions use mobile Internet. In developing nations, in particular, the only internet connection that many people have is the one that comes with their smartphone. Even in the developed world more than 50% of online interactions are carried out with smartphones.

Desktop and laptop markets have shrunk and while the tablet market appears to have stabilized – there are more tablets and smartphones sold each year than traditional internet access platforms.

##### **Mobile Forces You to Focus:**

Screen sizes and resolutions have come on in leaps and bounds. You need to focus on the most important actions and tasks for your users. You need to ensure content is minimal and delivered by priority to the user.

There's much more room on a desktop screen and you can afford to add in extraneous clutter there and extra features but smartphones don't give you that luxury. The user-experience on smartphones is also determined by how they are used – on the move and in short, sharp bursts of activity which is very different from sitting at a desk and browsing for hours on end.

##### **Mobile Offers New Capabilities**

On the desktop, the Internet is the standard way of interacting with most businesses but in the mobile landscape the app is king.

That gives design and development teams brand new capabilities to deliver content in new and exciting ways. Then of course, there's all that built in functionality on a smartphone that doesn't exist (and wouldn't be very useful) on a desktop. GPS, accelerometers, pedometers, new input devices (touchscreens, gesture controls, eye tracking), etc.

The mobile app experience can be very different from the one delivered on the desktop. The former allows you to tap into a whole new world of inputs and functions – the latter allows you to deliver larger more complex experiences.

Designing with mobile first as a principle lets you focus on the power of the mobile computing platforms available rather than limiting you to a "miniature desktop experience".

### **A Note of Caution:**

While mobile first can be an excellent strategy – it is, of course, important to do your research first. There are applications, particularly in the enterprise environment, where mobile is a minor secondary concern compared to the desktop.

Mobile isn't the right platform for every product and you need to be certain that "mobile first" is right for your user base before you pursue it as your key strategy for development and design.

### **Content is King for Mobile Too**

The primary consideration of any UX design must be content. Content is the reason people arrive at a web site or download an app – design is important but only if the content is up to scratch. Learning how to ensure content remains king is one way to ensure the longevity of your website and application design.

Two of the world's leading websites are Reddit and Wikipedia. It would be hard to argue that either of these sites excels in aesthetic quality or visual design but what they do have is content – more content than nearly any other site on the internet. This content keeps the users engaged and keeps them coming back for more.

Sure, a little pretty design wouldn't go amiss – there's nothing wrong with making a site content rich and visually appealing but users will opt for content over aesthetics when they have to make a choice.

**How do you create content:** Many websites subcontract their writing but there's nothing wrong with the design team creating their own content either. In fact, it can be incredibly valuable to have someone from within the organization create content and then pass it on to a writer to "beef up" afterwards. There's no substitute for knowledge in content.

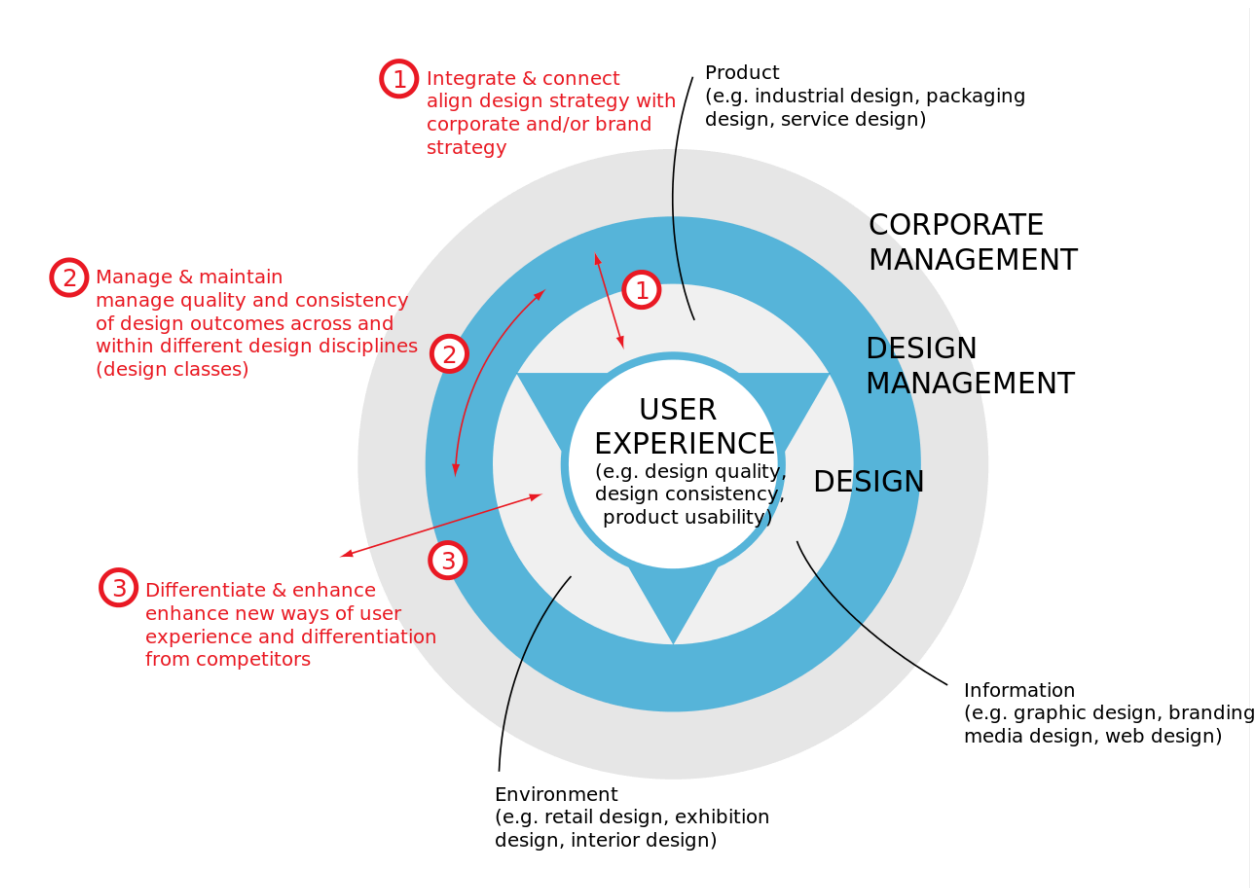
If you're redeveloping a site or an application – don't forget to raid the old version for as much useful content as possible (though it should also be edited or rewritten to polish it before implementation).

### **Design Promotes Content Not the Other Way Round**

If content is king, then design needs to be carried out with delivering that content in mind. There's no point in developing a beautiful new home page design if you find that when you add the copy; there's not enough space for it to have the appropriate impact.

You want to develop key content before you begin design work. Don't use placeholder text and find out at the last minute that your placeholder was 1/3 the size of the information you want to convey.

Then examine how you will highlight content in your design too – some words, phrases, text blocks, quotes, etc. will need different levels of prominence from the standard copy. It's important to work out how you will handle this early or you risk everything blurring into one amorphous blob when the design is complete.



### Mobile First and Content First

Mobile first is a brilliant design strategy and one that can be enhanced by the use of a content-first approach to design. By developing content early, you can start to examine the breakpoints for responsive designs. If the breakpoints for the content aren't manageable – you may need to rewrite content until it works on mobile first but it's going to save a ton of time finding this out in the early part of your project rather than waiting until the end and finding that the content overwhelms the design.

As screen sizes get larger, you can then examine how you can let the content grow to accommodate new breakpoints. **Starting with mobile first makes this much easier than trying to chop things from a large, cluttered desktop screen to get a tablet experience and then chopping more to get to mobile.**

### Functionality and Mobile Design – Don't Shrink the Screen, Focus on the Tasks

Mobile design and design for other devices such as laptops and desktops does not mean producing the same product, just smaller. It means focusing on what matters to the mobile user and delivering a differentiated user experience. To do this successfully, you need to focus on tasks and how they are executed.

Mobile is different and not a desktop experience. For example; the screen real estate is more limited on a smartphone, connectivity may not be guaranteed, battery life is precious, etc.

Then there are the benefits of these devices to consider; they are more personal than a desktop, they're always on (when battery permits) and you can talk directly to the user through them. They have motion sensors, GPS, accelerometers, and more too.

### Focus on Mobile

The limitations of screen real estate mean that less is more. You need to keep features to a minimum. You have to focus on the key tasks the user wants to carry out and you will have to drop stuff from other platforms to make mobile work.

## Create Value

Everyone and their donkey is releasing a mobile app today. If you can't clearly distinguish your app and the features it provides from everyone else's – yours will get lost in the black hole of the app stores.

## Personal Functionality

If mobile devices are personal then the functions they deliver should be personal too. Make them fun, customizable and integral to your user's lives. Getting tasks done doesn't need to be a chore.

## Consider the Context of Use

When will your user access your functionality? When they're bored? When they're busy? When they're lost? Make the functionality fit the situation and you're on the road to relevance to your users.

## Functions that Fit the Device

The functions you offer need to follow patterns that deliver the experience on every device in a consistent way:

- **Keep the user in the picture.** It may take time for data to process but you need to let the user know when you've received their data and how long they will wait and you need to do that immediately.
- **Pay attention to the detail.** Apps need to deliver seamless experiences where users can carry out tasks efficiently. They also need to look and feel good to use.
- **Watch the interface.** Users are all thumbs, literally, on smartphones. Can your tasks be easily completed with someone's thumbs? No? Then rethink them.
- **Keep things intuitive.** This is back to the thumbs issue really – keep buttons, tabs, boxes, etc. to a minimum and keep content easy to access and navigate.
- **Keep controls consistent and at the bottom.** This lets the user focus on the content and manipulate it as they need to. Task oriented design means delivering the meat first and the vegetables second.
- **Reduce user input where necessary.** Pay attention to whether **auto-correct** is a help or hindrance on the device for your tasks. Make sure that you help the user pick the right keyboard to get the job done. Don't forget to support a landscape orientation – particularly if you want a lot of typing done. Think about providing an orientation lock to make this easy.

## Think about Navigation

A lot depends on the complexity of your task but navigation on mobile can fall into these categories:

- **None at all.** If you're releasing a weather app for your neighbourhood – a single screen could (and should) deliver all of the task for the user.
- **Simple tabs.** When you need to break consistent content up into several areas. (Think Twitter)
- **Drill down menus.** When you need to deliver higher levels of complexity. Group content and make it easy to move through the hierarchy. (Think your settings menu on your smartphone.)

## Get into Gestures

If you're going to enable gesture control (and that's down to how your users like to complete their tasks) you need to consider:

**Discoverability.** How will the user know they can use gestures to complete their tasks?

**Two-handed gestures.** They can be useful but they can also be a hindrance. If you've got a coffee in one hand; how will the user complete their task with only single handed controls?

## Keep Communicating

It's important to keep the user informed as they go about their tasks. You need to:



- **Have a method to deliver feedback.** From the tactile (think the vibration function) to the visual (progress bars) – you need to keep users aware as to whether things are going well as they execute their task.
- **Ask for confirmation when necessary.** Modal boxes are annoying but the exception is when you want to protect the user. If you're asking if someone wants to delete a photo, for example, you can ask if they're sure before you carry out the action.

