Usability Matters

Mobile-first UX for developers and other accidental designers

Matt Lacey

MANNING

Usability Matters

by Matt Lacey

Chapter 1

brief contents

1 Introduction

This chapter covers

- What usability is, and why it matters
- How user experience contributes to success
- The six components of great apps
- Why each component is important

As someone involved in the creation of mobile apps or games, you'll want what you create to be as successful as possible. Regardless of how you define success, you can't overlook the factor that the experience you create for the people using your app plays a role in achieving it. Throughout this chapter, I'll introduce six components that contribute to great, successful app experiences, but I'll start by defining what I mean by usability and the experience of using an app.

1.1 What's usability, and why does it matter?

When it comes to creating apps, you'd probably rather spend most of your time focusing on code. Sadly, the successful developers I know say that the code only accounts for between 10% and 40% of their success. Many factors contribute to a successful app, but I believe that expanding your focus to also think about the experience of using an app is the easiest way to make simple, tangible improvements.

In this book, you'll learn how to think about usability and the experience you'll want your app to convey. Plus, you'll gain lots of practical tips and general advice for changes you can make to improve them. Rather than tell you how to address specific scenarios, I'll give you the knowledge to address all scenarios. These aren't the things you'd normally see in a book about design—that's because this book is specifically for developers like you.

Usability beyond mobile apps

You can apply the principles, knowledge, and lessons in this book to more than mobile apps. While this book focuses on mobile app examples, the principles behind the advice can be applied to websites and apps that run on any device.

In some ways, building for all the constraints of mobile can make other software development seem easier. If you can create an app that works on multiple devices, it's much easier to build something that has to work only on one. Similarly, if you're able to optimize for multiple forms of input or output, then non-mobile software that doesn't need to account for as many situations can seem simple by comparison.

1.1.1 Usability matters to everyone

If you're making the conscious decision to expand your app development knowledge beyond just code, or you're looking for a way of making your app stand out among the competition and keep people using it, usability should matter; it matters to the people who use the app. Usability is about how easy your app is to use. People don't want to spend their time trying to work out how to use your app or remembering how it's different. An app that's easy and intuitive will keep people using it repeatedly. They'll also be more inclined to tell others about it in a positive way.

1.1.2 Usability, UX, and design

Design and *designer* are terms loaded with nuance and interpretation. They mean different things to different people and cover a broad array of concepts and topics. Everything from graphic design to branding, animation to copywriting, iconography to user interfaces, usability to human-computer-interaction, and user experience (UX) to information architecture, all fall under the umbrella of design. You don't have to be an expert in each area of design, but knowledge and awareness of each will help. This book covers many aspects of design, but two that need special mention are usability and UX:

- *Usability*—An aspect of design that looks at how easy something is to use. The only way to measure usability is by testing it with real people who use apps. We'll come back to testing many times throughout this book, because getting feedback from the people who will use and are using your apps is important. To learn how to run your own usability tests, I recommend reading *Rocket Surgery Made Easy* by Steve Krug (New Riders, 2009).

- *UX*—A popular term that many people have appropriated in recent years. It's a formal discipline with recognized practices and techniques. More than just focusing on software design, it encompasses all aspects of interaction with a product, service, or company.

This book isn't trying to teach you to be a UX expert or to give you license to call yourself a designer. This book will enable you to take the apps you're already developing and create better experiences for the people using them so you (and your app) can be successful.

1.1.3 The formula for app success

App success is based on three important factors: value, experience, and luck. You can be successful with one, but striving to address all three is best. The distinction between these factors can sometimes be blurry, but this book focuses on the second. It's about the experience that you design for the people using the app.

Please don't make the common mistake of thinking that usability and UX are only about how your app looks. The experiences created by using your app can be much richer and more nuanced when you consider more than just the visuals. If this weren't the case, this would be a book about user interface design rather than usability and the user's experience. In this section, I define the three success factors:

- *Value*—What a person gets from using an app. This could be a single benefit or many.
- *Experience*—Relates to how the value is provided. It's about the feelings and emotions that a person has when using an app.
- *Luck*—The variable that you don't have complete control over.

Let's look at value first. Some ways that your app can provide value for the people using it are

- By making a task possible
- By making a task simpler
- By making a task faster
- By earning a person money
- By saving a person money
- By entertaining
- By informing
- By educating
- By serving as a distraction or way to pass the time (although this is rarely a good reason for creating something)

Your app's ability to meet the needs and desires of the people using it is part of the value it provides. You need to make sure you're building the right thing for the people looking for the value you provide, not just making something possible. Sometimes a mobile app isn't the best way to meet a person's needs or to provide the value they seek.

NOTE Meeting user needs is the fundamental principle of providing value. If you can't provide value, it doesn't matter how good the experience is. People won't use the app.

The experience you create for the user is dependent on the value you're providing. There may be some overlap between the two when defining the context of use and the target market for the app. Experience is also, in part, about how you deliver value. The experience you create for the people using the app is what impacts success. Traditional UX tasks, such as user research and testing, can contribute to the experience you create, but they aren't the same thing. In most cases, you'll want the experiences you create to be positive, but this isn't always the case.

When you might want to create a negative experience

You will want to consider two occasions when it might make sense to provide a negative experience. The first is to discourage bad or negative behavior. For example, if a player was to turn on their teammates in your game, you might want to do something to discourage them from doing this again. You could reduce the effectiveness of their armor, restrict the strength of their weapons, artificially limit the responsiveness of the game, or include another creative way of letting them know the consequences of doing things they shouldn't. Or, your app can force a negative experience upon the person using it when they do something that's detrimental to the other people who use the app. If someone's repeatedly reported for spam or offensive content in your chat app, you may choose to limit the length or frequency of messages they can send.

The second reason for deliberately creating a negative or challenging experience is if you want to increase positive emotion in the future. (This is a common strategy used in games to keep people playing.) If someone must struggle and fail a few times to achieve something, such as completing a level or defeating an opponent, they'll feel much better when they do achieve their goal. In this scenario, the positive feeling will be higher after they've failed a few times than it would've been if they had achieved their goal on the first attempt.

