## 1.1 What is UX?

**In our quest to learn about UX, an important place to start is to be clear about what we mean by the term “user experience”.**

**We know that companies like Google, Apple and Facebook invest massively in their user experience design teams. But what is this “user experience” that they’re trying to design?**

Well, if you type “user experience” into Google, and you’ll find this definition on Wikipedia. Which kind of makes it the semi-official definition.

It says “user experience includes all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours and accomplishments that occur before, during and after use of a product, system or service”.

Now that’s a very technical and very academic definition. And what I think they’re trying to say - and what’s a much more precise definition - is that user experience is what it feels like to use a product system or service.

**So user experience is not just about usability. It’s not just about how fast you can get through a process. Or how few mistakes you make when doing so. And it’s not just about functionality. The ability to do A or B or C. How users *feel* when they’re doing A or B or C is just as important.**

What sort of feelings are we talking about? Often they’re subtle, but extremely powerful. Consider these:

* I feel in control
* I feel confident
* I feel smart
* I trust this product

Simple emotions and very powerful when you consider their opposites. I feel out of control or I have no control. I’m doing something important - like buying a house - and it feels like the system or the process or some anonymous account handler has more control than I do.

I don’t feel confident about this system or product or company. I don’t think it’s going to work well or do what I want it to do. I feel stupid. Or I don’t trust this website or app or the company behind it.

These are not the sort of emotions we want our users and our customers to feel when their using our products.

**So UX isn’t just about solving problems as we discussed in the first video. It’s also about generating positive emotions and we’re going to explain this emotional aspect different types of design that go into making high quality products. The first two - functional design and aesthetic design - are generally understood. But the third - experience design - isn’t as familiar and often gets overlooked.**

Let’s take a look at each one.

Functional design determines what a product is built to do. It defines the engineering that gives a product its capabilities. So with a car, for example, things like horse power, engine size, the type of transmission, how fast it can go from 0 to 60 will determine what sort of car it is - whether its a sports car, a family car or an SUV. Each of these types of car would have a different functional design underpinning it.

The second type of design is aesthetic design. How does the product look? How visually appealing is it? What sort of personality does it have? What do its looks say about the product’s brand? And you can imagine that the sports car, the family car and the SUV would each have a different aesthetic design corresponding to the type of car they are.

The third category of design, the one that’s less familiar to us all, is experience design. What does it feel like to drive this car. What does it feel like to sit in the driver’s seat, hands on the wheel, as you drive this car along the road.

How responsive is the steering wheel? How smooth is the gear stick. How easy is it to adjust the controls on the dash.

What sort of sounds does the car make? What sound does the engine make - for example - when it’s idling in traffic? Or cruising on a motorway? Or overtaking on a country lane?

What’s it like to do mundane things like stick your golf clubs in the boot? Strap the kids into the back seat? Or adjust the position of the driver's seat?

**Consider this important point. All these small details on their own might seem trivial or mundane. But when added together, these details determine what it *feels* *like* to use this car. And companies that are serious about creating great products don’t leave any of these details to chance.**

**Because - here’s another important point - experience design doesn’t happen by accident. It’s a deliberate, intentional focus on the small details that matter to customers.**

Mercedes, for example, doesn’t the workers on the factory floor determine the smoothness of the gear stick. Or the sound of the engine. Or even the feel of the buttons on the dash. Of course not.

And it’s not that their workers aren’t capable. It’s just that Mercedes knows that these details create a massive emotional reaction with customers. And they want that emotional reaction to be a positive one. So these details are too important to be left to chance. And they are defined long before the car hits the factory floor.

**We talked about the three types of design in a sequence: functional design, aesthetic design and experience design. But that’s not how it works in companies like Mercedes. You don’t leave experience design until last.**

All these small little details are defined at the same time as the functional and aesthetic design. The three types of design need to be working together in harmony. And we’ll see an example of that in another video when we look at the design process at BMW.

Finally, the details aren’t there for fun. Each detail has been carefully crafted to solve a genuine problem for users, be it air conditioning, entertainment, navigation or pacifying kids.

Before we move on to the next video it’s worth recapping what we’ve learned about UX. If you’re ever asked in an interview “What is UX”? A smart answer is to say that it’s a problem solving discipline. Identifying user problems and building software to help solve them

On top of that, you can also say that great products solve user problems *and* generating positive emotions while doing so.

**In the next video we’ll look at another example of experience design, and how it shows that functional and aesthetic design, on their own, aren’t enough to create a great product.**

# 

## 1.2 Why experience matters

**Let’s look at another example of the three types of design. This time from the smartphone industry.**

These are some of the many functional design drawings of the technology underpinning the Blackberry 10 smartphone. This was the first Blackberry touch screen smartphone released using Blackberry’s new touch-ready operating system. It came to market in early 2013.