### Keep on Task

If a user exits your app without completing a task; when they open it back up – they should be brought right back to the same task in the same place. Make it fast and easy to restart a process and your users will love you for it.

### Consider Graphics

The icon you use for your app – should ideally be representative of the key task your users want to carry out. Make it blindingly obvious at a glance what your app does and it will be much more likely to be downloaded and used.

### Consider the First Time of Use

Keep things simple and focus very much on key tasks when your app first launches. You can add complexity over time. If it's confusing or hard to carry out a task the first time that your app is used – it's likely that it will never be used again.

### Micro-moments: Are you designing for them?

We need to equip ourselves with methods and tools to research our users' lives before moving to the solution space. We need to understand our target users' culture. In these fast-paced times, it can be hard to get time to stop, stand back, and look at what's going on in the big picture.

**Micro-moments** during which the users get hold of their phones to accomplish a specific task.

Most of these can be categorized under the "I-want-to-share" title that currently takes up so much of our everyday lives. I want to know - I want to do - I want to buy - I want to go

As UX designers we try to bridge several worlds, connecting those of the target users with the business goals and all this mediated by technology. Therefore, we need to be able to look at both sides of the screen: what users are doing and what the company wants to offer them. With the idea of micro-moments, we have new touchpoints to consider, understand and design for during the customer journey map.

Most often, the journey starts at the moment the **user starts to consider a category or a product**. This earliest stage determines what the user wants or needs. Then, the journey takes the users to where they're **evaluating and making comparisons among available options**. The third stage is when **users actually buy something**. Last comes the **after-purchase period**; which some call the: the **Ultimate Moment Of Truth**. **This moment can then be defined as the instant when a customer creates content based on an experience with a product or service and publishes it in their community for others to find.**

Throughout this journey, there are lots of touchpoints that we can translate as micro-moments. If we – as designers – can identify them, we will be catering for the particular types of needs of our users. It's about creating the content or tools to enable people to accomplish whatever they want in that particular moment. To do so, Google suggests the following activities:

- Make a moments map.
- Understand customer needs in-the-moment.
- Use context to deliver the right experience.
- Optimize across the journey.

- Measure every moment that matters.

What do these suggestions recall for you? As always, it's about placing user experience approaches up-front in the business strategy. It's about starting with the users' needs and contexts and moving from there, iteratively and always with "reality checks".

As UX designers, we have plenty of touchpoints on the road where the user considers a product/service, evaluates and compares like items, chooses and buys one, and then uses it happily ever after. Under these phases, **we need to consider how micro-moments play a role in the user experiences. To achieve this, use customer journey maps to understand your users, their needs in-the-moment, and use context to deliver the right experience, make every part of their journey as best as it can be, and measure every important moment.**

### **The Heart of the Matter, Information Architecture in the Mobile Age**

In general terms the way you design information architecture (IA) for the mobile environment is very similar to the way that you design the IA for the desktop. The approach you use is near identical in fact but there are some important things to bear in mind when you do so:

**Deliver what's important on the mobile platform** – that means you should focus on what's important to the user when they're using a smartphone. That doesn't mean that your users will necessarily be using your app when they're on the move or on the street – you need to research the way that they will use the app. That, in turn, lets you focus on the content that's of value to the user and cut down the volume of data that you present on mobile.

**Content should be focused on mobile** – that's smaller, simpler and easier to handle content than on the desktop. The limit of screen real estate makes large volumes of content more confusing on mobile than on a larger monitor.

**Navigation should be simple** – that means limiting the number of menu options and keeping things focused on what's important to the user.

### **Information Architecture Design for Mobile**

The process then follows a simple process:

#### **Consider Click Investment**

You want to build your information architecture hierarchy so that the **minimum number of clicks are used to reach the most popular content**. Less popular content can be farther away but not too far away. If people abandon a desktop website in seconds because of poor loading times – it seems likely that they would abandon a mobile website if they're forced to wait and load multiple screens to get to the content they most value.

You can reduce levels of frustration with click investment by ensuring that you communicate the process clearly throughout the task. That means setting user expectations as to what will happen after each click before they make it.

#### **Keep it Simple Stupid (KISS)**

This comes up a lot in design and there's no exception when creating information architecture for mobile. The simpler things are – the fewer issues with the overall experience.

So limit user choice; if your user research shows there are areas of your content that aren't valued by mobile users – eliminate that content. You can always offer it via a desktop experience on mobile for the occasional user that really wants that content. The simpler your IA – the less chance that the user becomes frustrated with it.

Then plan your app's IA – use a simple drill-down structure and nest content into clear, easy to understand and well-defined categories. Make sure that you understand the path from top-level information to sub-levels.

**Keep categories to a minimum** - That doesn't mean abandon useful data but do think about the fact that people get lost more often the farther they drill down for information. 5 levels is enough on the desktop, with mobile you almost certainly want to aim for less.

**Keep content and categories together** – no empty pages, no empty links. If a user clicks on a navigation item, it must deliver some relevant content too.

**Keep links to a sensible number too** – don't overwhelm the user with dozens of links. Ideally, keep the number of links displayed to fewer than 10.

**Prioritize by popularity** – “deck placement” is vital to controlling the simplicity of the experience. The more popular something is within the app or site; the more vital it is that you make it easy to get to. However, this can have an impact on the success of new features or content; as it will be placed lower in the deck and become harder to find. You may need to strike a careful balance between the two.

**Label navigation items clearly and concisely** – this applies to links as well as menu options; this is very important on phones with very small screens.

### Think About Input Devices

Not all smartphones are touch screens. You need to ensure that you maintain the “tap worthiness” of navigation items (minimum 30 pixels) whilst still providing quick methods for non-touch screen users to get to data (0 to 9 as shortcut keys will be better than 1 to 10 as they match the numeric keys on the phone).

### Provide Clear Navigational Cues

Navigation on mobile is more complicated than it is for desktop. Make your “breadcrumbs” explicit – if you use a back button, put a label on it to tell the user where it takes them. Make sure you have a home page icon that is clear and distinct and will take the person back to the beginning of their interaction.

### Getting into the Mobile App User's Mind – User Research for Mobile Applications

Mobile user research is the key weapon in the UX designer's armoury to conquer some of that market.

### Qualitative User Research Methods for Mobile Apps

Qualitative research is concerned with identifying ideas and trends. It is, by definition, not measurable research (you are not collecting numbers) but rather research which provides a flow of concepts that the researcher must analyse to gain insights for product development.

There are 3 key research methodologies for qualitative user research for mobile apps:

Lab Studies

Journal/Diary Studies

Observational/Field Studies

#### Lab Studies

Lab studies are designed to test whether the user can actually use the product. A lab study is conducted using a mobile device in an environment where the user can be observed by the research team.

Lab studies are great because they allow the moderator to interact with the subject and ask questions to gain further insights. When set up with high-end equipment the whole team is able to observe the tests and they can easily gather visual cues to the way users feel.

There are issues with conducting lab tests in that they do not bear a 100% resemblance to the real world. They may suffer both from moderator bias and the need that participants often feel to please the moderators with the way that they approach the task.

#### Journal/Diary Studies

A journal/diary study poses the questions; “When do users actually use our app? In what contexts do they use our apps?” and they involve the user writing down the when, how, why and other observations each time they use the app.

Journal studies can be very good at determining how a user actually uses the app and for delivering honest feedback to the research team.

Their biggest drawback is that they rely on the user remembering to note down each use of the app. They also require a lot of management to ensure that they provide the most value.

### **Observational/Field Studies**

You use a field study to determine; “When are users using the app? What context do they use the app in?”

A field study requires a researcher to observe (and usually film) a subject as they interact with the app.

This can generate a lot of useful data and is pretty easy on the participants, however, field studies offer no guarantee that the people you are observing are “typical users” who represent the whole user base and they can be intrusive and awkward to carry out.

### **Quantitative User Research Methods for Mobile Apps**

Quantitative research results in “hard numbers” which can often be more powerful when convincing stakeholders to make a decision than the outputs of qualitative research. However, it’s worth remembering that quantitative research normally answers different questions to qualitative research and both are necessary to develop the best user experiences.

There are 4 main types of quantitative technique used in mobile app research:

Online Surveys (including Behavioural Tracking)

Behavioural Analysis

Automated Logging Activity

Experience Sample Methods

#### **Online Surveys (including Behavioural Tracking)**

Online surveys will involve using a questionnaire combined with automated data collection reflecting the user’s actual behaviour. They are designed to answer the questions; “What do users think of our product? What do they perceive the value to be?”

Online surveys are incredibly cost-effective and can create large amounts of data on a geographically diverse basis which, in turn, leads to statistically significant sample sizes and results.

The down side is that there is no direct contact with the user in large processes like this and you may find that you can identify patterns but can’t deduce the “why this should be?” from them.

#### **Behavioural Analysis**

Behavioural analysis works to determine the answer to the “how do people use our app?” question. It is conducted via installed software on the users’ devices that track the way people navigate through the app and/or how they conduct the tasks on your app.

This can be very useful to be certain **about what people do as opposed to what they say they do** (a perennial problem for researchers) and to understand traffic patterns.

Conversely, the data will not show the user’s context or their intentions. This is why behaviour analysis is often conducted alongside a survey – to try and develop a clearer picture of “why”.

#### **Automated Logging**

Another way to answer the “how do people use our app?” question is through automated logging – that records events that are generated on the mobile device and without user input. This is a very good way to ensure that interaction data is objective and consistent. Like behavioural analysis, automated logging is limited in the ability to interrogate context or intent.

#### **Experience Sampling Method (ESM)**

This method is particularly useful in mobile user research because it consists of sending automated messages to the study participant at the point they carry out an action to ask about their experience. It helps answer two main questions; “How does the user use our app?” and “What do they think about that?”

There is a risk of introducing cognitive bias into the captured data and studies need to be well-designed to minimize this potential.

### Mobile Usability Research – The Important Differences from the Desktop

When you test a mobile product with users, you need to examine that product in the right context because the mobile environment isn't the same as the traditional desktop environment.

- **Mobile is mobile.** It might sound silly but when users aren't sat still – their context constantly changes. Their location varies, their ability to pay attention varies, their connectivity may drop in and out and even the way they hold or deal with the device itself can change.
- In light of the ever-changing context it should be apparent that **users are going to prefer short, simple interactions on mobile.** It's not to say that they will never carry out long, complex interactions but these will be rare.
- Mobile has led to a wealth of new **personal data generation.** GPS positions, pictures of the user, their friends and family, communication data between colleagues, friends, etc.
- **Mobile creates real privacy concerns.** Not only are devices more susceptible to being lost and/or stolen but they also contain far more personal data than a desktop typically would.
- Mobile may be mobile but it's not a single experience. **Everyone has a different device with slightly differing capabilities.** Then there's the difference in browsers, operating systems, etc. a broad range of capability means more testing and a need to rely on W3C standards.
- Mobile also encompasses tablets. That's a different experience too – **tablets aren't desktops but they offer more “desktop like” functionality** than a smartphone does.
- **Mobile offers different forms of input.** The mouse isn't the mobile experience – touchscreens, voice inputs, user movements, etc. are the mobile experience.
- Mobile brings more users than desktop computing but **users with profoundly different needs than desktop users.**

### What does this mean for mobile usability testing?

Constant change of context for mobile users is going to have an impact on research. Field research becomes more important for mobile testing because it offers “real world” insight that may be difficult to create in the lab. But lab research becomes equally valid to field research when steps are taken to try and create realistic field conditions and lab based research can save both time, money and resource compared to conducting work in the field. It was also shown to deliver better results in terms of reported issues.