Creating a great experience in your app is important because you can't rely on luck. For example, large companies that have had success in the past release new games to a muted response. Or, individual developers have seen their games suddenly become viral hits months after release. Luck is as good as any name for this factor, but it isn't totally out of your control. Becoming a viral hit, being repeatedly featured in the app store, or having positive reviews in the press isn't something anyone can guarantee, but there are things you can do to increase your chances of these happening and the luck you and your app experience.

In the first century AD, the Roman philosopher Seneca said, "Luck is what happens when preparation meets opportunity." It still applies today. You can't guarantee the opportunities you'll have, and you may even have to work to create them. But if they do come and you're not prepared, you'll miss out.

You're more likely to create something lots of people use if it's something they want. People will share an app more if it's easy and there's a benefit. The press and websites are more likely to write about your app if it's unique, if it's high quality, and if you tell them about it and ask for a review. Stores are more interested in featuring apps that are new, of high quality, and show the unique features of the platform.

This book is about experience, as that's the easiest factor for you to control in a way that can be beneficial to your app and help it stand out from the competition. Value is often easy to compete with as functionality is easy to copy. Luck is partially out of your control, so it isn't something you can rely on. The experience of an app is embedded deep in its design, and that's the hardest to replicate. But a great, intuitive experience can enhance its value and increase your opportunities for luck.

1.1.4 Great usability experiences are intuitive

Intuitive is a word that's often used to describe apps. I regularly see release notes claim, "Now, more intuitive to use." There's a problem with such a claim. It's that whether something is intuitive is subjective—for something to be intuitive, it must meet the expectations of the person using it.

You'll find some official lists of things software should do to be considered usable at www.usabilitynet.org/tools/r_international.htm, but you can't tick off the items on a checklist and then claim your app is intuitive. You can claim that something is intuitive for you, or you can mention other people who've said they find it intuitive, but you can't claim something is intuitive for me. Only I can say if I find your app intuitive once I've used it.

To find out if people think your app is intuitive, you must ask them. This raises the question, "How can this book show how to increase the usability of an app and make it more intuitive if that can only be determined by the people who use it?" This is a good question and shows you're paying attention. Before releasing them more widely, test the apps you create based on the advice given in this book with a sample of real people the apps are intended for. Only the people the app is intended for can say if it's right for them.

This book provides guidance and instruction on improving your apps based on common conventions, best practices, and many years of investigating what features of hundreds of apps led to people having a positive, intuitive experience. Making something intuitive is like the ideas behind the principle of least surprise: software should behave as the person using it expects it to. Applying these concepts to an app isn't intuitive itself. To help you think about all aspects of an app, I've broken the app experience into six components, which I'll introduce now.

1.2 Six components of great app experiences

You can think about or visualize the structure of an app in many ways. Throughout this book, we'll look at six different components of an app experience:

- Context
- Input
- Output
- Responsiveness
- Connectivity
- Resources

Five of these components relate to logical areas of functionality, or parts of the architecture of an app. The sixth, context, is about how to view the app, so we'll look at that first.

1.2.1 Context of use underpins everything in an app

Context is the situations and circumstances surrounding the use of an app. It's about looking at the big picture and seeing how the app fits into the wider environment. Think of considering context as looking at your app through a pair of glasses with three lenses, like that shown in figure 1.1. Each lens allows you to see a different aspect of the app's context, and it's only when you see all of those that you get the complete picture.

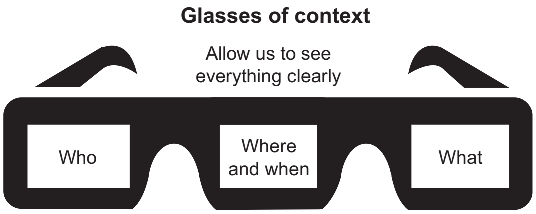


Figure 1.1 Considering context is like looking at an app and seeing three different aspects of its use.

An app that considers context appropriately provides an experience that fits seamlessly into the life of the person using it; it meets their needs and works as expected. An app experience that doesn't fully consider context can end up not working for the audience it was intended for, because it lacks what they want and performs poorly on the devices they're using. Ultimately, people will struggle with the motivation to keep using the app. Different people have different needs, expectations, and requirements. Considering the people who'll use your app allows you to be sure you're providing what's right for them.

Also, the environment where the app is used can influence how a person interacts with it, and the location can provide extra hints to help optimize what a person experiences from an app. Knowledge of the capabilities possessed by the devices your app will run on allows the creation of an app that makes use of the facilities and functionality available to it. This knowledge also enables the creation of an app that can integrate with the wider experience of using the phone and the other apps installed on the device.

Ignoring context is like only testing your app with a mouse in the simulator on your PC while sitting at a desk when real people using the app will do so by hand, in the field, with poor connectivity, and while distracted by another part of their job. In chapters 2 through 4, I'll take you through the elements that make up the circumstances of use to consider as the context.

1.2.2 Input includes all ways data and information get into the app

Input encompasses all the ways of supplying data and instructions to the app. It's the first of the components that relate to the high-level structure of the app. Figure 1.2 shows these components (in bold) and their relationships.

Input for a mobile app can be about much more than tapping a screen and filling out a form. An assortment of input sources and mechanisms are available from the