Now, Blackberry had a great track record when it came to technology - the Blackberry messaging service, getting email on your phone, corporate-level security - these were all innovations that Blackberry brought to the handset market. So let’s just assume for argument's sake that they kept up this track record when it came to the functional design of this new phone.

Aesthetically, the phone looks the part. Here are some screenshots of the key screens of the phone, and it looks as good as the other operating systems did in 2013. We’re talking about iOS 6 - the ugly one before the new flat design was released in iOS 7. And long before Google’s Material Design was implemented in Android.

Consider the design of the Blackberry Hub, shown here. And the time this was a new way of presenting a user’s incoming communications: emails, text messages, tweets and so on - all displayed in one stream. You’re probably familiar with the concept, as it has since been adopted by both iOS and Android, but seen first on Blackberries.

So again, it's safe to assume that when it comes to aesthetic design, the Blackberry 10 is doing the business. As far as experience design goes, let’s watch a video of five people using the Blackberry 10 for the very first time.

*[Mashable video here]*

Not surprisingly - looking at that video - the phone was a flop. A few months after the release of the phone, Blackberry had so much unsold stock of the Z10, that it was technically bankrupt and put itself up for sale.

The phone that was supposed to rescue the company had failed badly. And a company that once dominated the handset market, was soon to exit the market altogether.

Why did the phone fail? Why was it so difficult to use? Will explain that in more detail later. The short answer is that the phone did something radically different than existing phones: it had no home button. And the alternative they developed - a series of swipes and gestures - wasn’t sufficiently well designed. Users were confused doing the most fundamental task: navigation…

**But more about that later. The key point here is that functional design and aesthetic design, on their own, aren’t enough. If you don’t get the experience design right, those other two things won’t even matter.**

**Think about the people we saw in that video, who were generally light hearted and laughed their way through the experience. And the reason they were light hearted was because they didn’t pay for those phones. If you had spent several hundred dollars buying a Blackberry 10 - or you were locked into one on a 2 year contract - and you brought the phone home and couldn’t turn it on, I reckon your emotions wouldn’t be light hearted. They would probably range from frustration to anger to regret. And those aren’t the sort of feelings we want people to have when they use our product.**

**Because when people are feeling this way, there isn’t any room in the brains to admire our functional or aesthetic design. Their negative emotions will override everything else. And all the hard work spent on our clever functional design and beautiful aesthetic design will be wasted.**

**Let’s consider the key points from this example:**

* Great products need all three types of design. If your team or organisation is relying on functional and aesthetic design alone, you are being exposed to the same risk of failure as the Blackberry 10. The two types of design on their own are not enough
* Negative emotions have a multiplier effect. In other words one negative experience can outweigh many positive ones. Because people don’t notice good design as much as they get frustrated by bad design. Without a dedicated focus on experience design, the risk of negative emotions outweighing positive ones gets higher

## 

## 1.3 UX is a state of mind

So far we’ve looked at two examples from organizations with large budgets. One who uses its money wisely to make excellent products. And the other who - despite plenty of resources - still managed to get it wrong.

**The final example we’ll use involves an organisation with practically no budget. And I want to use this example to emphasize a key point. UX isn’t about money. Having the right state of mind is much more important.**

This is the exterior of a small guesthouse or B&B on the Dingle Peninsula on the west coast of Ireland. Outside of Dublin, the Dingle Peninsula is probably the most popular tourist destination in the country. So if you run a B&B, it’s a competitive market.

From a functional design perspective, it couldn’t be more functional. It’s just a big house with lots of bedrooms in it for guests.

Aesthetically, it’s definitely got charm. The bedrooms look really nice, and the owners are obviously making an effort. But there’s nothing in this room that you probably couldn’t buy in IKEA. So it’s nice, but it’s not the Ritz.

But when it comes to experience design, the owners are pulling out all the stops. There are gourmet yogurts and Bloody Marys for breakfast. Beautiful flowers in the garden that maximise the sea views. When you order herbal tea, you get real herbs in the pot. And not alone do you get free range eggs for breakfast. You get free range chickens running around the front lawn, which is manicured like centre court in Wimbledon.

And customers notice these details. This B&B has been ranked number one in its market for several years. Nearly all the reviews refer back to these small details. Two examples here mention:

* The great food and beautiful rooms
* The bed is so comfortable
* The linen smells so sweet
* They turn the beds down at night
* All the touches were provided

The owner of this B&B doesn’t have a big budget or a large team. He’s not part of a chain or a franchise. And although I’m sure he’s never heard the term experience designer, that’s actually what he is.

He’s got empathy for his customers. He observes them closely to understand what really matters to them. And he’s constantly tweaking and improving and experimenting with his product to get the details right.