### What Can We Do to Simulate Real Life in the Lab?

This then leads to a question of how to simulate the field conditions within the lab to ensure that our research is accurate:

- Play music/video in the background while the user works to carry out their tasks
- Use people as temporary distractions by having them walk into the environment or interact briefly with the user
- Using stop-start techniques (for example, using a bell to indicate the user should start a task and to take a break from a task)
- Using treadmills to simulate the effect of using the device on the move
- Setting out a walking course with obstacles to simulate the effect of using the device on the move

In short, you need to help users capture that changing context. If you let a user operate the mobile device in the same way as they'd operate a desktop – your research will be flawed. However, if you can break things up a little; your lab based research may offer even more value than field based research and it will be easier to carry out.

### Limits of Prototypes in Mobile Usability Research

Low-fi prototypes need careful implementation in mobile usability research. There's still nothing wrong with using them for initial feedback but it's important to note that they suffer a serious limitation in that **gesture controls can't be realistically tested with this kind of prototype**. So, if you're going to use low-fi prototypes you need to treat the results with a little caution until you can test gestures with higher fidelity prototypes.

### Heuristic Evaluations

The most common usability heuristics still apply to mobile usability research – however, there are more heuristics which can be applied to usability testing in the mobile environment. It's important to familiarise yourself with these heuristics to ensure that your testing provides maximum value.

### How Do We Capture Data in Mobile Usability Research?

There are dozens of tools for capturing data from these studies. 4 which are free to get started with:

**UXCAM** works in both Android and iOS environments – you can run up to 100 tests free of charge before you have to sign up for a paid plan. It enables you to capture exactly what is going on on the user's screen to capture data. You can also use it to capture the user's facial emotions (using the phone's camera) at the same time. Though we'd warn that this requires consent in advance – you shouldn't spring anything on your users after the testing.

**Watchsend** offers 100 free recordings of what's going on on screen before you have to pay. It's easier to integrate with beta-releases than UXCAM which means it can be adopted at an earlier stage of the product development lifecycle. There's no facial recording at the moment though.

**Userzoom** is the big daddy of the usability testing world for mobile but you have to ask them for a free trial – there's no quick option to get started. It has the best selection of testing options and analytics. It can go beyond just screen recording and into click-testing, tree testing, card sorting, etc.

**Skala Preview** is a free to use app for iOS and Android and can be downloaded from the relevant stores. However, you also need to have Skala Preview (for Mac) to make it work. It's best used for testing tap point sizes and ergonomic functions. It has the nifty feature of allowing live updates if you use Photoshop CS5 – so you can change designs to test for mockups on the fly in real time.

### Remote Research Methods for Mobile Applications

Users of mobile apps are going to face continued distractions when using their smartphones and there's no better simulation for these distractions than monitoring the user in their ordinary environment. There is a strong business case for remote research for mobile apps and it should help designers build better mobile user experiences.

Remote research is used in many UX projects because it allows the researcher to **learn about the use of a product in the "natural environment of use"**. This is becoming an even more important area of research in the mobile age where smartphone users are interacting with mobile apps "on the fly" and in amongst many distractions. **The context of use for mobile apps often informs how those apps are used.**

UX researchers should look (wherever possible) to **combine remote and other methods to get the complete picture of mobile app use and usability.**

### Remote Research – A Non-Technical Solution

Not all remote research requires an investment in software or hardware; in fact there is one remote method which relies on good old fashioned pen and paper. That's the diary (or journal) study.

**A diary study** is a long-term study in which users make notes on when they use the product, how they used the product and any frustrations that they encounter. It is important for the researcher to choose a sample size large enough for results to be statistically significant and which allows for the inevitable "drop out" as some study participants fail to complete their diaries or are unable to continue with the study.



## Remote Research – Technical Solutions

Online surveys, Behavioural Analytics, Automatic Logging, ESM (Experience Sampling Methods) (see above)

## If Your User Can't Find You, You Won't Have Any Users – Mobile Applications and Discoverability

There are two main stumbling blocks to discoverability for new apps which include:

**The chart based model of the app stores.** It can be hard to get on the charts without being discovered.

Though if you can get a foothold on these charts, it then becomes incredibly easy for users to find you.

**The preference for curated lists.** Given that consumers find searching for apps a chore at the moment – they often turn to high-credibility sources that offer lists of Top 10 or even Top 100 (or more) types of app. It can be hard to get your app on to these lists without being discovered too.

There are **6 stages which will contribute to your app's discoverability**:

Pre-Launch Marketing Efforts

Early Collaboration

Initial Promotional Efforts

App Discovery Tools

User/Grassroots Activity

App Store Optimization (ASO)

## Other Methods of Improving Discoverability

These methods may not be applicable to all app releases but can certainly make your reach go further:

- Use print media – buy adverts or get features written in print media. It's not dead by a long chalk.
- Advertise in stores – this won't be easy for many app developers but if you can find retail partners willing to take advertising, it's a great way to get noticed.
- E-mail marketing – if you've got a list; get your app details out to that list.
- Your website – have a dedicated page for the app on your website, make sure it's easy to find from the homepage.

## Fit the task

### Goals of Mobile UX

Easy to use, easy to learn, satisfying, friction-free

#### Easy to use:

- User know where the controls are
- Discoverability is reduced
- No understanding is required (intuitive)
- Features are where users expect them to be
- Users feel in control

#### Easy to learn:

- User can find their way over time
- User remember shortcuts, rules, routines
- Learning curve is short
- Confusion is minimised

#### Satisfying:

- Produce loyalty behaviour (user are using less and less apps (4 on average) and are also uninstalling some)

- User enjoy using it
- User look forward to it
- Stress response is removed
- Users feel more bonded to it

### Friction-free:

Every cognitive interruption to "flow" to "rhythm" and establishing a bond with the user (keep them feel good) adds friction. We call this "synch". Synching with your user is very important.

Cooper: Cognitive friction is the "resistance encountered by human intellect when it engages with a **complex system of rules** that change as the problem permutes."

In mobile, we need to maintain flow. **Distraction** is inherent to mobile use so we need to minimise friction to avoid adding to the distraction problem.

### 2 rules + 5 mobile UX best practices

L layout      T tasks      N navigation      V visual      C control

#### Rule 1: Context is key for mobile

in the dark, with one hand, with different finger sizes, with different visual ability, with different levels of technical familiarity

#### Rule 2: Fat-finger friendly, always

44 pixels = fat finger friendly target

#### 1. Use clean layout

To support visible and intuitive controls: visible controls, back button and navigation. Intuitive Controls, Icons and Navigation

#### 2. Make the main task transparent

#### 3. Make the navigation contrast 'pop'

#### 4. Avoid visual density (too much colours)

#### 5. Less play and more control

### Designing for the Mobile Environment – Some Simple Guidelines

There are many things to consider when designing for mobile and while many are standard UX considerations; there are going to be mobile specific design considerations too. Are you going to integrate your mobile offering with your current offering? Will you use responsive design or adaptive design if you do?

Great iPhone Apps, offers **three categories for mobile web access**:

- **Microtasking**: When the user interacts with their device for brief but frenzied periods of activity
- **Local**: When the user wants to know what's going on around them
- **Bored**: When the user has nothing better to do and is looking to be entertained or otherwise diverted

### Basic Design Considerations for the Mobile Web

#### Small Screens

You don't have as much screen real estate for mobile devices as you do for PCs and laptops. That means, normally, you'll be designing for multiple screen sizes. You need to make a decision early as to whether to use **responsive design** (where the device handles the changes in display) or **adaptive design** (where your servers handle the changes).

You want to focus on a “mobile first” approach which means designing for the smallest mobile platforms and increasing complexity from there.

### **A good process to follow would be:**

Group device types based on similar screen sizes and try to keep this to a manageable number of groups  
Define content rules and design adaption rules that enable you to display things well on each group of devices

Try to adhere as closely to web standards (W3) as possible when implementing flexible layouts

Don't forget that there are many different browser types available for the mobile web and the wider Internet too. You want to ensure that you support as many of these as possible – including those that are no longer current (such as BlackBerry and Nokia WebKit).

### **Keep Navigation Simple**

Keypads and touch screens don't make for precise navigation like mice do – so try to:

- Prioritize navigation based on the way users work with functionality – the most popular go at the top
- Minimize the levels of navigation involved
- Ensure labelling is clear and concise for navigation
- Offer short-key access to different features
- Remember to offer a 30x30 pixel space for touch screen tap points
- Ensure that links are visually distinct and make it clear when they have been activated too
- Make it easy to swap between the mobile and full site (if you choose to implement separate versions)

### **Keep Content to a Minimum**

Don't overwhelm your users – respect the small screen space. Keep content to a minimum.

Make sure that content is universally supported on all devices or avoid it. Think Flash and then don't use it, for example.

Make page descriptions short and to the point – for relevant bookmarks.

### **Reduce the Inputs Required from Users**

The less the user has to fiddle with their phone; the more they're going to enjoy using your mobile web offering. Consider:

Keeping URLs short.

- Offering alternative input mechanisms (video, voice, etc.)
- Minimizing inputs in forms (you can always ask for more data when the user logs on to the desktop)
- Allowing permanent sign in (most smartphones are password or fingerprint protected – the risks of staying logged in are less than on the desktop)
- Keep scrolling to a minimum and only allow scrolling in one direction

### **Remember Mobile Connections Are Not Stable**

Mobile connections can be a colossal pain point in areas with patchy service. Don't make things hard on your users. Try:

- Retaining data so that it's not lost in a connection break
- Minimizing page size for rapid loading
- Killing off ad-networks, etc. on mobile sites which consume huge amounts of bandwidth and data
- Keeping images to a minimum and reducing the size of those images
- Reducing the numbers of embedded images to a minimum (speeding up load times)

### **Continuous Integrated Experiences**

As users move between mobile and the desktop they're going to expect similar experiences. Remember to:

- Maintain continuity. If they log into your webstore on mobile they should be able to track orders and make purchases just like they would on the desktop.
- Maintain consistency. Offer the option to switch between mobile and desktop offerings at will.

- Maintain brand. The look and feel of each version should be similar.

### **Using Mobile Apps – The One Thumb, One Eyeball Test for Good Mobile Design**

Luke Wroblewski suggests the “one thumb, one eyeball” test as an efficient way of coming to grips with this problem. It may help make your mobile designs more user friendly and enhance the mobile user experience.