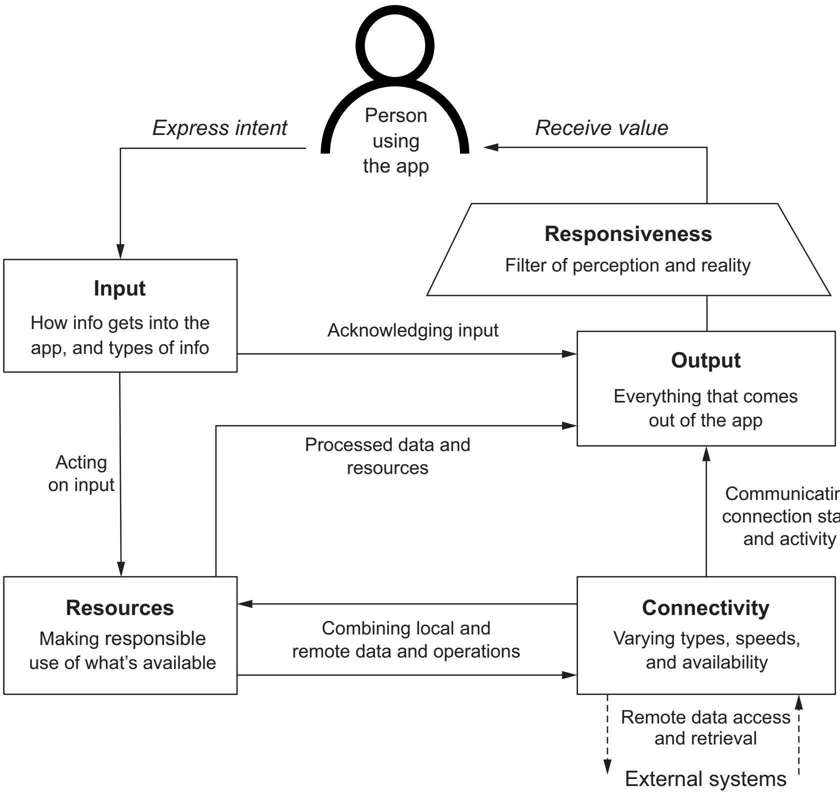

Person using the app
Express intent
Receive value
Responsiveness
Filter of perception and reality
Input
How info gets into the app, and types of info
Acknowledging input
Output
Everything that comes out of the app
Processed data and resources
Acting on input
Communicating connection state and activity
Resources
Making responsible use of what's available
Combining local and remote data and operations
Connectivity
Varying types, speeds, and availability
Remote data access and retrieval
External systems

Figure 1.2 How the five structural components of an app relate to each other, the user, and external systems

device, the person using it, and remote sources. For the people using the app, it's rarely their aim to use an app just to gather input. Whether saved for later use or to give an immediate reward, aim to optimize the input-gathering elements of an app to the point where the people using the app can receive the maximum benefit from their input.

NOTE You can avoid much direct input, but when necessary, gather it quickly, simply, and accurately. Whether the input is optional or necessitated by the app, input is a means to an end and not an end in itself.

If you're not considering input fully, you may end up creating a long, slow, tedious, and error-prone process that degrades the experience of the people using it to one that frustrates and annoys. As an example of how input can impact usability, consider the need to get a person's location. Three ways of doing this, each with different user experiences, are using the device's GPS, tapping a point on a map, or typing in a street address. Chapters 5, 6, and 7 dig deeper into considerations of how your app can handle input.

1.2.3 Output includes and goes beyond what is shown onscreen

When considering output, the first thing you'll probably think of is the screen. Devices in use today have screens that cover a broad spectrum of physical dimensions. You can't always use the identical output on a 3 in. screen, on a 10 in. screen, or in a response that's read aloud to the person using the app. It's essential that everything displayed on any supported device is readable, usable, helpful, attractive to view, and easy to navigate.

Output is also about more than what's displayed onscreen. In addition to visual output, there's also audio and physical output to consider. Beyond that, you need to consider what is sent to other systems and services as well.

When the output from an app is poor, it leads to an experience that's unusable in some contexts or by some people. Why would you want to make your app unusable? On the flip side, an app that meets the needs of all use cases ensures that people can receive the information and notifications they want in a simple and easy-to-consume manner. Chapters 8 and 9 cover the options and specific considerations for providing output as part of your mobile app experiences.

1.2.4 Responsiveness: how output is perceived

People use apps because of the value they'll gain from doing so. Sometimes the experience is enough on its own, but people typically want value. Because of this, it's beneficial to optimize for that and to avoid having people wait unnecessarily before providing it. A responsive app doesn't waste people's time, but helps them use their time efficiently and productively.

Responsiveness is the filter through which a person interprets the amount of time a task is perceived to take. To create an app that's responsive involves controlling and influencing the perception of the time a task takes. Often this means making something as fast as possible and then giving the impression that it takes even less time.

Responsiveness is about more than just speed; it also ensures that what's output is appropriate and targeted to the circumstances of the people using the app. It's the difference between someone being unsure about the app (because it shows a blank screen while it's presumed to be doing something), and someone confident that the app is busy working as hard and as fast as possible for the desired outcome. Part 4 of this book is about responsiveness and what you can do to objectively improve the time your app takes, and to improve the subjective perception of how long a task takes.

1.2.5 Connectivity changes and isn't always guaranteed

While some games and utility tools, like a calculator or flashlight, need no form of connectivity, the majority of apps depend on data that will change. So they'll need to be updated or will be dependent upon interaction with another person, machine, system, or service, such as a backend server. Because apps are designed to help the person using them, it's incredibly frustrating when a connectivity issue leads to lost work or unavailable information that's needed.

If an app that needs server connectivity suddenly loses it, the experience of the person using the app is determined by how the app handles the situation. If the app loses data and stops functioning, or crashes, it reflects poorly. For apps built to survive such scenarios, people can continue using the app, confident that if there's a temporary setback, their information is safe, their progress is remembered, and everything will carry on as normal once connectivity returns. There's a lot more to consider about connectivity, but the implications of not handling the loss or lack of connectivity are easy to appreciate.

If you're fortunate enough to live in a part of the world where a 4G signal is readily available and Wi-Fi is ubiquitous, it's easy to think that everyone else also has the same situation. This isn't the case for a large part of the world. Care should be given to what happens in an app when someone doesn't always have a fast, cheap, reliable connection. The way variations in connectivity are handled in an app can be the difference between an app that's unusable and one that always provides some value, even if that's showing older, cached data or saving gathered information to upload later.

Chapter 13 explains what to consider when regarding connectivity. In it, I'll talk about how the type of connection can vary, how the connection can't and won't always be possible, and what you can do to create the best experience for people when they don't have perfect network conditions.

1.2.6 Resources are finite and must be managed

Due to their portable nature, mobile devices have several constraints and restrictions. Device manufacturers must balance the size and number of components they can fit into the dimensions of a specific device. Similarly, they must balance the cost of the components against the cost people are willing to pay for a device.