**And here’s the critical point. His B&B isn’t number one in the market because of its bland functional design. And it’s not number one in the market because of its IKEA-inspired interior design. It’s number one because of all the small details added together that create a knock-out experience that customers love, and vote about, and talk about and spread the word about.**

Now, at this point you might be asking So What? I’ve been watching several videos on this UX course and none of them are about software. Why do I care about phones or cars or B&Bs.

Well, let me explain. Consider these companies. Not all of them are in the software industry per se. Some are in the travel industry or news or retail or automotive. But all of them realise that software is now critical to their success. And they invest massively in UX design.

Why? Because in a competitive/commoditized industry like airline travel- for example - one of the few ways to differentiate yourself from the competition is through great experience design. Ryanair, the European budget airline realised this recently.

**And all our customers are using products designed and developed by companies like Google and Facebook and Apple. And these companies are setting the bar very high when it comes to what people consider to be good software. So if we’re not spending time considering these little details and understanding what details matter to our customers and understanding how we can design them, then our software is going to feel second-rate in comparison.**

# 

## 1.4 Product integrity

**This entire course is structured to explain what an experience designer does. And we’ll be spending a lot of time looking at the tasks and techniques that are the experience designer’s responsibility.**

**But at a very high level, there are two overarching objectives for the UX designer. And these are to do with product integrity and product desirability. Let’s look at product integrity first.**

In any organisation the obstacles to creating a great user experience generally don’t come from the users. The obstacles come from within the organisation that’s trying to create the experience itself.

You might recognise some of these stakeholders and some of these roles. And what typically happens on a project, is that these stakeholders are pulling and tugging at the product, feeding their requirements into the product’s design. YES this feature should be in the product, or NO this feature should not be in the product.

And that’s fine, that’s their job. They’re trying to realise some of their objectives through the release of the product.

**But the problem is none of these stakeholders typically have any product design skills. They’re not thinking about the small little details that go into creating a great experience for the user. Often they're not thinking about the user at all. Instead they’re just thinking of their own narrow piece of the puzzle.**

And what’s missing from this picture are two things. Somebody representing the user. And somebody representing the product.

And the job of the experience designer is not just to represent the user when all these conversations are taking place and these requirements are being gathered… but also to represent the integrity of the product itself.

I’ll explain what I mean by product integrity using this app as an example. If you live in the UK and Ireland you might be familiar with Hailo, a taxi app, very similar to Uber, and certainly a very popular app in Dublin.

Here’s how it works. Once you launch the app, it zooms to your current location, showing you nearby taxis. Press “Pick me up here” and then “Hailo now” and, hey presto, it starts looking for a cab. And in a few seconds I have a driver matched who will pick me up in 12 minutes.

What people love about the app is the immediacy and simplicity. It’s right here, right now, with just two presses a button - no messing around.

But let’s consider all the things this popular app doesn’t do.

* You can’t order a taxi for some other time. You can only order a taxi right now. So you can’t pre-order a taxi for that early morning trip to the airport for example
* You also can’t order a taxi for somewhere else. So you can’t send a taxi to pick your husband or girlfriend or parents up from the station, for example
* And finally, you can only order one single taxi. So you can’t order multiple taxis to take your team out to dinner after work, let’s say

You might think that these are serious weaknesses in a taxi finding app, but it’s precisely because it doesn’t do these things that the app is so great.

You can imagine if the app was made and developed by a larger organisation - let’s say by the organisation that operates taxis in Dublin - that various stakeholders within the organisation would be demanding that these additional features be included. And look what would happen.

* Well users have to be able to tell us what time they want us to pick them up
* They might want more than one taxi so we have to give them this option
* They might not actually be at the address, so they have to be able to tell us where they want to be picked up at
* And now they're ready to place the order…. once they’ve agreed to our terms and conditions of course

So these requirements, while well-meaning, are counter-productive for everybody. They make the app less appealing and more difficult to use. Meaning less usage and less fares for the drivers. A lose-lose situation.

**And that’s what happens in so many organisations, where unstructured requirements gathering takes place without a strong voice representing the product. Yes, some of these features might seem valid, but if they damage the product and the business, are they really worth it?**

**And that’s the role and responsibility of the experience designer; to represent the product and to keep sight of what’s best for the product, the business and the user.**

Before we move on to the next video, let’s clarify what gives a product integrity:

* The product is coherent - it’s clear what it does and what it is about
* The product has a singular purpose - it’s not trying to be all things to all people
* The product is focused on solving user problems - it doesn’t become a vehicle primarily focused on solving stakeholder goals

# 

# 

## 1.5 Product desirability

**The second high-level responsibility of the experience designer, is to focus on desirability. I’ll explain this concept by talking about three key ingredients of any successful product: viability, feasibility, desirability.**

Viability concerns money. Any product has to make money or save money for the company that’s creating it. And the amount of money it makes or saves has to be less than the cost of building it. There might be other reasons for building a product, but often it comes to money.