#### **The Usage of Smartphones - ThinSlices offers these insights into mobile usage:**

- People use their phones in 68% of cases at home rather than in work
- 72% of smartphone users don't let their phones out of their reach at any time
- More than half of all mobile phones are now smartphones
- Half of all smartphone users consider their phone to be their primary access point for the Internet
- There are 7 categories of usage on mobile – 3 categories account for 77% of all time spent on the smartphone – socializing, shopping and “me time”.
  - self-expression (interests and hobbies),
  - discovery (seeking information),
  - preparation (getting ready for other activities),
  - accomplishing (managing health, productivity or finances),
  - shopping,
  - socializing and
  - “me time” (relaxing or being entertained). 46% of all time spent on a smartphone.
- Nearly half of mobile users only use their smartphones for traditional phone activities – calling and texting. These users do not download apps or surf the web. Indicating the potential for growth in the app/mobile web space on smartphones.
- The most used apps in the world are social apps (Facebook, YouTube, Google+, WeChat, Twitter, Skype, Whatsapp and Instagram) but the most used app is Google Maps (suggesting that “on the go” access is very common on smartphones).
- Usage times vary by culture (Chinese use theirs most after lunch, Europeans in the afternoon and Americans in the evenings).
- The average user interacts with their phone 150 times a day!

### **The One Thumb, One Eyeball Test for Good Mobile Design**

Luke Wroblewski, Product Director at Google, notes that in a distracted environment, the best form of interaction with a smartphone is one, which delivers high speed interaction with very easy to use functionality. He calls the typical mobile usage experience a "one thumb, one eyeball" experience, since the highly distracted environment causes most mobile users to engage in one-handed use with short spans of partial attention.

The one thumb, one eyeball test is thus about finding out if your mobile design allows users to easily use the app with one hand and partially distracted attention. In other words: **Can users perform a certain number of tasks with just one hand in under 60 seconds?**

If an interaction is measured in minutes or seconds, anything that complicates it is likely to hinder the user experience. **Users engaged with smartphones in that 150 uses a day are often not going to have the time to play around for 5-10 minutes working out how to interact with an app or mobile website.** They expect that you will cater for their “need for speed” in the design, and if you don't, they'll go elsewhere to someone who will.

### **My Head Hurts! Cognitive Friction in the Age of Mobile**

There's nothing more annoying than when things don't function the way we expect them to. When we develop user interfaces (UI), we try to relate it to the “real world” – think the floppy disk for “save” in most applications. Why? Because we take our cues in the digital world from ones we've already developed in our

ordinary lives. **Cognitive friction occurs when the cues don't match our expectations.** Learning to avoid cognitive friction in UI design can help make the user experience (UX) of products greater.

### **What is Cognitive Friction?**

**Cognitive friction occurs when a user is confronted with an interface or affordance that appears to be intuitive but delivers unexpected results.**

The more times that cognitive friction occurs when a user interacts with a product – the more jarring the experience of using that product is likely to be. If it occurs often enough, there is a reasonable likelihood that the user will seek an alternative means to carry out their task.

### **How Do You Avoid Cognitive Friction?**

Avoiding cognitive friction is the job of the user experience design team in conjunction with the UI and interaction design teams.

#### **Before Development**

There are ways to approach the design that can help identify where cognitive friction is likely to take place and to help avoid that:

**Conduct user interviews** – learn how your users expect to undertake the task with the product. Ask questions specifically relating to the interface during prototyping and sketching.

**Create task flows** – if you can create the simplest possible task flow with the greatest degree of automation (e.g. the computer not the user does the work) you are likely to minimize the opportunity for cognitive friction to occur.

**Create easy to use information architecture (IA)** – the simpler the IA, the less often a user is going to get lost or confused when interacting with the product. Extend this to creating navigational models and get users to test them and feedback on them.

#### **During Development Iteration**

You can also test for cognitive friction during the development lifecycle at each iteration of the product:

**Conduct expert evaluations** – cognitive walkthroughs and heuristic evaluation can be valuable tools when deployed by usability analysts. They may even uncover cognitive friction that isn't immediately obvious to users.

**Run usability tests on wireframes and prototypes** – nothing beats direct interaction with the product to test interaction design and examine how users will react to that design. The more user input you get, the more cognitive friction is likely to be identified and thus you can focus on eliminating it in the next iteration.

### **Understand Your Users**

persona: empathy capsule, insights and design direction. Defined by the roles, goals and tasks

We look at the tasks (Behavioral profiles, what they do and goals, tasks, sub-tasks) + Roles (hats they wear, shared needs and prioritised content)

#### **There are two sides to UX:**

**Usability:** Easy to use, learn and understand, eg intuitive, easy to navigate and few errors (Usability testing)

**Desirability:** has right tasks, meets needs, supports goals eg clear call to action, desired content, helps decision making (Field study). **This one is more important.** You need to have what they want.

## WHAT CAN YOU DO WITH PERSONA DATA?

Validate Requirements  
Identify Usability Requirements  
Use for Stories, Scenarios, Flows walk-through  
Use for Usability Test task validation  
Steer conversations away from features!  
Steer org toward customer-centric culture  
Get buy-in  
Guide Design  
Justify thinking  
Unify debates  
Herd cats  
Improve user advocacy & empathy



### Users on mobile:

Distractions  
Deep emotional context  
Deep social context

marketing persona: what the user tells you, not what they actually do (different from design persona)

### User Personas for Mobile Design and Development

The user persona is one of the most valuable tools in UX work. It allows the entire design and development team to keep a simple user story or user stories in mind as they work.

There are two types of persona that are particularly useful to user experience designers and they are the proto-persona and the user persona.

**The Proto-Persona** – is a persona created when there's no resource available to carry out research on users. They're based on whatever research can be gleaned from other sources and the best-estimate of who the team is working for. The proto-persona isn't as useful as the user person but it is better than nothing at all.

**The User-Persona** – this is the most common persona for a UX designer to be working with; it's a simple story of the user's goals, behaviours and pain points. **The user-persona is created through user research.** Both types of persona tell a story of what people do, why they do it, and what they need/want from a product. They are there to help communicate a guideline of what the eventual user experience should be.

### A user persona should:

- Elucidate observations from your research
- Focus on the now and not what might happen at a future point
- Be realistic and not seek to idealize behaviours or desires
- Help set out the groundwork of your user experience design task

It should also help you understand things about the user(s) themselves:

- The context(s) in which the product will be used
- The behaviours of the users at the moment
- The attitudes of the users in general
- The wants/needs from the product that you're designing
- The difficulties that the user wants to overcome in their current situation



- Any objectives that the user may have

### How Do You Create a User Persona?

There are a great number of ways to create user personas but a simple process would be to:

- Get a group of potential users and either interview them or observe them
- Look for patterns in their behaviours, responses, actions, etc. and then group the people that display such patterns together
- Then based on those groups, create archetypal models of the people (e.g. a stereotype)
- Build the user persona based on the stereotype but drawing out details of context, behaviour, need, etc.
- Disseminate the user personas among the team (and if necessary – seek feedback and amend them)

### What are the Benefits of Using User Personas in UX Design?

User personas have the main benefit of keeping the team focused on an outcome with a focus on how the user would perceive that outcome. They also help with preventing scope creep and self-referential design (e.g. designing the product as though the team or the team leader will be the user).

### How are Personas Used in UX Design?

The persona is there to ensure that every effort made by the team considers the user's voice. Failure to do this creates risks of failure in the design process itself.

When you create scenarios for product use, for example, the persona should be used to give that scenario context. It will then tell a story of how the product will be used rather than how we think it might be used.

There are 4 key uses of the persona in product design:

- Personas should be used to validate (or not) the decisions made by the design team. E.g. "is that what the user really needs?"
- Personas are used to develop priorities when ideas conflict for resources or time
- Personas should be available during ideation sessions to act as both a source of inspiration and to keep things grounded and focused on the user
- Personas should be referenced when critiquing ideas or iterations of the product

### How Does Mobile Differ from Traditional Use of User Personas?

In reality, the process for creating a mobile user persona is the same as the process for creating any other user persona. However, the context in which the user will be accessing the app or website plays a much more critical role in mobile UX.

Your research needs, whenever possible, to be observation based and conducted "in the wild" to ascertain how users really behave. Interviews may not be the best way to determine what people actually do (in fact, interviews often reveal large discrepancies between what people say they do and what they actually do).

If your users are accessing your product "on the fly" as they move around; their needs from the product are going to be very different from a product, which is only accessed on the desktop at their desk in their office.

### General Mobile App Usability Best Practices

There are no absolute rules in usability design and your users should shape your design decisions as much as best practices do.

- Content and all text should be easy to read. This should be true in any setting, including where bright sunlight is affecting the legibility of the screen. Fonts should be set at a minimum of 11 points and be consistently used throughout. Contrast between text and background is key to legibility in bright lights.

- Content should be available even when the internet is not. While it is not possible to maintain all content on the device, core content should be available even when an internet connection (both 3/4G and Wi-Fi) is not.
- Control spacing and size matters. These should be easy to tap and control by the user.
- Android apps – 48 dp target size
- iOS apps – 44 x 44 px target size
- Multi-step tasks that are not contained within the normal interface should be presented in modal views.
- The most important CTA (Call to Action) buttons in your app should be “sticky” or permanently in view.
- On screen content and transitions should be delivered quickly and painlessly to avoid frustrating users with long wait times.

### **Form Guidelines for Usability in Mobile Apps**

- Form labels should either exist above the fields or be created as floating labels to make certain users know why they are completing a form.
- Forms should attempt to communicate what is necessary in a field to avoid user error.
- Forms should be validated as each field is completed and not at the end of the process.
- Forms should begin with the cursor already placed in the first field and with the correct input keyboard displayed.
- Forms should ensure that CTA below the fold is automatically scrolled up when a user completes the form.
- Forms where lists are presented as options should display these horizontally above the keyboard rather than vertically. This ensures that the screen real estate is used effectively when a keyboard is present on screen.
- Forms which require numeric entry should display a numeric dial pad style keyboard when required.
- Forms can benefit from stepper controls when users need to edit quantities. This is also true for shopping baskets.
- Forms and search facilities should offer slider controls to set maximums and minimums.
- Forms that require date/time input should offer picker controls to make this process easy on the user.
- Forms that require calendar entries should consider displaying a visual calendar – this is particularly important in travel/hotel booking style apps.
- If an app crashes whilst a user is engaged with a form it should auto-recover any data entered and return the user directly to the point in the process when the crash occurred.

### **The Four Big Don'ts of Mobile Usability**

In addition to what to do, Google also turns their attention on what not to do. They've identified four areas in mobile app development that cause either user confusion or user distress. This reduces the overall appeal of the app and the user experience dramatically.

**Don't copy UI elements between platforms.** It may seem like a good idea to ensure that your iOS and Android and Windows apps all look exactly the same but each platform has a defined look and feel. These lead to users developing clear pictures of the conventions associated with that platform. If you change the UI to resemble a different platform; it doesn't benefit the user it leaves them unclear as to how to proceed.

**Don't use underlined URL links.** The underlined URL is part of a different model of online interaction – it belongs in websites that are accessed via browsers. Mobile apps don't work that way, they rely on buttons to signify a change in screen as opposed to a link which takes you to a different page. Google adds a corollary to this rule too – don't hard code links because they need to be manually changed and broken links spoil the user experience.