All this leads to a situation where your app will often end up running on devices without the space, power, or capabilities you'd choose. Yes, there are people with devices containing several hundred gigabytes of storage, more CPU and GPU cores than you can count on your fingers, and more sensors than you might believe can fit inside the space available. But these are the outliers. Even if you're in the position of being able to know that all the people using your app have super-high-specification devices, you can improve the experience of using your app by fully considering how to use the available resources. Doing so will also help anyone who doesn't have such plentiful resources.

The most important resource to consider is power. As a mobile device isn't always connected to a power source, power must be considered as limited. Aside from some line-of-business scenarios, people use their devices for many things, and in all parts of their lives. If using your app drains the battery, you don't just prevent that person from continuing to use your app, you also stop them from living their lives as they want or performing other, possibly more important tasks. Your app or game is important. It just might not be *as important* as an email about a job offer, a phone call from a relative, or messages from a romantic interest.

Apps that respect the resource constraints of mobile devices don't just avoid draining the battery, they also don't fill the disk with unnecessary files that can't easily be deleted, and don't prevent other apps from making use of all the sensors and capabilities of the device. Chapter 14 will help you better understand and consider how to use resources within an app and what can be done to prevent unnecessary power usage. This will help you ensure that the value your app brings to the person using it isn't canceled out by negative consequences of preventing that person from doing other tasks.

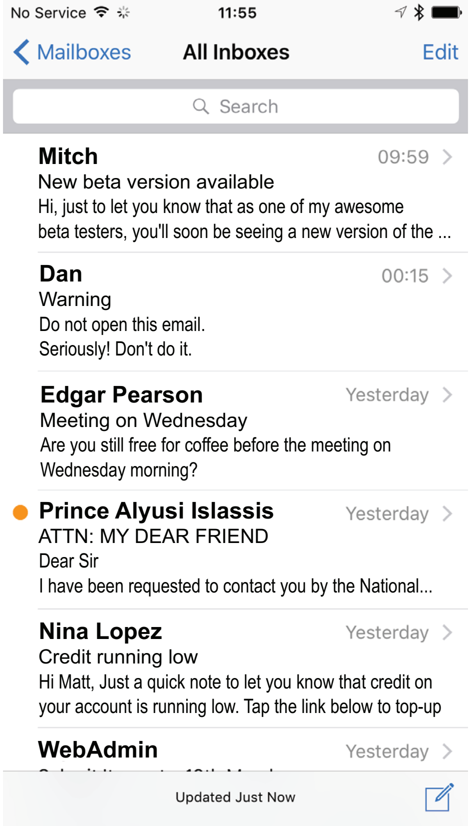
1.3 How considering all six components can make apps better

The six components are designed as a structure for thinking about the experience of using a mobile app. By considering how each component impacts the app that you're building, you can ensure you're creating something that meets the needs and expectations of the people using it, works well on the devices it runs on, and overcomes the challenges of being mobile.

Right now, you're probably thinking this all sounds good in theory, but how does it work in practice? Let me walk you through three examples to show you how the six components can help highlight ways of improving users' experiences.

1.3.1 Example 1: an email client

Let me start with something familiar and that happens every day—an email client (figure 1.3). You may think email clients have been around long enough and are so well known and understood that there can't still be room for improvement. Not so. By using the six-component structure, in just a few minutes I came up with the following ideas on how to improve the experience.



Mitch 09:59
New beta version available
Hi, just to let you know that as one of my awesome beta testers, you'll soon be seeing a new version of the ...

Dan 00:15
Warning
Do not open this email.
Seriously! Don't do it.

Edgar Pearson Yesterday
Meeting on Wednesday
Are you still free for coffee before the meeting on Wednesday morning?

● **Prince Alyusi Islassis** Yesterday
ATTN: MY DEAR FRIEND
Dear Sir
I have been requested to contact you by the National...

Nina Lopez Yesterday
Credit running low
Hi Matt, Just a quick note to let you know that credit on your account is running low. Tap the link below to top-up

WebAdmin Yesterday

Figure 1.3 An email client

Context

Context covers a wide range of users, devices, and locations, as everyone uses email. Even if it isn't their primary method of communicating, it's almost impossible to use other services, including mobile phones, without an email account.

We know that on mobile devices people may not have a lot of time, and so a person using the device may get an idea for an email, but they don't have time to send it now. This suggests that the ability to start, save, and resume drafts of partially completed emails may be important, and so provides an area in which you can improve the experience.

Additionally, if a person is using email on multiple devices, it may be common that they read an email when it arrives on a phone but want to respond from a different device. There may be an opportunity to make this easier for people by adding a simple way to affix a specific flag for such messages. The benefit would be to distinguish any message for follow-up and any that will need to be on a specific device for responding.

Input

Email found popularity on the desktop, where messages are primarily text-based and entered via a keyboard. On mobile, this isn't always the case. Adding attachments or embedding content from other apps is desirable, but is hidden by some mobile mail clients deep inside a multilevel menu. Making this easier would help users who want to compose rich emails, some with multiple attachments, on a mobile device. Related to this is looking for easier ways of integrating with other apps.

Selecting who to send to should have a smart search that doesn't just rely on names, but also uses parts of domains. You can send me an email via me@mrlacey.com. If this were saved in your contacts without my name, it'd be hard to find (based on common search algorithms for name matching). But if you started to search for my surname (Lacey), that should find the address. Unfortunately, many email clients don't have such advanced search capabilities. Adding this ability will help make it possible to compose emails faster.

OUTPUT

Many emails are still sent with formatting, assuming they display on a large screen. This isn't the case for emails read on mobile. Being smart about reformatting or adding an easy way to switch to a text-only view if reformatting isn't possible could make many emails much easier to read, compared with manually zooming to view and navigate.

In addition to the grouping of threaded or otherwise related emails, it may also be useful to indicate if unread mail from the same sender has been received, even if not in the same thread or visible among the messages onscreen. This would help avoid situations where someone reading messages in the order they were received replies to a message without first realizing they've received another email that nullifies the first or provides additional information that would be useful when replying.

RESPONSIVENESS

Email clients normally respond to the receipt of new emails in a way that applies to all messages, but there can be times when this isn't desirable. Just as it's possible to configure my phone to audibly ring when certain people call me, even if the phone's in silent mode, there may be times when the same is true for email. My wife rarely emails me, but when she does, it's normally about something important. I receive several hundred emails a day and so don't want notifications about all of them, but any from my wife, I'd prefer to be notified about regardless.