Feasibility refers to technology. It’s got to be build-able. If not by us, then by somebody we hire to build it. And it’s got to be built at a price that makes it viable, and in a timeframe that’s reasonable.

Desirability means that somebody has to need or want this product. In other words, it has to be solving a problem for somebody - or else it shouldn't exist.

And if it is solving a problem, and people want to use it, then when they do come to use it, it’s got to create a positive feeling that makes them want to use it again.

Many companies focus on the first two variables - feasibility and viability - because they lack the skills and knowledge to look at the third. That's why you see so many products that are viable and feasible, but that nobody needs to use, or wants to use, or enjoys using.

A great example of this is the jukebox 6000 by Archos. This was one of the very first portable MP3 players. Released in December 2000, about 10 months before the first iPod.

From a viability and feasibility perspective, things couldn’t have been better for Archos. They had figured out a way to mass produce this product, at a cost that was lower than the price they sold them for, just at the moment in time when this market, for portable MP3 players, was about to take off.

Absolutely perfect, you would think. Except… when it comes to desirability.

If you remember the first iPod, there were so many things about it that seemed new and different to other consumer electronics products at the time. And these were the things that people loved:

* It was small
* It was minimalist
* People thought it was beautiful
* It was easy to use
* And that ease of use was exemplified by the scroll wheel, that allowed you to scroll through hundreds or thousands of songs in a matter of seconds

And the Jukebox 6000 didn’t stack up on all these factors.

* It wasn’t beautiful or minimalist
* It was relatively big and chunky
* And it was a little heavy

When it came to transferring music from your PC to the device, it was slow. Slower than the iPod.

Also the 6000 wasn’t plug and play, you couldn’t just stick it in to your computer and get started. Instead you had to go to the Archos website, and download some drivers, install them on your PC, and that was never a great experience

And it didn't have a scroll wheel or anything like it. So to select song 385th on your list, you had to tab through your collection 385 times.

**So despite the 10 months head start they had over Apple, despite being feasible and viable, the 6000 wasn’t desirable.**

**And that opened up a gap for a product that solved the same basic problem - portable mp3 playing - but which also delivered a fantastic experience while doing so.**

**And it illustrates the point, that feasibility and viability on their own aren’t enough to ensure success. Your product needs to have desirability as well.**

As an experienced designer it's up to you to ensure that your product is desirable.

Think of it as three concentric circles. The first priority is to identify whether there is an actual problem that your product can help solve - does anybody need it? This is the most fundamental requirement.

If there is a problem, then we have to ask if our product is actually solving the problem. Another basic requirement.

And if it is solving the problem, we can’t stop there. We have to make sure the product isn’t solving it in a mere functional way like the Archos. It has to be solving the problem in a way that creates a great experience while doing so. That’s desirability.

[Colman - possibly introduce the lovable triangle here]

## 1.6 Making a business case

**Throughout your UX career, it’s highly probable that you will spend most of your time working in a business environment. That is, working for an organisation where profit and loss, the bottom line, is the key business driver.**

**And most people working in a business environment don’t have much interest in UX. They don’t care so much about user research, different types of design, personas, prototypes, wireframes and all the rest.**

Instead, they’re much more interested in the OUTCOMES of UX. What can UX do for them? For their team? Their department? The company? The bottom line?

Being a really effective UX designer isn’t just about being good at design. It’s about being good at communicating the benefits of design. The business benefits.

To do that you must distinguish between the inputs of UX; and the outputs or outcomes.

You need to be an expert at the inputs: research, analysis, design and prototyping. But you have to understand that your colleagues mostly only care about the outputs. And those outputs can be summarised in one word: money

**Before we continue, let’s step back and be clear about the objectives for this lesson. We want you to be able to build a business case for your UX projects. We want you to be able to communicate the business benefits in terms your stakeholders or clients will understand. And we want you to be familiar with some of the most common business metrics.**

First let’s talk about business benefits. Ultimately there are only two benefits to a business:

* Increase revenue
* Reduce costs

There might be other ways of talking about them, but really it all boils down to whether the activity the business is undertaking will make money or save money.

When we elaborate a little more, we see terms like these being used:

* Increase revenue
* Reduce cost
* Increase customer acquisition: customers being a source of future revenue
* Increase conversions: getting more people to do what we want them to do - register, buy, contact us, and so on
* Increase customer satisfaction: which will lead to more repeat business and revenue
* INcrease customer retention: keeping customers for longer = more revenue
* Reduce customer churn: churn is another word for losing customers
* Reduce time to market: getting our products into the market sooner so we can sell sooner and - of course - increase revenue

If you work for an organization that sells products or services online, such as an airline, then your projects could potentially help with the following:

* Increasing sales
* Increasing sales / visitor
* Increasing revenue / sales

**Let’s look at a few case studies where UX projects had a positive impact on the bottom line of businesses.**

The first case study we’ll look at is for Supervalu, the most popular supermarket chain in Ireland. Like many retailers it faces fierce competition. In this case the competition comes from other irish brands, as well big UK chains, and the German discounters.