**Don't take users to browsers.** You want to keep an app user inside your app and not send them elsewhere. Sending them to a browser risks them not returning to the app and provides a “clunky” user

experience. If you must call their attention to some content online – use a browser within your app to deliver that content; don't send the user elsewhere.

**Don't demand an instant rating in the app store following download.** If you want good ratings for your apps, you have to give the user some time to get used to it and to appreciate the experience. You should be asking regular repeat users for ratings not those that have just installed the app. They'll be certain to give you better reviews and reviews with a higher level of content.

### Persona: Toward Best Practice

Conduct "day in the life" visits - Field studies = walk in their shoes. Empathy is a UX requirement.

#### EXAMPLE- MENTAL MODEL FOR GOING OUT TO EAT

- Enter the restaurant
  - ....
- Order food
  - Get menu
  - Look at menu
  - Choose food
  - Server arrives
  - Give order to server
  - Wait, talk
  - Cook prepares food
- Eat food
  - ....
- Leaving the restaurant
  - .....



#### CONSIDER THE TASK ENVIRONMENT



##### **ACTIVITY: Empathize with users via tasks.**

Describe in verbs the mindset or problems users are trying to solve (3 minutes)

think in terms of **Verbs**

Usually easier to talk to the user on their work post. You get the empathy and see what they are doing.

#### 5 AREAS OF TASK ANALYSIS

The Customers Language

The Environment (physical and social)

The Functional Needs and Priorities

The Cognitive Demands and Needs

The Task Space (problem-solving, decision-making, sense-making)

A **task flow** maps the users's activity. Start with the mains task first and follow the flow.  
Designing for tasks: Start - tasks...sbu-tasks....goals...Finish

**What is a task:** Something a user wants to do + the need of a user expressed as design element + how a user interacts with a system + how we measure all success with an interface.

Task oriented : a user-centred approach. That's how user naturally thinks and how they go about executing a task. They don't care about the features or functionalities, they care about how they can get familiar, how much our design affect their mental model.

work through all the data from interviews – what have you learned? What trends emerge from this? What can you use in your product?

## Fitting the UI style guide

### **Smartphone vs Tablet:**

It has become an industry standard to put “mobile first” when designing for the mobile web and while this is, in general, a good thing – it has also left to a neglect of the tablet platform in mobile design.

The smartphone tends to dominate discussions of the mobile internet and there's no major surprises about that. Figures released this year show that smartphones not only dominate the market but are continuing to grow (in terms of market penetration) at about 13% a quarter. The humble tablet, on the other hand, seems to have reached peak market penetration and is expected to decline in sales from this point forwards according to figures from IDC, the data analysts.

Smartphones tend to be the preferred platform for users aged between 18 and 34 but for those older or younger than that market sector; the tablet is the dominant platform according to Flurry analytics. This may be because the tablet makes for an easier transition from the laptop/desktop market for older users and is more convenient in terms of screen real estate for younger users (who are also less likely to be “always on the go” or have less access to a smartphone than the 18-34 year old market).

#### **smartphone:**

- More mobile than tablets and more likely to be used in wide variety of locations and scenarios
- More likely to suffer from distracted use (because of their mobility)
- Generally of lower (or at least smaller) resolution than tablets
- Rarely the primary access point for users to the internet (though in developing nations they may be)
- Likely to be used for multi-tasking (users flit from app to app rather than spending significant times doing any particular task)
- More personal and thus more likely to be used for social and emotional contexts
- Less likely to have controlled lighting available when in use

#### **tablet:**

- Less mobile, they're too big to be user friendly in all spaces (though we've noticed this doesn't stop people from trying nonetheless)
- More likely to be used in a fixed position (at home, in the office, in a café)
- More likely to be used by more than one user and are thus less personal and emotional platforms
- More likely to provide higher resolution (or at least bigger) experiences than a smartphone

Both tablets and smartphones require “fat finger friendly” design in that 44 pixel targets are a minimum for buttons, icons, etc.

**Ergonomics matters:** minimise detailed manipulation-offer tappable visual choices

recommended touch target size is 9mn/34px - minimum 7mn/26px

minimum spacing between elements is 2mn/8px

visual size is 60-100% of touch target size.

Top left is the least used area, and bottom right.

User prefer the vertical mode (so design mode by default)

### Content first, navigation second for mobile

Screen layout default:

back left side: action right side, main controls lower panel (most of the time)

top panel controls are visually lost here, use tabs instead to clearly articulate the controls

### Think context sensing

Defaults should auto-defect usage contexts.

Advance settings -concealing (to not overwhelm the user) if using hidden doors, secrets panels - show a UI hint or fade in/out a hint. This will reduce discoverability time.

### The Mobile User Experience

We already know that using the mobile Internet is, in some ways, more challenging that using the internet from the desktop. Some of the issues that users encounter include:

- Smaller bandwidths for data transfer. 4G is definitely light-years ahead of 2G data transfer but it's still much slower than high-end cable connections in homes and offices.
- There's no keyboard. Sure, there's a keyboard on a touch screen but it's not as easy to use or as accurate as a keyboard plugged into your PC.
- There's no mouse either. Users have become very dependent on mice over the years to interact with their IT and this can make it difficult to work with the mobile internet.
- The screen real estate is seriously lacking. Coming from an awesome desktop wide-screen to a tiny little phone screen can be hard to get used to. This is particularly true when designers and developers try to cram way too much on to that small screen by reducing font sizes, etc.
- Developers often aren't familiar with or don't follow mobile usability guidelines and mobile web standards. This can cause websites and apps to deliver sub optimal experience on mobile.
- User interfaces haven't reached a level of standardization when compared to desktops too. That can leave users confused as every designer tries something new.

Comprehending information on a mobile (vs desktop) is twice as hard because of one single factor: the screen size of the device. Two explanations for this:

- When you can't see as much of the content – you must rely on your memory to keep the context alive. A lack of visible context reduces the level of understanding that the user has of the content.
- When you can't see as much of the content – you must navigate around the screen more. This distracts the user from the content and it requires additional time to carry out. In turn, this makes it more likely that the content will not be retained in short-term memory.

### In short, small screens make us stupider when it comes to reading comprehension

#### What Can Be Done About This?

The only solution to this is for designers to simplify their content. It needs to be in plain English and kept to the bare minimum required to convey the important information. The less scrolling that's required and the more context that's available on screen at one time – will result in better retention.

For companies that are catering to audiences on the desktop, laptops and tablet platforms – this will mean redesigning the mobile experience to cater for the mobile user and their reduced screen real estate.

Use high contrast for quick glance.

Providing in-context controls is helpful and useful, they support de drill down and provide the control when you need it and don't clutter the screen. This improves making the main task apparent.

make the main task apparent

avoid extra navigation

**Skeuomorphism** is where an object in software mimics its real world counterpart. The “trash can” is, perhaps, the most recognizable skeuomorphic object. Though the good old “save” icon was once skeuomorphic but following the demise of the floppy disc – it no longer bears resemblance to the world of today.

Skeuomorphism helped a generation through the learning curve of coming to grips with a digital era. But, it also began to hold us back. We became familiar with the concepts and they entered the language and our day-to-day lives but skeuomorphic design led to huge amounts of clutter on the desktop. They brought too many useless details to our computers which we no longer needed.

**Flat design** mandated that graphic user interfaces (GUIs) should be freed from clutter. There was no need for bevelled edges, gradients, reflections, and skeuomorphism. The digital interface should be exploited for its own strengths.

## **Don't Make Me Think – Key Learning Points for UX Design for the Web**

### **Self Explanatory is the Way Forward**

Whenever a user opens an app or a web page they should be able to tell you at a glance what it's for. It should be as obvious as can be what the purpose of things is.

### **Don't Make Me (or Anyone) Think**

If you make people think, you make them unhappy. Users don't want to treat your site or app like some sort of cryptic crossword – they want to know what they should do immediately and then do it. The more you make people think, the more likely they are to go elsewhere to get the job done.

### **Let Me Go Back**

As long as the back button works, mistakes don't matter too much from a user's perspective. The back button is the most used feature on the internet today. Make sure people can get back to where they started if they get lost.

### **People are Habitual**

If something works well – people will tend to stick to using that. Even if there's a better way to do something out there – it's unlikely that they'll go looking for it. That doesn't mean that they won't eventually have it called to their attention but if you make things usable; you make them sticky.

### **Get to the Point**

Again with the time saving – people don't look for pleasantries on the internet; they want to get to the point. Make sure your sites and apps do just that.

### **Some People Love Search**

You need to have a search facility within your web app or site. There's a certain group of users who have become search-dependent. If you don't give them the option; they'll go elsewhere.

People remember the experience of interacting with web sites and applications. They learn where to find what they want and how to do what they do. When you mess with the site map – you mess with their ability to recall these steps and make things confusing for returning users.

### **Show Them the Way to Go Home**

The home button is the ultimate emergency exit for users. If they get really, really lost and even the back button isn't helping – they need a single click to get them home. Make sure it's obvious and immediate.

## **Areas of UI that Help with Discoverability and Learnability**

When making applications (and especially mobile applications) discoverable and learnable you'll want to focus on UI to facilitate that and in particular:

**Action Bars** – these provide the opportunity for the user to “bank” their most commonly taken actions. This is in contrast to the navigation bar more traditionally used in web applications. They remove the clutter from the screen and focus on what a user is most likely to want to do in any given circumstance.

**Social logins** – it's a good idea to first consider whether users will want to link their social profiles to your app. However, if they are going to be OK with that – logging in with a social network can save time, save account creation and still provide you with useful data for your app design team.



**Big Buttons** – the bigger the button, the clearer it should be that the user needs to take an action. Think Badoo and its' tick/cross approach to dating, for example. Of course, big buttons are best in apps that are going to offer a limited number of options to a user at any point in time. Otherwise they're likely to become very annoying and require a ton of scrolling to handle properly.

**Notifications** – when something changes for the user, let them know. Notifications don't have to be dramatic and highly intrusive either. The simple digit count on message apps so that users are aware there are new messages available to them, for example, is a great way to update someone without intruding into their day.

**Discoverable controls** – mobiles offer a whole new set of interactions with a device. Touch-sensitive screens, accelerometers, GPS readings, swipe controls, gesture controls, image controls, etc. it can be incredibly useful to make these controls make themselves known as users experiment with their devices.

The good news is, of course, that many of these discoverable approaches are already available as UI patterns. You don't always have to reinvent the wheel in order to assist your users via UI design.

### Testing for Learnability and Discoverability

It can be hard to test for both of these concepts prior to implementing a design. They are a consequence of usage and as such even prototyping may not lead you to the answers that you seek when it comes to these concepts.

It may best to agree that you need to iterate a product from an alpha or beta version to a launch version to test learnability and discoverability. **They should also be revisited at a later date once your user base has become more familiar with the product and you can gain a better understanding of how these concepts played out in "real life".**

### Good UI patterns:

pull to refresh, plus button, hamburger icon. You can innovate, but you need to research to make sure it works with users.