When sending a message, it's often a case of pressing Send and hoping everything goes through OK. After pressing Send, a person may move on to doing something else while the app sends the email in the background. If something goes wrong and it's not possible to send the message, or a delivery failure response is received, the app should notify the person accordingly. It's not helpful for the email to remain in an outbox only for the person to discover this when returning to the app later. It's better to aid the process and help the person achieve their goal rather than being a passive bystander in failure.

CONNECTIVITY

If a person tries to send an email when they don't have a data connection available, the app could store that message and automatically send it when a suitable connection is available. The person who sent the email wouldn't have to concern themselves with whether they have connectivity and with manually triggering a send.

If people try to send emails with large attachments when using costly or slow network connections, they might appreciate a prompt to confirm the desired action. Doing so may help them avoid an unnecessarily large bill or a message not being sent, as they expected it to go quickly.

Resources

Mail clients are generally good at managing resources related to data network connections, but there isn't always the same consideration paid to local resources on the device. When attaching a large file to an email, some clients create additional copies of the file, which can impact disk space and I/O. This should be avoided. Also, loading the contents of any attached file into memory before sending is an extravagant action that can slow the performance of the app and unnecessarily consume resources.

1.3.2 Example 2: a news app

As a second example, we'll consider a simple news app, like the one in figure 1.4. You'll find many apps like this in all the stores.

This news app shows a list of stories, grouped by categories. After selecting a headline, it's possible to read the story and see one or more related images.

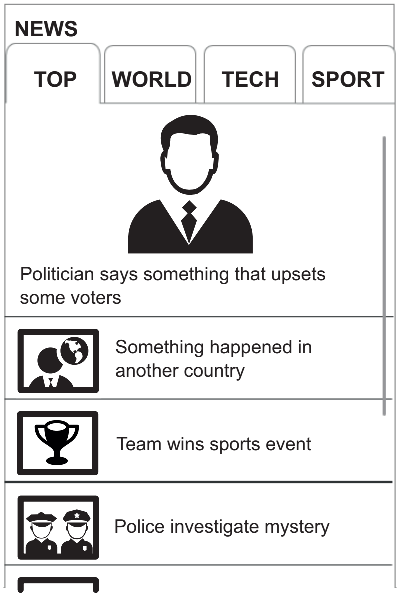

NEWS
TOP
WORLD
TECH
SPORT
Politician says something that upsets some voters
Something happened in another country
Team wins sports event
Police investigate mystery

Figure 1.4 A generic news app

Context

Because there's lots of competition in general-purpose news apps, specific targeting of content would provide a way to make the app unique and help appeal to and meet the needs of a particular audience. Short, text-based stories are good when a person has limited time, but it'd be beneficial to find ways to provide or link to more or related information if desired.

Input

Rather than show the same news to everyone, it may be possible to use the location of the device to identify and highlight relevant local news. This may be of greater interest and importance to the person using the app. Other information and apps on the device may also make it possible to identify stories more likely to be of benefit to the targeted audience.

Instead of showing the same stories, in the same order, to every person using the app, it's also possible to use a record of the stories the person has viewed previously to prioritize new stories. Working on the assumption and expectation that people will continue to be interested in the same topics they were interested in previously can be a way to highlight relevant stories a person would miss if otherwise sorting stories based on mass appeal. By tailoring the content without the person needing to do anything other than use the app, the quality of the experience and potential value grows over time with increased usage.

NOTE In this example, it's also important to be aware of the potential risks of any activity that filters news so you don't create a news bubble or reinforce a bias by only showing one side of a story.

OUTPUT

If people have little time to catch up on the news, they'll benefit from the display of easy-to-read summaries. They can also read the full version if they have more time available.

When people don't have the time or inclination to read a story, they may appreciate other ways to keep track of current affairs. This could be a pictorial view of images relating to stories, or a way for the story to be read aloud by the app so they can listen to the stories of interest while also performing another task. Formatting or presenting content in these ways achieves a better experience by ensuring that a person gets the news they want in a way that matches their needs.

RESPONSIVENESS

For an app based on providing information quickly, it's essential that the person who wants to use the app doesn't spend time waiting for content. To ensure that the reader can get the latest stories quickly, it could be beneficial to preload any recent stories in the background before the app launches. This works well if the use of the app occurs at predictable times or intervals, such as in the morning and evening. For example, the app could load new content in the early hours of the morning or late afternoon, ready to delight the reader by always having the latest content waiting for them when they launch it at breakfast or on the train for their commute home. Preloading images relating to stories would also allow content to load faster.

Building on the idea of using analytics from past usage as a heuristic for the stories that are likely to be of most interest to the person using the app, this data could be used to filter breaking news notifications. Rather than treating everything new as breaking news, filtering content would allow the app to notify people of the latest updates only on topics that interest them. This would mean the app's always there with relevant content and that it could preempt a person's desire to hear about what's most important to them.

CONNECTIVITY

Preloading content would also serve to ensure that there's content to display even while launching the app when there's no way of reaching the server to get the latest stories. As long as the age of the stories is made clear to the person using it, it's preferable to have some content to show the reader, so they can always read something regardless of network connectivity.

Loading data in the background can also be expensive, based on the location, type, and cost of connection available. Making this functionality configurable would be desirable for some people who'll use the app.

RESOURCES

If the app takes input from other local sources, such as location or other data stores, you should consider the frequency of such checks and the cost of available computing power and battery resources. If preloading content and pre-caching images or videos, you need to

- Be careful of consequences to disk space usage
- Ensure an adequate process is in place to remove older files when no longer needed
- Make certain that the device maintains an adequate amount of free disk space

People are seldom impressed when improvements to the experience of using a single app come at a cost that prevents using the device for anything else.

1.3.3 Example 3: a tower defense game

As a final example, we'll look at something different—a tower defense game. Figure 1.5 shows an example of this.