The project at hand was to redesign Supervalu's online shopping, the new frontier in this competitive landscape.

But it wasn’t a redesign for the sake of it. It had to deliver real, tangible business outcomes. And by tangible I mean, outcomes that had an impact on the bottom line, that involve money.

**The bulk of the work was carried out by research agency Bricolage, a little bit of help from ourselves and UX consultancy Each & other. They followed the process, using research to identify problems that could be fixed, designing simple but effective solutions, such as decluttering the interface, and reducing the number of steps at checkout, and then working with the development team to make sure everything was implemented correctly.**

The design outcome was a beautiful new, streamlined shopping experience for users. But the business outcomes are what made it a success:

* 51% increase in online sales: that’s revenue
* 84% increase in registrations: that’s more customers who will bring in more revenue
* 620% in mobile mobile shopping: and this is future revenue. If more shopping is going to be done on mobile in the future, this statistic is ensuring that the shopping platform is ready for that shift

The second case study also involves an irish business, Tesco Mobile, which is a small subsidiary of the giant British supermarket chain.

Tesco Mobile is a small brand in another highly competitive sector, mobile phone networks, dominated by two giants, Vodafone and Three.

The executive at Tesco Mobile wanted to redesign their online shopping platform to make it a more effective sales channel.

Again, by following the process, and particularly doing a lot of user research to identify the problems with the existing platform, the UX agency Fathom was able to deliver some incredible results, including:

* 9.5% increase in customers
* 17.7% increase in revenue
* 170% increase in order value: that is people were spending much more money each time they bought something

These are the types of results that will help a business audience understand the benefits of UX.

Our final case study takes a different viewpoint, focusing on cost reduction. The organization is Mozilla Corporation, a large not-for-profit organisation that makes Firefox web browser amongst other tools.

It’s a big company with half a billion dollars in revenue and over 1000 employees.

Every year millions of users seek support on the company's web forums. And the Mozilla team tried to answer them every day.

Customer support is a cost for every company. It’s easy to measure the cost and it’s easy to identify the top issues that customers have when calling or emailing customer support. It’s always a good place to start when building a business case for UX work.

In this example NNG worked with Mozilla to cut the number of questions being asked on the forums by a massive 70%. From 7000 questions per month to 2000 per month.

The numbers of questions answered within 24 hours increased from 40-60% to 80-90% which means they were delivering a vastly improved service for customers.

For many organisations these results would translate into reduced costs. Less money is being spent on customer support.

**When building a business case what you’re trying to do is to convince your organisation that the money spent on the UX project will ultimately increase revenue or reduce costs by the same amount or higher.**

**So you need to do your homework. Estimate how much will be spent on the project. And estimate how much additional revenue or reduced costs the project will accrue to the business.**

**Be realistic. And make sure you have some evidence for your claims.**

For example, a project that aims to reduce the number of steps on the checkout process. If you can point to statistics that say 20% of customers drop out of the sales funnel during checkout. And that this translates to a loss of 500k. And that your project will hope to reduce the dropout by 20% and increase revenue by approximately 100k. That’s a solid business case assuming the costs of designing and developing the new checkout are less than 100k.

**Your business case doesn’t have to be a large document. A short, well-reasoned case for your work, with clear business benefits is all you need.**

# 

## 1.7 UX is a process

**Process isn’t the most exciting word, but UX is nothing if not a process.**

**And companies that are serious about design, like Apple or Mercedes, are also serious about following the design process.**

And, at a high level, this is it. The classic design process that’s used whether people are designing a car or a phone or a chair or a house.

And it couldn’t be more simple or logical.

Working backwards you can see how that logic unfolds. Before you release a product to your customers you need to test it. Does work, is it safe, will it explode when anybody turns it on.

Before you test it, you have to build it. That’s self-evident.

Before you build it, if you’re smart you will design it. To make sure everybody on the team know what it is we’re building.

And before you build it, you’ll do some research to understand, what is it we want to build, and why.

This is fundamentally it. But in reality it’s slightly more refined. Let’s take a closer look at process. We start with research where we engage with our customers or end users to understand the problems we’re trying to solve for them.

When you conduct research you’re gathering data and you need to spend time analyzing that data to clearly articulate the precise problem or set of problems we’re trying to solve.

Design is where we solve the problem. That is; we design a solution that we think is going to work for our users. And part of the design process is to create a prototype, a mockup of our solution that we can test with users to validate that it does actually solve the problem, before we go to the expensive business of building it.