Floating context menus, very usable, you don't need to look for it, they just appear. Like social media to share.

**What's the correct spacing?** is elements are too close together this is a problem.

scrubbing is pretty not finger friendly (like when you're watching a video or listen to a track)

**Provide Transition hints:** curl transitions can work great but animation or fade work well too. Make sure it's the right design for your task.

**Should you provide instructions:** most of the time, no it should be easy and intuitive. Instructions should be short.

**Avoid crowding / watch proximity / use 2 to 4 colours,** this is enough most of the time. Minimalist colours is best.

Start up screen, login: present most common tasks up-front (pre-login)

**Mobile First:** what is the most common tasks, what is the context of use, and put this in a small area, so focus interactions.

Provide a unique experience.

**Task first:** what is the MVP

## TASK FIRST



- ✓ It should be intuitively obvious to the most casual user
- ✓ Design for the most common 80% of personas, first
- ✓ Generate several designs and test them all
- ✓ When designers cannot agree, test each design



Agile UX means regular contact with users (typically a weekly/monthly usability test). UX reviews can fill sprint needs.

### Content first:

## CONTENT FIRST

### Adapting to a Layout-Less (Responsive Web Design environment)

Start with:

#### A. COPY (Typography; Font size)

- H1
- H2
- Body
- Bullets
- Motivational text/ slogans/ taglines
- Etc

Next add:

#### B. UI ELEMENTS (widgets, ui patterns)

- Text box
- Buttons
- Checkboxes/radio buttons. Dropdowns
- Tables
- Etc



Make sure it's easy to digest, adapt to various screen size regarding typo, and font size, then think about how you can add your UI elements (widgets, UI patterns) Then think of your interactions elements a (interactions, flow, task efficiency (trim, reduce, remove features) and finally the media (images / media - visual design cues). Think about how responsive - adaptive design can enhance the experience.

### Platform considerations

lifecycle:

- 1- real world need, personas
- 2- business requirement
- 3- create wireframe (task oriented design)
- 4- usability test (validate with users)
- 5- visual design
- 6- code
- 7- to app store/google app or site marketing

### Try to be consistent:

with touch screen experience, pick one:

- highly visual
- tap to select
- flick to scroll
- multi-tasking

hard key experience:

- scroll through menus

- click key to select
- click key to scroll
- multi-tasking

Hybrid experience:

- select menu for actions
- touch/hard key to select
- back button
- multi-tasking

**Be mindful of the OS style guide** (android is different from iPhone) you can deviate sometime but most of the time no.

### 5 ways to enhance your experience:

- show all content (don't hide things from them)
- support mobile UX tasks
- cater to mobile contexts (adaptive design) = enable greater precision in task success, greater conversions, the social nature of mobile
- support mobile usability
- do accessibility! (vision, hearing, motor and cognitive)

## WHY ORGS ARE GOING 'RESPONSIVE'

### GOOD UX REASONS:

Fixed resolutions offer sub-par experience  
Pinch and zoom fatigue on mobile (m. sites)  
Scale to a Mobile App-like experience rapidly  
Boost conversion rates  
(Hint Video is part of that story- see right)

## IF YOU DID GO RESPONSIVE

### ONE SET OF CODE TO:

- MANAGE (one change to all)
- DESIGN (adapts per device)
- DEVELOP (one version of code)

### BETTER MOBILE UX STRATEGY:

- AVOID 'NEEDING' AN APP
- AVOID CRAPPY M. site
- SCALE TO FUTURE DEVICES/ VIDEO

### PROVIDE A BETTER EXPERIENCE:

- ADDRESS ACCESSIBILITY
- ADDRESS INTER-OPERABILITY
- OFFER GREATER CONTEXT SENSITIVITY

### If you don't go responsive:

more code to manage, dev and test  
less access to CSS 3 enhancements  
less strategic UX optimisation  
resolution madness intensifies

### Responsive is not that easy:

CSS 2 to deal with (use multi stylesheets)  
old version of IE to support, or not  
still lots of testing (and cost)  
over downloading to deal with (optimisation of download speed - for example: apple loads regular images then replaces high resolution images with javascript)

## Native vs Hybrid vs Responsive: What app flavour is best for you?

At some stage, during the design of your mobile app, you're going to have to make a decision about how you will develop the app. Your choices include native apps, hybrid apps and responsive web sites and the decision you make will affect the final product. Learning to make the right decision requires understanding the way each type of development works and how it affects the final product.

There are three main options for creating a mobile app. You can create a native app, a hybrid app or a mobile responsive website which offers similar functionality to an app. Each option has its advantages and disadvantages and it's important to understand these in detail before committing to a particular development path. The path you choose is likely to be informed by the resources available to you and what you're trying to achieve.

### What is a Native App?

**A native app is designed to run on a specific mobile operating system.** It won't run on other mobile operating systems. So, for example, if you were going to develop a native app for iOS you'd do the development in Swift.

The biggest advantages to native apps are that they can normally access all the functionality of the chosen device easily and that they are more likely, if developed properly, to run without error on the device. However, this comes with a trade-off. A native app cannot be run on a device which doesn't utilize the same operating system. That means if you want your app to run on Windows, iOS 7, and Android, you're going to have to develop the app 3 times, once for each operating system. This can make the development process both slower and more expensive.

Many companies will develop their app for a single operating system when they choose the native route. **If the app is successful in that environment, they will then go back and recreate it for other operating systems.** It's worth noting that **at this moment in time, iOS 7 apps are, generally, more profitable than apps released for other environments.**

### What is a Hybrid App?

**A hybrid app is designed to work on multiple platforms.** It's written using a single standard code language (such as C# or a combination of HTML5 and Javascript) and then compiled to be executed on each platform. Device specific interactions will normally be managed by the use of plugins for that operating system.

The biggest advantage of hybrid apps is that they enable support for multiple operating systems at a more cost effective price point than developing multiple native apps. Users, as long as the development is done well, will not normally be able to distinguish whether an app is native or hybrid. In fact users, in general, don't care about how your app is developed – they just care if it works on their device and does what they expect it to do.

**There are disadvantages to hybrid app development. It can present serious challenges if the app requires complex interaction from the device – there's a limit to what plugins can achieve on this front.** The costs of supporting a hybrid application may be higher than the costs of supporting a native application too.

### What is a Responsive Website?

**A responsive website can deliver similar functionality to an app. In fact, with a little creativity you can keep the differences to a minimum and ensure that the home page jumps straight into a "full screen" display.**

Responsive websites will be developed using HTML 5 and Javascript. Typically developers will take a "mobile first" approach to development with the mobile version offering a simpler and pared down experience to the desktop version of the same site.

**The biggest drawbacks to using responsive websites are firstly that the app cannot be distributed through an app store; this can be a deal breaker if you're looking to monetize downloads of your app. Secondly, there's the issue that the user will need constant connectivity to make use of the website.**

This may not be a problem in highly developed markets where mobile broadband is near ubiquitous but can be a serious issue in developing markets.

It's worth noting that while, at this moment in time, apps appear to be the driving force behind the mobile web – there is room for serious disruption. With the average user already using up to 30 apps a month and with more than 250,000 apps being released a year, there may come a point of overload from a user's perspective. At that point, **it is possible that if there are enough well designed mobile websites, that users will move away from apps and back to browsing and individual sites to provide their online experience. Then, apps might just become “launchers” of mobile websites.**

### **Mobile UX design checklist**

- consistency of controls (OS and app consistent actions)
- proximity (closeness of items if similar)
- visual hierarchy (top important to bottom less)
- colour contrast and emphasis (low contrast used sometime to save the battery)
- intuitive icons
- **make the main task apparent**

prototype and test early, paper or clickable prototype.

Mac: [keynotekungfu.com](http://keynotekungfu.com) or for others: [keynotopia.com](http://keynotopia.com)

### **Social UX considerations:**

Social factors are very important in app development.

Identify what is happening in the real world and bring it to the app.

### **Social UX basics:**

Activity of others ("7 users are looking at the room just now")

Status of others (someone bought that ..)

Content of others (recommended, viewed etc..)

Following liking tagging

Location base content

This has nothing to do with social media!

What do you do with this and why?

**Social navigation:** help users discover or benefit from peers. improve usability; improve desirability,

## **UI mobile components and patterns**

### **The Anatomy of a Smartphone – Things for Designers to Consider for Mobile Development**

If you're going to design for mobile, then it's a good idea to have a grasp of what makes up the modern smartphone in terms of hardware and software. You can then start to think about the possibilities these offer when creating user experiences. The better you know the device, the more you can exploit its capabilities within your products.

### **The Software**

There are two operating systems that dominate the mobile landscape: Android and iOS. However, there are other operating systems to consider as well.

#### **Android**

Founded by Andy Rubin and supported by Google and drawing the backing of many of the major hardware and software developers in the smartphone sphere; Android has gone from strength-to-strength since it was conceived on October 2003. It is the world's leading smartphone operating system and has been dominant since late 2010.

## iOS

Apple's operating system was, for a considerable period of time, the leading smartphone operating system. It is unsurprising that it should be in 2nd place in terms of market penetration today. Apple does not license its operating system to other hardware providers nor does Apple operate in the budget sector of the smartphone market. While it may be second in terms of market penetration – there's no doubt that the average iOS user spends more money on apps than the average Android user.

## Windows

Windows may be the dominant operating system on the desktop but it's a bit of an "also ran" on the smartphone. There are fewer apps available for this operating system and it tends to be a tertiary consideration (if it is considered at all) for most developers.

## Blackberry

Blackberry was once the dominant force in corporate handset provision. Poor management decisions led to Blackberry suffering a huge dip in market share after smartphones launched. The operating system continues to exist and support millions of subscribers – however the company launched their first Android operating system handset in November 2015 and the future looks dim for Blackberry. It is unlikely that developers will focus on Blackberry without an overwhelming business driver for doing so.

## Firefox OS

Firefox's open source OS was launched back in February 2012. While there are a few commercial handsets running Firefox OS at this moment it's not a primary consideration for most developers.

## Sailfish OS

Sailfish is based on Linux and Mer and has a unique programmable UI in the form of Jolla (a multi-tasking interface). It is a project from MeeGo (consisting mainly of ex-Nokia employees) and is yet to achieve any significant commercial uptake.

## Hardware

A smartphone is, at heart, a miniature computer and thus it shares many similar components with computers – such as processors and memory. These components are constantly upgraded in new generations of phones and it's important to consider how to balance the demands you place on this hardware. **If you take advantage of latest generation technology; you rule out many existing smartphone owners from using your product.** Go back too far in generations and you risk releasing a product which isn't as powerful as your competitor's.

However, in this section we want to look at smartphone specific hardware options that are commonly found in smartphones that are not commonly found in desktop/laptop computing. These are the hardware items that designers will want to consider taking advantage of when developing smartphone user experiences.