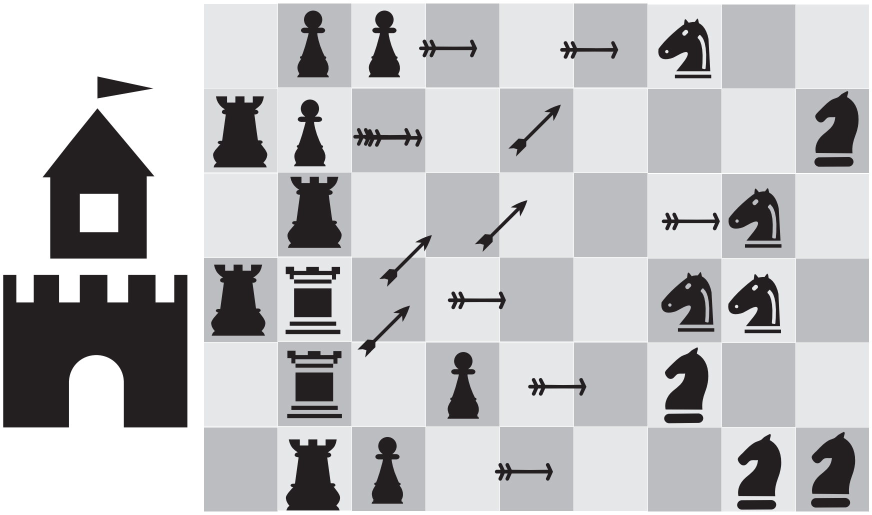


Figure 1.5 In this chess-inspired tower defense game, rooks and pawns are placed on the board to stop the knights advancing from the right as they try to storm the castle.

This is a popular strategy game where the aim is to position resources (towers) on a playing space to defend against approaching enemies. When successful, this earns a form of currency to obtain more towers to help guard against more formidable opponents.

Context

You'll find lots of competition for this type of game. To stand out, you'll need to find a way to differentiate your app from the alternatives and target a specific audience for it. Games like this often require playing for several minutes at a time. This contrasts with the common desire for short game play on a mobile device. The ability to have short as well long games may be appealing to players and will remove barriers to playing.

The size of the screen the game runs on impacts how much is visible or the level of detail, and may make play easier on devices with larger screens. You'll need to decide if this matters. In a single-player mode, this won't make a big difference, but it can create a disparity between players on different-sized devices who are playing against each other.

INPUT

With finger-based input, the hand obscures much of the screen and playing area on smaller devices. This makes it easy to accidentally position a tower in the wrong place. Having the ability to undo or modify such actions aids players' confidence; they won't suffer as the result of an accidental action or make a mistake from which they can't recover.

OUTPUT

With this style of game, there are times when things are quiet and other times when there's a great deal going on at once. During the busy times, the players may appreciate the highlighting of important events through appropriate sound effects and haptic feedback, in addition to information visually appearing onscreen. Some players might see this as making the game easier, but others may appreciate the assistance.

RESPONSIVENESS

With a graphically rich game targeting many different devices, it's necessary to have multiple copies of the visual assets used in the game, adjusted for different screens. An app will be faster if it only loads the images that are necessary and optimized for the specific device. There's no point taking the time and resources to load images the game won't use or that are bigger than necessary when equally good smaller assets are also available. By making the game load faster and providing informative feedback during the loading process, the delay for the person waiting will be minimal, and they'll be confident in what the app is doing while they wait.

CONNECTIVITY

A single player-mode means that the game can be playable offline. Having this would be in addition to an online mode that allows people to play against each other.

With an online mode, there'll be a need to handle connections lost part way through a game, without penalizing a person for network conditions beyond their control. Balance this with the need to consider players deliberately breaking the connection with the server and leaving the app when they're losing. The best behavior for this scenario will be dependent on the specific rewards and penalties the game employs for winning and losing. The key point is to ensure a player isn't penalized due to factors beyond their control. An online game would also need extra considerations for backend security to prevent other forms of cheating or misuse by unscrupulous players.

Resources

Because what happens onscreen varies greatly between times when little is happening and times when there will be lots going on, this presents an opportunity to save resources when updating the screen. If nothing's changing or moving onscreen, there's no need to repaint the entire screen at a rate of 60 frames per second. By adjusting the refresh rate of the drawing algorithms used at different points in the game, it's possible to reduce overall battery consumption by not wasting processor cycles.

1.4 Why you need to consider the six components in your apps

Lots of apps have been built without knowledge of the six components and have still gone on to be successful. But to rely on luck instead of striving to make your app the best it can be to optimize your chance of success is a fool's errand. Considering each component will lead to improvements in every area of your mobile app.

You won't make a better app by tricking people into prolonged and continued use; that's not the aim. The goal is to create something that keeps people using the app because of what it provides or enables them to do. You'll better meet the needs of the user by first understanding what those needs are. You can help a person make better use of their time by making the app faster and easier to use. You can remove barriers that hinder comprehension and use by adopting common conventions and tailoring the app to the user. And, you can also create real value and grow by word of mouth through a best-in-class user experience. These points are important because of your competition and the expectations of those who are using or will use the app.

1.4.1 Experience is an important differentiator

Competition between apps can be fierce. Depending on the platform (or platforms) you build apps for, you'll be competing in the store with as many as several million other apps that all are vying for attention and installation. And, once installed on a device, you'll be competing with several dozen other apps for use. But it isn't that simplistic. You aren't competing against all other apps in the store or all the apps installed on a specific device. Few people will think, "Hey, I need a new app. I'll go to the store and see what's there."

Retail therapy through apps doesn't work the same way as purchasing physical products. The people who do browse the store looking for new apps for fun are part of a small minority. Similarly, people don't typically take their phone and choose from all the installed apps to find one they want to use at that time, unless they're bored.

Boredom

Boredom is a valid use case for many mobile apps, like games and social media apps that let a person browse through an endless stream of content. Solo journeys on public transportation and the period in the coffee shop while you're waiting for your drink: these are the times for which such boredom-busting apps were built.

Creating an app that people can use when bored or have a few spare minutes and want to pass the time isn't, on its own, an opportunity to compete. Any game that's playable for a few minutes or even a few seconds at a time meets this criterion, and they aren't all competition for your app. If the reason to install an app is that people might use it when they're bored, it's no different from myriad other apps. In turn, this makes it hard to persuade people to install the app.