This is an iterative, circular process that may take a few cycles before we’re happy that our design really work. And only at this point are we ready to build and test and ship the product.

For the moment, let’s go back to the simplified version of the process. And let’s use a real-world example, and see how the design process might have played out during the creation of this building. This is a school in the midlands in Ireland, an award-winning school, that looks nothing like the school that I went to as a kid.

Working backwards you can imagine the design process went something like this:

Before they let the kids into the school, everything was tested rigorously. The fire alarms, sprinklers, Bunsen burners in the science lab. Just to make sure that everything was working properly and safe.

Before they conducting the testing, the school was built. Completely built. There was no scaffolding, or blocks, or workman’s tools lying about when the kids entered the school. Absolutely everything was finished.

And before any construction started, before they started digging the foundations, the school was designed.

And the design would have been incredibly detailed. It’s not enough to say that we want light on the ceiling, you need to specify exactly where they go and what type of light fittings you want; the exact position of the windows and the types of material for the window frames, and so on.

And…. before the school was designed you can imagine that the architects had some fundamental research to do. First, what sort of building do you want us to design? Is it going to be school, or a hospital, or a hotel? And if is a school. What sort of school is it going to be?

**Is it a primary school, or a secondary school, are both? Is it a boys school, or a girls school, or both? Will there be 100 pupils or 1000 pupils? Is it going to be sporty or is it going to be academic? Is it in an urban location, where everybody walks? Or is it in a rural location, where everybody drives? All these questions would need to be answered before the architect puts pen to paper and starts the design.**

And that’s the design process. But it's not all of it.

Because there are even more fundamental questions that need to be answered if you’re going to create a truly great product.

And whoever was responsible for this school — the principal or the board of management — tried very hard to answer these questions.

* What sort of school to we want to create?
* What sort of ethos are we trying to impart form the design of this school?
* How do we want the kids to feel when they come into school every morning.

This is a vision. And it’s the role of the experience designer to help define this vision.

And to create a truly great product you need to spend time up front, before any designing or building takes place, defining what sort of product you want to create. Which is why the research phase - or the discovery phase, or whatever you want to call it, is probably the most important phase of the process.

What’s magical about following this process is not just the output, but several other fringe benefits that make the product development process so much smoother.

First the vision is clear and the product is visualised in high fidelity from the beginning. Everybody knows what they’re building. Everybody knows that the product is going to look like when they’re done.

**And this is particularly important in software, because software is invisible until it’s finished. And it’s too easy for everybody working in a software project to have a different idea in their heads as to what is actually being built.**

The process also has a natural structure around which all the stakeholders and actors in a project can coalesce. This process works for everybody, not just UX designers. In my experience, project managers, developers, testers and and all the other participants always prefer structured projects with clearly defined vision, objectives and designs. It makes everybody’s job easier.

Ideas can be iterated cheaply during the design phase. It’s much cheaper to change your mind when your sketching ideas on paper, than after construction has begun.

And finally, the three success factors: desirability, viability and feasibility, given equal importance upfront. It’s not a case of trying to slap on a bit of desirability at the end.

We have added content here about the business case for UX.

## 1.8 Agile and UX

This totally linear design process has been modified slightly to fit into the not-so-linear agile software development world.

**Some agile practitioners would claim that there is no place, and there is no time, for design in agile. In fact, you don’t need design if you’re truly agile. All you have to do is get your working code in front of users and modify it based on the feedback you receive.**

**While this sounds attractive – the fact that we can save time by not doing design – in reality it’s a false economy. The only thing you learn from putting badly designed software in front of users, is that people don’t like badly designed software.**

But it is a challenge to fit what is often a three or four month design life-cycle into a two or three week sprint. So let’s take a look at how it’s done.

First of all in an ideal world the linear design process wouldn’t actually be linear. In an ideal world we’d be in a continuous improvement cycle.

Once we’ve designed, built and tested our product, and released it to customers, we’re in a position to carry out research on the live product, and start designing improvements and repeating the process over again.

This is the ideal world of continuous deployment. And if you have a software as a service product, or a popular app in the app store, you probably live in this world.

In the classic waterfall project management process, those steps can be played out over a 12 month period, depending on the scale of the product that was being built.

But if you think about it, this is quite a risky approach. Anything could happen in those 12 months. You could run out of money during the design phase. And all the work completed up to that point would be thrown away.

You get a new boss while your working on your prototype. And he or she decides to change course, and cancels the project. All your work thrown away.

Competitors could release a new product shortly before you go live, changing your strategy. And rendering all your work up to that point, null and void.

Anything could happen over 12 months, and waiting that long to release any code into the wild could potentially see a lot of time and effort and money wasted.