## Touchscreens

The touchscreen interface has developed rapidly over the last decade. There are two main variables to consider when designing for touchscreens:

**The sensitivity to touch** – resistive screens are made of two layers of conductive material with a gap between them to serve as resistance. They are not as responsive as a capacitive touchscreen, which uses a layer of glass with a conductor over it that senses a break in the electric field when a finger comes into touch with it. There is also the haptic screen which provides tactile feedback to the user when they touch the glass. The Super AMOLED (Active-Matrix Organic Light Emitting Diode) screen includes touch sensors on the display and is a further improvement in both performance and thickness of screen.

**The quality of resolution** – TFT (Thin Film Transistor) screens offer reasonable resolution but are power intensive, IPS-LCD (In Place Switching Liquid Crystal Displays) offer improvements in both resolution and power consumption (the Retina screen is an IPS-LCD). Then there are OLED (Organic Light Emitting Diodes) and AMOLED (Active-Matrix Organic Light Emitting Diode) screens to consider too both of which offer advantages over LCD.

**GPS (Global Positioning System)** GPS chipsets are found in most mid-range and high-end smartphones. They enable, with varying degrees of accuracy, the handset to pinpoint where on earth the smartphone is located.



Designers will need to consider location based services for certain products. GPS allows them to automate much of the grunt work for this.

**Accelerometer** Not all phones contain an accelerometer but many do. An accelerometer measures the speed of movement of the device and the direction in which the device is moving. This enables the pedometer function common in many health apps as well as other more ingenious uses of motion data.

**Camera** Nearly all smartphones come with on-board cameras today. These vary in terms of their quality in both resolution and availability (front and/or back mounted cameras) and can be used as input devices.

**Heartrate Monitors** Many phones come with a heart rate monitor now and those looking to create mobile health apps will want to tap into this functionality.

### **Fingerprint Sensors**

High-end smartphones often incorporate finger print scanners and finger print recognition technology. While this can be a useful security measure – over-reliance on finger prints can become a barrier to user acceptance.

### **The Rumble in the Board Room – Mobile Splash Screens What Clients Want and What They Should Get**

First impressions count. So how do you make a good impression on the first time user of a mobile application? Through a mobile splash screen. Unfortunately, there are often conflicts between a client's wants for the splash screen and best practices. If you can learn to manage these conflicts – you can create better mobile user experiences for end users.

It can be tough being a user experience designer at times. Your role is to represent the user and deliver the best possible end product but there are times when the stakeholders (and particularly clients) want you to do something entirely different. **One key area of conflict in the current mobile environment is the mobile splash screen.**

**The splash screen is the very first screen that a user sees when they open up your app on their smartphone or tablet.** It's your chance to make a good impression on your user and set the tone for what's coming with your application.

The splash screen dates back to the days when Flash first became common on the web. **It takes time for actual content to load and execute – so to reassure the user that something was going on –the splash screen was used to catch their attention and show them that they weren't just waiting for nothing.** You'd think something so simple would be an easy task to agree the way forward on wouldn't you? Unfortunately, it's not quite that easy in many instances.

**What the App Stores Recommend:** Both Google and Apple have developed guidelines for mobile apps and both of them say that there's **a specific use for splash screens in mobile apps. That is to give the user the impression that their app has loaded faster than it actually has.**

**The idea is to design a launch image that appears identical to the first screen of the app when it is loaded.** Text on the image is static and thus can't be localized (except for Windows 8.1 or above). It's also important to try and be consistent with the UI so that nothing changes between the splash screen and the first real screen of the app. In addition, **you ought to indicate that this is a loading screen to ensure the user doesn't become frustrated pushing buttons on the splash screen.**

### **What Clients Want**

Clients, on the other hand, often don't want this. They've become used to seeing splash screens in games and major desktop applications. **Those splash screens are all about branding and marketing. They reflect the heavier workload of apps and games, etc. that aren't optimized for mobile.**

The trouble is that a designer who applies the same technique to the mobile splash screen is, in essence, telling the user that they're going to be waiting a long time for the app to load. That's a bad idea as many app users are about as patient as website users i.e. if it takes too long to load, they're leaving and not coming back.

### **How Do You Handle This Conflict?**

The only way to handle this conflict between best practice and client demands is to draw your client's attention to the best practice guidelines. You need to make the case for the user experience and explain how using branding and marketing approaches to the splash screen could damage the success of the client's app.

UX designers have to be consultants for their clients (even when the client is an employer, not an external agency) and their role is to provide options and recommendations to the stakeholders in the project.

If the client chooses to pursue a marketing and branding approach after you've given your recommendation – they're free to do so. It's their money when it comes down to it. As long as you've provided the right information to the client and made the right recommendation – you've fulfilled your responsibilities.

### **User Input and the Mobile User Experience – We're All Thumbs Now or Maybe Not**

Mobile platforms are both limited in terms of their user input and also, in other ways, vastly superior to traditional laptops and desktops. Designing apps or web sites for mobile means taking into account the differences in the way that users can interact with a device and shaping the experience of the final product to take advantage of the strengths of the platform while avoiding, as best as possible, the weaknesses.

**On the desktop** user input is reasonably easy to coax from the user. There's a nice big keyboard and a mouse for the individual to take advantage of. Of course, a good user experience still tries to avoid demanding too much effort from the user in terms of input but there's a greater degree of freedom for designers than there is on mobile.

**Mobile input is substantially different.** Keyboards aren't as easy to type on when they occupy a tiny space on a touchscreen surface. There's no mouse at all. Then there's the addition of gesture controls and eyeball tracking that aren't commonly available to the desktop user. So what makes for a good user input experience on mobile?

### **Simple Guidelines for Mobile User Input**

#### **Keep it Simple Stupid (KISS)**

You don't want to overwhelm users with page after page of data collection on mobile. It's simply not practical to try and extract the same volumes of data from a user that you would on a desktop. So keep it simple stupid and try to reduce the input to essential data only. You want to be as thorough as possible when it comes to reducing the number of fields and try to capture data in the most efficient manner possible (for example – you could take a Post Code or ZIP Code instead of large amounts of address data).

#### **Offer Default Inputs When It's Practical**

If you know, for example, that 85% of your customers come from the United States and that 10% come from the UK (with the other 5% coming from everywhere else) – your default country option should be the United States and if you offer a drop down list to choose from, the next option at the top of the page should be the United Kingdom. There's no need to make users hunt for data that you can reasonably predict. This won't always be practical but if you can find a decent default choice for a field – make sure that you have it filled in.

### **Think About the Device Capabilities**

You can use motion capture, the camera, the internal accelerometer and gyroscope for positioning, the GPS and of course, voice on a mobile for input options. Mobile app designers seem to already have a pretty good handle on these ideas and take advantage of them wherever they can – mobile web designers, on the other hand, are flagging behind. It's important to take advantage of simple ways to collect inputs and really build on the capabilities of the devices people use to access the mobile web.

### **Give the Right Keyboard for the Right Task**

For touchscreen smartphones – if you want the user to enter data – try to give them the right keyboard to do it with. If you're capturing a telephone number; provide a numeric keyboard to get people started quickly. If you're capturing a name; provide an alphabetical keyboard. This isn't rocket science and it saves people from having to switch between the two and thus enables them to focus on providing the right input.

### **Don't Forget Not All Phones are Smartphones**

There are still people accessing the mobile web on more traditional handsets and it's a pain in the posterior to enter data from these kinds of phone. Each letter requires multiple key presses and it can be really tedious to enter more than a small amount of data in this way.

### **Hold Information Where Possible**

Assuming the application doesn't hold hugely sensitive data – don't make users enter passwords to access the information. Also make it easy to transfer data between devices, there's no need for someone who has a desktop account to recreate the wheel when opening a mobile account. Data should move seamlessly between the two whenever possible.

### **Offer Error Minimization**

It needs a little thought but if you can integrate auto-complete or spell check or predictive text into your application; you can make it easier on your users when it comes to providing the right information at the right times. However, you don't want to overdo this as it can become annoying when applied to fields where there are no standard inputs (for example name fields).

### **Kill CAPTCHA**

There is no bigger irritant online today than the CAPTCHA. You want to design your mobile applications to avoid CAPTCHA whenever possible. The small screens make it less than ideal as a security measure and there are better ways to secure an application (on any screen) than with CAPTCHA.

### **Tell Me About Me – User Profiles for Mobile Applications**

With the move away from websites to applications on mobile devices; there's a need for many apps to create, maintain and manage user profiles so that users can effectively interact with the app and so that the app's creators can collect and analyse data useful to their businesses. There is no "one size fits all" method of building user profile interfaces on mobile but there are some best practices that may be worth considering as you design your user profiles.

Mobile apps usually require user logins and for users to offer up a certain amount of information to get the full experience from the app. This requires designers to pay some attention to the design of the user profile interface – how can the profile be made to work in a way that is pleasing to the user?

### **Keep Things Clear and Keep Them Concise**

A user profile page, like any other mobile page, is constrained in terms of screen real estate on a smartphone. That means that you don't want to try and cram too much information on to a profile page or you can make the page too complex for the user to want to interact with.

You need to pay careful consideration to the information that you actually need in order to accomplish this. The basic rule of thumb is to **only collect data on mobile if it is essential to the business or to the use**

**of the application.** You simply can't expect users to fill in 10 page forms with joy when they only have a thumb dedicated to the task of form filling.

This principle will then ensure that your user profile is streamlined. If profiles are viewable by other users – **you also need to consider what data will be shared and what will remain private to the individual.** However, that doesn't mean your designs need to be bland. You can employ all the visual design tricks associated with the mobile operating system to make the profile aesthetically pleasing.

### **At a Glance User Statistics**

Not every user profile is going to take advantage of delivering user stats – a lot depends on the mobile application itself but if it does deliver stats. These should be easy to see at a glance, stats can motivate users to become more involved with an app but only if they're readily understood and there's a basis for comparison (either with previous stats generated by the user or with other users' stats).

You may also want to consider privacy concerns when designing stats for user profiles; in particular is there a way that the user can opt out of sharing stats with other users? Or can the user determine which stats are shared and which are not?

### **Navigation Should be Intuitive**

Profile pages should be easy to get to and easy to get around. Menu options should be simple to access and deliver exactly the right options when they are opened. Sliding menu systems are, perhaps, the easiest way to achieve this. If there is a "social" element to the mobile app then it can be a good idea to offer two menus – one relating to functionality that pertains solely to the user and one that pertains to the social aspects of the application.

As with all mobile navigation, you should present the most commonly used items in a menu first and ideally you want **to keep options to fewer than 7 choices.** If you need more choices, it's better to break the menu down into sections with sub-menus that allow further choices.

### **Aesthetics Matter**

A functional design is a wonderful thing but in order for profile pages to appeal to users in the long-term they also need to be desirable. That means focusing some of your design efforts on making the menu systems and the profile page aesthetically pleasing.

You want to experiment with animations, textures, etc. to deliver a visual treat once you've ascertained the "usability" of your profile pages.

## **Getting Lost and Found – Maps and the Mobile User Experience**

The ability to harness GPS data and map data on smartphone platforms offers designers a chance to enhance the user experience of their products. However, in order for maps to deliver better experiences for users – it's important to integrate these features with UX in mind. There are some sensible rules that can be used to increase your chances of success when using location data and maps in mobile apps.