Also consider the moral and ethical implications for creating something designed for use when bored. While people may use your app when they're bored, you'll make the world a better place by not creating something purely to help people waste time.

People get and use apps for specific reasons. If they want to access a specific service, they'll look for an app that provides that. If they want entertainment, they'll look for an app that provides their preferred types or sources. If they want to stay informed about a certain subject, they'll look to apps that focus on that specific area. In all these cases, it isn't the app that the person desires but the information, functionality, content, or value to which the app gives access.

Alternative apps are only a few clicks away

With millions of apps available, they can't be, and aren't, all completely different. For almost any task imaginable, several apps are available that will let you do some variation of the task. Do you want to edit your photos, record progress of important tasks, track your fitness routine, find somewhere to eat dinner, communicate with friends, meet new people, listen to music, track the latest fashion trends, monitor the status of an upcoming flight, discover new products, see the forecast for tomorrow's weather, or read a book? If so, you'll find many apps providing different ways to do all these things, more than you ever imagined possible.

The ready availability of alternatives means that you can't rely on your app's mere existence for success. With so many options, it's necessary to distinguish what you provide to separate yourself from the competition. Uniqueness isn't enough. To truly compete, your app should have meaningful differences that make it more desirable.

Even if there isn't an alternative or competitor to an app right now, you can't rely on this always being the case. Every day sees the release of thousands of new apps. If you've based your app on a new or novel idea, there's little to stop someone else from coming up with the same idea. If you've spotted an opportunity for an app, others may also see the same opportunity. In fact, the existence of your app in the store may show others who have the same idea that it's valid and serves a need that people want to solve. Furthermore, if you create an entirely new app and it becomes popular, your success will attract competitors who'll look to benefit from your success. You can't stop people from trying alternatives, but you can ensure that the experience of using your app is such that people come back after trying alternatives.

Competition isn't limited to other apps

Many apps are created as alternatives to a website or other computer system, or as an alternative to a manual or paper-based system. Imagine going back to using the website of your social media network of choice rather than a third-party client that doesn't have every feature you require. Or consider remote workers who find it easier to record information on paper and later enter it into a terminal because it's easier than struggling to enter the details on the mobile device they carry with them.

Switching to any of these alternatives may be simpler for some people than changing apps. It's important to be aware of alternatives and ensure that the experience of your app is sufficiently preferable so that people start, and keep, using it.

Experience still matters, even when there's no competition

Two scenarios exist in which there are no alternatives to your app. These aren't an excuse or reason to ignore the experience of the person using the app, and a full consideration of the six components is still important in these scenarios, as it's good for the person using the app and ultimately the business providing the app. The two scenarios without alternatives are

- The official app of a service for which third-party apps aren't available
- Internal enterprise apps (also known as line-of-business apps)

For an app that's tied to a specific service, in the mind of the user, the experience of using the app is directly tied to the service. Consider an online movie-streaming service. No matter how good the backend servers or the range of titles available, if the only way to watch the movies is via a slow, clumsy app, it's an unpleasant experience. The user isn't going to think, "The service is good, but the app is poor." The user is going to think that the whole service is bad. In such a scenario, the concern shouldn't be that a person will start using an alternative app; the concern is that the person stops using and paying for the entire service and they'll go to a competitor.

Within a business, the incentives for a high-quality experience when using an app are different but just as important. For a long time, and in many situations, there's been an argument that the experience of enterprise apps doesn't have to be good; people will use them as they are because they have no choice in the matter. Here are some of the reasons why this shouldn't be the case:

- *Such apps are typically tied to productivity.* If an app's causing someone to do their job slower, it can have a financial impact on the business. On one app I worked on, I made changes to optimize algorithms and use caching such that doing an often-repeated task was now one second faster. The consequence of the time saved when multiplied by the number of times the task was performed and the number of people using the app resulted in a change that was worth the equivalent of around $15,000 per day (approximately $4 million per year) to the business.

- *Apps that are difficult to use frustrate or annoy the user.* It's generally accepted that you don't want to annoy or upset your staff if they'll be representing the business to customers; they may project their negative feelings on the customer or potential customer. I've seen, first hand, the implications of a staff member complaining about the app they must work with in front of a potential client and the cost that had to the business when they didn't win that customer's business.
- *Unduly complex and complicated apps require training.* Not only does this have an initial cost to the business, but it also adds to the cost of onboarding new staff and ensuring temporary and cover staff are able to use the app correctly.
- *Repeatedly having to use a tool that frustrates the user can be bad for morale.* Many studies indicate that a happier staff works harder and is more productive. Giving people tools (apps) that make them less productive doesn't make good business sense. Additionally, studies also show that unhappy workers end up taking more time off work due to illness. Again, this has a financial impact on a business.

Hopefully, these examples demonstrate the importance of creating apps that fully consider the experience of the user, and that it's insufficient to have an app that's merely functional.

1.4.2 Meet the expectations of those who'll use your app

It's rare for any app you build to be the first-ever app that a person uses. People will compare your app with other experiences, including the operating system (OS) and other apps that come with a device.

People are spoiled. They've seen and used a wide variety of high-quality apps and other software. In light of this, any app you create won't just be judged on what it does, or on how it compares to other apps in direct competition with yours, but with every positive experience they've had. You might create an app that's a million times better than your competition, but if it isn't easy to understand in comparison to other apps used on a regular basis, it still won't be seen favorably. Being the best of a bad bunch isn't good enough. You should strive to be one of the best. Period.

People have learned to expect more over time. Go back a relatively short time to the start of the millennium—the majority of software wasn't great. PC software was mostly the same, a series of battleship gray forms. The rise of mobile phone apps over the last few years has changed all that. People want and expect something fast, beautiful, and easy to use. It should help them achieve their desired task and get out of the way. It should improve their life or at least make it simpler or easier, and it must never frustrate or prevent a person from doing what they want. It's also the case that what was once the high-quality exception soon became the norm, and people want better still. Consumer expectations are high and likely only to increase. A desire for continuous improvement exists in people's minds.

These comparisons and level of expectations aren't limited to the consumer app space. The experiences people have outside of work inform their opinions relating to work too. It shouldn't be the case that people bemoan the fact that the apps they use at home or on their own devices are easier to use, faster, better, or in other ways preferable to those provided as part of their jobs. As mentioned earlier, when looking at alternatives, such negative app experiences can be financially bad for business.