One of the tenets of agile is: wouldn’t it be better if we released software more frequently and reduced this risk. Rather than waiting 12 months for one release how about we have a release every three months. And if you are being very rapid and are working on two-week sprint cycles how about four releases in two months.

And this is where people began to wonder, how or where does UX fit in to these very tight short development cycles.

And the answer came from these guys, IDEO, a product design company based in San Francisco, and Google Ventures, Google’s venture capital arm. These firms use a rapid fire methodology that allows them to work with their clients to identify problems, solve them, and build product prototypes, all within a short period of time, such as a week.

And their approach was adapted by Jeff Gothelf and Josh Seiden, in their book Lean UX. To summarise very briefly, what they are saying is that you still follow the design process but you just squeeze all the steps into a much shorter, more compact timeline.

And instead of trying to solve the big problem upfront, take the agile approach and start solving elements of the big problem, smaller problems essentially, piece by piece.

**~~Later in this course will take a more detailed look at how you can compress user experience design into shorter time periods. That module will be called running a design sprint. But for now it’s enough to know that the process is the process and it doesn’t change - despite what some people might say - even when it's being compressed to work in an agile development environment.~~**

# 

## 1.9 Humans are not machines

**There are things that are inherent about technology that make it complicated. And then there are things that we humans do that make it even more so. And we are going to look at both of these factors.**

**And there are good reasons for us to discuss this. If technology wasn’t complicated, there would have been no need for the user experience industry to appear and help simplify it. UX exists to solve a problem and it’s important and useful to us to understand what that problem is.**

A lot of material in this module is taken from two highly influential books in the UX space. The first is “The Design of Everyday Things” by Don Norman, first published in 1988.

The second is “The Inmates Are Running the Asylum” by Alan Cooper from 1998.

These books really set the template for what has become the UX profession. And we will touch on some of what they cover in this course, but we’ll only be scratching the surface. So I’d strongly recommend you have both these books in your UX library.

Back to the topic of technology being complicated, the first factor is that humans and computers are completely different beasts, as summarised in this table from Alan Coopers book. My laptop, for example, can remember 1 million phone numbers without making any mistakes.

But it doesn’t recognise my face. And I’m the only person it knows and I use it everyday.

Me, on the other hand, I can remember about three telephone numbers, on a good day. But I could recognise somebody on the street that I haven’t seen in 20 years.

Computers are incredibly fast, they’re error free – it’s the software we put into them that typically makes the errors – they are totally logical, and totally predictable.

Humans, in comparison, are incredibly slow, we make mistakes all the time, we’re not logical at all – we’re irrational, emotional, and unpredictable.

**And a lot of friction you see when you see people frustrated with computers, is this clash of the human and the machine. The human meeting the machine, and the machine is programmed to expect the human to behave like another machine - to be totally logical, not make any mistakes and to be really fast which, of course, we’re not.**

**And it's our job, the experienced designer’s job, to make the machine behave better. To be polite, to be nicer, to expect us to make mistakes, to expect us to be slow, and not to punish us for these human qualities. To be more considerate and understanding instead.**

## 1.10 The danger of features

**Another reason technology is complicated is because of features. While this may seem counterintuitive – surely features are a good thing – it’s actually one of the most important concepts for a UX designer to grasp. Features are not always a good thing.**

**Features and technology go hand-in-hand. What technology actually allows us to do, is to add more features to our products.**

We’ll examine this by looking at telephones. Some of you might be old enough to recognise this type of phone – known as a rotary phone in the United States.

This phone has only two features. You can make a call by lifting the handle and dialling the numbers or you can receive a call by lifting the handle and starting a conversation.

Because of the limited number of features, there are only two controls on the phone: the handset and the rotary dialler.

Because there are only two controls, the phone has a really simple uncluttered interface. Simple interfaces are more intuitive, and much easier to understand.

So simple in fact, that my three-year-old daughter knows how this phone works, even though she’d never seen her parents use a rotary phone before she came across this one at an exhibition.

Fast forward to the 1980s. We have new technology – integrated circuits, also known as microchips. These have allowed us to add new features to the phone. And these features are activated by these buttons here. So we have more features requiring additional controls, and now the interface is more crowded. And because of this, it has become less intuitive.

But hold on a second I might hear you say. Is it really less intuitive? It looks simple enough to me. Well this is an experiment that we conduct in our public classes. Hands up everybody who knows what the MR button does? People have a variety of answers.

“it dials the last number”. Are you sure? No.

“it’s memory recall”. What does that mean? I’m not sure.

Of over 800 people who came to our courses last year, nobody was able to confidently say what the MR button does.