There's no doubt that the smartphone has brought users and designers a wealth of choices when it comes to new facilities; one of the most popular of those is the GPS (Global Positioning System) functionality. GPS can be used to calculate the user's position and be combined with map data to provide ways for users to find locations and get directions to them.

However, it is important to consider how you will implement maps within your mobile websites and applications before automatically deciding on their conclusion.

Maps within mobile apps and websites fail to take into account certain usability considerations.

## **Usability Considerations when Implementing Maps in Mobile Designs**

### **Scrolling and Panning**

One of the nice things about maps is the ability to pan across them by dragging the map. This functionality works extremely well in a desktop environment when the user clicks on the map and then drags the map

with a mouse. It is more difficult to pull off on a smartphone screen where the user interface is a touch screen.

Users will often scroll the page rather than pan or vice-versa because the boundaries of the map and the boundaries of the screen conflict. This may be overcome with the use of borders around the map so that the user has a distinct area for panning within.

### **Pinpoints Conflict with Target Sizes**

It can be easy for designers to forget that touchscreens are not the most sensitive of devices and that ideally, for a user to click on an area, it needs to be about **36 x 36 px**.

Pinpoints on maps are all too often rendered smaller than this 36 x 36 px target which makes it difficult for the user to affix on a target and interact with it.

There is also the issue that if a map is presented at too high a level; there may be many pinpoints found at the same location on the same screen. Differentiating between these pinpoints via touch is a frustrating experience. This may be handled by using notifications that allow the user to touch a single point and then choose from a list of pins or by presenting a map at the appropriate magnification to ensure all pins are distinct and separate.

### **Help and Escape**

The final tip for implementing maps is one that applies to many areas of UX design. Always give your users a little assistance (you can use pop up messaging to aid understanding of gesture controls or other functionality that may require explanation, for example) and a way to get out of the map and back to a more familiar interface.

### **Help in mobile applications**

When users start getting to grips with your mobile app they're going to need a little help from time-to-time. The good news is that there are standard design patterns which can be employed to reduce design time for help in mobile apps. There are also some useful guidelines that can be employed to improve the UX of help in general. Familiarizing yourself with these can make your help functionality more valuable to your users. Users sometimes need help. Mobile apps are still new enough that not every user is familiar with conventional usage patterns and most mobile apps are likely to introduce new concepts or ideas at some point in their interaction with the user. The good news is that there are some excellent general best practices for help design and some popular design patterns that you can utilize too.

### **5 General Best Practices for Providing Help in Mobile Apps**

**Help (and/or support) should be very easy for the user to find.** In apps they are most likely to look to the tool/tab bar for this.

**Users should be able to access more than one form of help and/or support.** This can come in the form of tutorials, self-service FAQs, click to call to speak to an agent, notifications, etc.

**Tutorials/introductions should be utilized whenever a user first interacts with an app.** There should always be the ability to skip this and come back to it later.

**In app help content should be made available when functionality is added or changed and when rarely used functionality is accessed.**

**Help videos can be really useful but only if the user has complete control over playback** – this includes the facility to start/stop/pause, move to the section they want to view and the volume of the playback.

### **6 Great Help Design Patterns for Mobile Apps**

Demonstrations / Tutorials / Single Screen Overlays / Walkthroughs / Tips / Single Screen Summaries

#### **Demonstrations**

A demo is generally a video or animated walkthrough of key functionality. They are best kept short and relate to a specific area of functionality. **You can use more than one demo** if you think the user will benefit from it.

Demos show the user the app without them having to work through a lot of screens to do so, they clearly articulate relationships between app elements and can articulate on and offline usage.

However, it's worth noting that they **should contain play controls, only be used when the app first launches** and be annotated sufficiently to allow the user to recognize their value.

They may not be completely successful in helping the user if too much is incorporated within the demo without a clear indication of context. It's hard to remember too much information without putting it into practice.

## **Tutorials**

Tutorials guide the user through the interface with simple explanations of interactions and UI elements.

While you don't want to serve a mega-long tutorial to a user on the first interaction; you can use tutorials to demonstrate as much of your functionality as you want - just make sure they're easy to find at a later date.

## **Single Screen Overlay**

An overlay is placed over the screen and has **up to 5 points of interest** explained. The overlay **should be easy to dismiss** when the user is done with it.

Overlays can be used throughout the application where necessary. You need to keep explanations simple to ensure readability. There should always be a means to recover an overlay if it is closed by accident. Overlays are great for explaining concepts within the context of any given interaction and can be really useful when explaining unusual or uncommon functionality.

They aren't as good when you're explaining highly complex functions (not enough space on screen) or processes that move from screen-to-screen.

## **Walkthrough**

A walkthrough takes a user through a process from start to finish; explaining and then getting the user to carry out each step along the way. This can be very useful to encourage additional use of the application or to encourage a user to try out different possibilities within an app.

**The best walkthroughs are limited to a couple of key tasks within the app.** They are also well introduced with the process objectives clearly explained before the walkthrough commences.

**Be careful that your use of walkthroughs doesn't "break" the user's own journey with the product** – they should be delivered "just in time" not "all the time". Very complex tasks can be overwhelming with a walkthrough – consider splitting tasks to make them simpler to carry out.

## **Tips**

Tips are designed to deliver **a single piece of information exactly when the user needs that information.**

They can be used when an app first opens, when a user enters a screen or when a user completes a step in a process. They are great for introducing new functions.

There should always be a way to turn off tips. They are best used to convey simple ideas and not to replace tutorials, walkthroughs, etc.

## **Single Screen Summaries**

This is a simple overlay with a short bit of text describing the screen about to be used. It can either be closed by the user or just vanish within a few seconds (around 5 seconds). **It should only be used the first time a screen is accessed and not be repeated every time a screen is used.**

The main drawback to single screen summaries is their ability to get lost in the mix for the user and they cannot be recalled.

## **Navigating the Maze of Mobile Apps – Design for Mobile App Search**



In the digital age, people have come to rely on search as one of, if not THE, foremost tools for navigation in applications and online. Integrating search into mobile applications can be straightforward and users will thank you for it. The mobile user experience relies just as much on effective search as the desktop one.

Thanks to the efficiency of Google and with a nod to many other well put together search engines – search has become the leading means of getting around the internet and getting round the insides of many applications. That means that mobile applications also need to take advantage of search functionality in order to deliver user experiences of a sufficient standard.

## **The 8 Patterns of Mobile Search**

The good news is that search is a well-understood area of functionality. There are 8 common patterns that are applied to mobile search and they can be used either individually or in combination with each other for maximum effect.

Auto-Complete / Dynamic Search / Explicit Search / Faceted Search / Saved and Recent / Scoped Search / Search Forms / Search Results

### **Auto-Complete**

This is often used in conjunction with explicit search; it's the idea that when a user begins typing a word – the software looks to complete it for them. This is particularly important on the mobile platform where typing is a matter of thumb presses rather than using a keyboard.

### **Dynamic Search**

Dynamic search uses a form of filtering, so that when a user begins to enter data – the content already on the screen is filtered based on the search field. This can be very useful when a user is trying to wade through large volumes of data – for example, a list of hotels in a certain area.

It's worth noting that dynamic search can be limiting when using huge data sets that are pulled from different data repositories.

### **Explicit Search**

This type of search requires an action from the user to start the search; such as tapping a button on the display or perhaps using a keyboard control. Results from the search will, normally, be shown directly below the search bar.

Explicit search can be used in conjunction with auto-complete for better UX on mobile devices.

### **Faceted Search**

Faceted Search is a more complex version of scoped (or filtered) search. It tries to deliver a huge number of search filtering criteria and allows the use of multiple filters to narrow down huge data sets. This is commonly implemented in a tray pattern that enables a user to set filters without having any prior knowledge of Boolean logic.

The controls and the results will be displayed at the same time, so that users can make tweaks to their filters “on the fly”.

### **Saved and Recent**

This search pattern recognizes that users often want to return to a search that they have used in the past. It allows the user to save their searches (or does this automatically on behalf of the user) and then to recall these searches at a later date.

A saved search will normally require user action to retain the search for later use; a recent search will, usually, be stored automatically by the app with no action required by the user.

### **Scoped Search**

A scoped search allows the user to define criteria for the search before they begin their search. For example, Google allows users to choose between image searches, web searches, news searches, etc. before entering the data.

The designer needs to examine the data set carefully for the search criteria and then focus on allowing simple filters that deliver value to the user. Remember the limited screen real estate on mobile and try to keep the number of filters to a minimum whilst delivering value.

### Search Forms

If you need a variety of data to conduct a search, for example if you ran a flight booking service – you might need to know the date of departure, date of return, number of people flying, seat preferences, destinations, etc. then you can use a simple form to collect all the data to deliver quality search results.

The idea is to keep this as simple as possible without depleting the quality of results. Only use fields that are absolutely necessary and don't waste screen real estate or inconvenience the mobile user with a limited amount of typing patience. It can also be valuable to focus on operating system input techniques when it comes to designing mobile forms.

You should, as always, follow form design best practices to ensure that forms deliver maximum usability and utility.

### Search Results

Once a user has performed a search, you need to share the results with the user. This can be done in a number of different ways:

- Thumbnail images of relevant results – for products, pictures, etc.
- Tabulated results – ideal for flights, hotels, etc.
- List results – a standard search result for data based pages, etc.
- Map results – where location matters

You can also offer a mixture of the formats or the ability for the user to switch between one format of result and another. The ideal is to find a default display which is the one the majority of users prefer and then to offer transitions to less popular choices.

With mobile bandwidth being constrained and data costs being met by the user of mobile apps or mobile websites; you may want to provide a small handful of results and allow the user to extend the results list if they can't find what they want in the original results.

It can also be valuable to offer an indication of the total number of results found so that the user can decide whether to scroll through the list or if it is dauntingly large – to refine their search before exploring the list.

### Evaluating your Design

mobile attention span: **4 - 8 seconds** vs 11s on a desktop

mobile content comprehension **50% less than desktop**. Do not take for granted your 'relaxed' cognition from your desktop.

To test - 3 stages: Preparing, facilitating and sharing your results

### 30 sec usability test:

Ask for feedback on a design:

- provide paper / proto etc
- ask them to "think aloud" - tell me what you are thinking and feeling
- ask "what would you expect to happen here?"
- point or press the paper / screen / iPad
- thanks them

Don't ask if they 'like something' because this is just an opinion.

## Evaluating using scorecard



### Rating your feature

Determine how much you agree with each statement:

- +2 Completely agree
- +1 Somewhat agree
- 0 Neither agree nor disagree
- 1 Somewhat disagree
- 2 Completely disagree

**Usable: Prevent errors and reduce confusions.**

- "The feature prevents users from making errors whenever possible".
- "The feature provides a cushion when a potentially serious error is about to be made".
- "Navigation and visual elements require no interpretation past a few seconds".
- "The user is able to easily reverse or continue their task without any problem solving".

**Satisfying: Make it "dummy proof".**

- "The feature requires no interpretation- it will offer an affordance for use".
- "The user can easily find what to do next".
- "The feature requires no understanding, knowledge or any prior background or knowledge".
- "The user can perform tasks successfully without help, training or discovery time".