1.4.3 Planning for success

The software business is more like all other kinds of businesses than many people wanting to create apps realize. At some point in time, it gained a place in the mind of the population as being a special case. I suspect this was due to the early stories of the success of a few developers. That anybody can have an idea, build an app, and make their fortune overnight is appealing. It's led to the idea that anyone who can write some code can build an app and be a success. Ultimately, this is unrealistic for all but a few "lucky" exceptions.

If you assume that you just need to write some code to be successful, now might be a good time for a reality check. To achieve success without doing the work to help achieve it is like buying a lottery ticket for a retirement plan. There's a tiny chance it may pay off, but that isn't something I've ever heard anyone recommend.

Beware of treating a mobile app differently than other products

The idea that a mobile app is special has thrown up some curious assumptions by those creating them. Compare an app with a website. The technical knowledge and skills to create the two are roughly similar, but you wouldn't hear the developer of a website blaming a search engine if no one comes to a new website they had built but not promoted. Oddly, however, many people creating apps seem justified in blaming the store if few people download an app they've made but not promoted.

Or, imagine a friend told you they wanted to leave their current job in a bank and start a business making and selling loaves of bread. You'd probably have lots of questions for them. "Lots of companies are already making and selling bread. How will you compete with such large, established competitors? Many types of bread exist. Which will you make? How will you persuade people to buy your bread? How and where will you sell it? How much will you charge? Will you be able to make enough money to pay your bills?" And so on. These are all reasonable and sensible questions to ask when starting an endeavor with financial considerations. If you were the friend, hopefully, you'd ask these questions of yourself too.

Now consider building a mobile app. A lot of similarities exist here. They're both low-cost, low-margin products with lots of competition. Both will require making something remarkable or sufficiently different from what already exists to be successful. Both will also require lots of effort to tell people about the new product, and persuade

(continued)

them of the benefits of purchasing. The questions you'd ask your friend are equally valid for both scenarios. Do you consider such questions before creating a mobile app or game? Treat a mobile app like any other product. Expect that making money and being a success with it requires as much effort.

"Failing to plan is planning to fail" is a modern-day proverb attributed to the well-known author on time management, Alan Lakein. It's a phrase that's often used in a variety of situations, and, I believe, it's appropriate to the business of mobile apps. There's also a corollary that is even more relevant, and that's that success which isn't defined in advance is harder to achieve. If you don't define success at the start of a project, when you come to measure it, you're making a judgment call based on what happened. This is the same as waiting to see what happens and then deciding if you want to call it a success or not. If you define what success looks like for you, you can do things that will help you work toward achieving it.

At the start of this chapter, I gave you the formula for app success, but I didn't define success in and of itself. What you consider a success might be different from what someone else would consider a success. That's perfectly fine. What you consider a success today may be different from your idea of it next year and will also likely vary between projects. Whatever your metric for success, you can achieve it in your apps if you provide value, use the six components to create an excellent experience, and do all you can to maximize your chance of luck.

It's time to delve deeper into the individual components. Context lies at the heart of any app experience, so we start there in the next chapter.

Summary

- App success comes from providing value, the experience of obtaining that value, and an element of luck.
- Experience is the most important factor to focus on when striving for success, as it's the one you have the greatest influence over and the one that can more easily set you apart from the competition.
- The six components of an intuitive app experience are context, input, output, responsiveness, connectivity, and resources.
- Context covers all the circumstances of use, and that knowledge lets you tailor the app for where, when, why, how, and who uses it.
- Input is about all the ways that data and instructions get into the app. Covering more than touch, understanding input lets you enable the people using your app to achieve their ultimate goals easily and as quickly as possible.
- Output needs to consider more than a miniaturized version of what's shown on a larger screen. Create visual output specifically for the mobile context in use. This may also include audio, haptic, or digital output.

- The perceived responsiveness of the app matters more than the time it takes to perform a function. Apps should be made to feel as fast as possible.
- Communication requires connectivity. Connectivity isn't universally available and can have different speeds, costs, and reliability. Apps need to tolerate all the different network conditions they'll encounter.
- Portable devices run under constraints that impact resources. The most important resource to manage is power.
- Businesses benefit when the experience of using internal apps is positive.
- Creating a high-quality experience helps you stand out from your competition and meet the expectations of the people using your app.

WEB/MOBILE

Usability Matters

Mobile-first UX for developers and other accidental designers

Matt Lacey

Free eBook
SEE FIRST PAGE

Just because a mobile app works doesn't mean real people are going to like it. Usability matters! Most mobile developers wind up being part-time designers, and mastering a few core principles of mobile UI can make the difference between app and crap.

Usability Matters is a guide for developers wrestling with the subtle art of mobile design. With each expertly presented example, app developer and designer Matt Lacey provides easy-to-implement techniques that instantly boost your design IQ. Skipping highbrow design theory, he addresses topics like gracefully handling network dropouts and creating intuitive data inputs. Read this book and your apps will look better, your users will be happier, and you might even get some high-fives at the next design review.

What's Inside

- Understanding your users
- Optimizing input and output
- Creating fast, responsive experiences
- Coping with poor network conditions
- Managing power and resources

This book is for mobile developers working on native or web-based apps.

Matt Lacey has been creating mobile apps since 2001. He's an independent mobile development consultant, a community leader, and a Microsoft MVP.

> "At last! A systematic treatment of the many different aspects involved in designing an app that people can use intuitively."
>
> —Alan Lenton
> www.IBGames.com

> "An engaging and easy read. It is instantly apparent this book distills years of experience into easy-to-digest chapters."
>
> —Desmond Horsley
> NSW Health Pathology

> "Highly recommended to any developer trying to wrap their head around app usability."
>
> —Clifford Kamppari-Miller
> The Tech Service

> "There were many gems in this book that brought on an 'Oh yeah, that makes sense!' kind of response."
>
> —Alvin Raj, Oracle

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/books/usability-matters

ISBN-13: 978-1-61729-393-1
ISBN-10: 1-61729-393-8
54499
9 781617 293931

MANNING US $ 44.99 | Can $ 59.99