**So straight away we have moved from a product that even a three-year-old can use, to a product where people are uncertain about what exactly it can do.**

What the MR button actually lets you do is save numbers to the phone. You can save up to 10 telephone numbers to the phone to save you the trouble of tapping the buttons everytime you want to call your mother for example. It was an early version of speed dial. So, if I want to save my wife's telephone number to the phone, what do I do? Again, this is one of the experiments we conduct in our classes and not one of our students has ever been able to get it right. Because of the convoluted process involved:

To save a number to the telephone, you need to do this:

* first you need to lift the handset
* then press the MR button
* then press the number of where you want to store it in the sequence of 1 to 10
* So if I wanted to save her as number 2 on the sequence, I would press button 2
* then you enter the telephone number
* then you replace the handset

Then to call my wife, what they need to do is lift the handset, press the MR button, and then press the number two.

It’s so complex people ended up writing down the names of the numbers they had saved and in what sequence. This defeats the whole purpose of the feature.

And there are two really important points here.

**First, features involve a trade-off. Each new feature crowd out the existing features. Each additional feature adds complexity, makes the interface more cluttered, and places an additional mental burden on your users, however slight.**

The second point is that just because you can add a feature, it doesn't mean that you should. Did this feature really add value to users of the telephone? Did anybody ever use it? Or did it just add clutter to the interface, and sow little seeds of doubt in people’s minds every time they use the phone.

If we fast forward to the present day, we see phones like this. A Nokia feature phone. The amount of technology and features in this phone is on a completely different scale compared to the previous phone. Let’s just list out some of the things this phone can do:

* it can make phone calls
* it does voicemail
* it does SMS, email, and you can get on the Internet
* it’s got a camera, calculator, radio, MP3 player, and Calendar
* it’s got a clock, alarm clock, and a stopwatch
* it’s got apps?? like YouTube and Facebook
* and you can configure it and customise it to your own preferences, such as adding wallpaper and setting favourite contacts.

On one hand, you have a remarkably complex device from a technology perspective. Yet, it’s not complex from a users perspective. This is actually the phone my parents use, because they think it’s actually easier to use than a smartphone.

**So how come the phone with only a small number of features was so complicated? And this phone with a huge number of features is relatively easy to use? Well, that difference is called “design”.**

The Nokia is really well designed. And even though Nokia missed the smartphone bus, they have a great track record at designing easy-to-use feature phones.

And this brings us to an additional point about features. They must be designed. You can’t just throw in a feature to your product and hope for the best. That’s what they did with the MR button and it didn’t work out so well.

**Let’s look at the consequences of ignoring all three of these points, using remote controls as an example.**

This is the very first wireless TV remote control designed by a company called Zenith in the 1950s. The remote had a limited number of features, such as channel up channel down, volume up volume down – and only had four controls – four buttons – to activate these features.

Fast forward to the present day, and this is a remote control for a DVD player. It has 43 buttons which are used to activate and control the many features of the DVD.

The cost of including these manifold features is that the really important features get crowded out. Like the play, rewind and forward buttons.

It’s a classic example of a product succumbing to the “feature arms race”. Let’s try and beat the competition by adding more and more features, regardless of whether anybody needs or

uses them.

Some of the features on this device include depth enhancer, multi-remaster. And advanced disc review. And if you’ve never heard of them, you’re in good company. Neither have we and neither have the hundreds of people attending our training courses each year.

**This arms race breeds a form of insanity. Adding features that nobody needs, nobody uses or even understands.**

Finally, the interface is not designed. At least not intentionally. The buttons are just thrown together. With no prioritisation, organisation or obvious logic.

**It’s like getting all the ingredients for a cake and pouring them out on your kitchen table, and claiming “this is a cake”. Of course, it’s not a cake, it’s just a mess. And so is the interface for this remote control.**

There’s a lot to learn from this example. So let’s quickly recap the key points:

First, features add complexity. This is a stone cold fact. The more features you add, the more controls you need to activate them. More controls make interfaces more complex, this is unavoidable. You need to decide whether your new feature is worth the complexity it is bringin to your product.

Second, features must be designed, which adds time and expense to your product development efforts. You also need to decide whether your feature is worth the time and money designing, building testing, supporting and maintaining it.

Third, just because you can add a feature, doesn’t mean that you should. Features must serve a purpose. They must solve a genuine problem for end users. Otherwise it is a lose-lose-lose situation: more complexity, more cost, no added-value.

Finally, here’s a handy checklist of questions to ask before deciding to add a feature to a product.

First of all, does anybody actually need it? And by anybody I mean an actual user, not a stakeholder in the business. What is the problem its solving? And Who is it solving the problem for? Can you prove that people need it? it shouldn’t just be random guesswork.

Second, what is the trade-off? What additional complexity are we adding to the product? What impact will it have on the core features and user experience?

Finally, what is the cost - in time and resources - of adding this feature? And is it worth it?

Answer these questions before adding features, and you’ll be making better more judicuous decisions.

# 