

Great, so we've been looking at user centered design.

- And we've been trying to think about how we use those approaches to understand the way that websites should be laid out.
- And we're primarily doing that so that we understand how to work with clients.
- And how to understand products, so we can deliver products in a way, on the web, that makes sense with the approaches that the client is requesting.
- And also understand what it is that users want to do.
- And then also use some examples to define what you can do, and perhaps what you shouldn't do, when you're working with a client or perhaps on your own with your own business, to put a website together.

So we'll start with some basic guidelines and these guidelines are roughly drawn up from two main sources. And I think that, to be clear, they are not exhaustive, they're not designed to be exhaustive. They're just to give you something to refer back to when you're starting getting going with this process. And when you're working with other people to do work like this. So we're gonna take a look at some samples of good design practice and bad design practice. And we're gonna be constructively critical about them, in order to find out what we might avoid doing, as well as what we might want to do. And we're gonna try and define useful guidelines, just for getting things done quickly. And as I was saying, this is not a lesson in design, it's not a design MOOC. This is just some guidelines to help you understand and work with designers and finalize designers. It's not a lesson in illustrations or graphics, it's just guidelines, basic rules of thumb, which you can apply in your projects. And it's not a full-blown lesson in user centered approach. It's either just a summary to get your head around what you might do.

- And we just learning to be critical about websites, so we can understand and avoid common mistakes. And look at examples of good practice, and see why they work. So as I've said, there's gonna be two main approaches that we're drawing on.

## 2.1.5 Different approaches

- We will mainly draw from two basic approaches
  - Cognition / Evidence-based principles ( e.g. Edward Tufte)
    - What is the **information**? How do we make the **information** clear?
  - User-centric approaches
    - Who are our users? What do they want? How do they get the information?

And the first one we haven't really talked about yet, and Kate hasn't mentioned it either. This is the cognition and evidence based principles,. And this is slightly different to some of the other approaches, but it's also very useful when you're working on your own, or with a client, and you don't have lots of users. You can use a model of what you think your site is supposed to offer, to try to design your approach. And one of the great exponents of this approach, the evidence based, cognition based principle of design, is Edward Tufte. And one of the main ideas that he tries to get across to people, in terms of understanding good design process, is to try to be clear about

- What the information is?
- How do you make information clear?

This is useful when you don't have lots of time to work with users. It's useful when you're not really sure what it is that people are going to do. Maybe it's a totally different kind of site. So then you've got to narrow down where the content is and what that content's gonna be. So that's that sort of Edward Tufte approach.

And we're gonna use that alongside the stuff we've already been talking about, the user-centric approaches. And roughly summarizing those approaches, the user centered approaches are really focused on

- Who the users are,
- What the users want, and
- How they're gonna get the information.

Yeah, who they are, what they want, how they're gonna get their stuff. So this is the first, kind of, rule that I am going to summarize for you now. User experience first. Now, a lot of people will tell you that the Edward Tufte approach, which is all about information and evidence, and the user centered design approach, which is all about who those people are, and what they want to do, are pretty conflicting approaches. And that's true. They're both useful, but they're not necessarily on the same page about everything. And that's not a problem, we can cope with that.

- But there's one area where both those approaches fundamentally agree, and that's on the idea that user experience is primary. Before you do anything else you should consider the experience, okay? User experience must come before any code is written, all right? And user experience must be consistent.

I think I've been saying this all along and it's kind of obvious, but, consistency and the inner coherence of something is what makes it understandable and readable, yeah? If something looks like it's come from one side, and the rules apply across that side, then you're starting to make some kind of sense.

- But also, the way that the user is supposed to use the site has to be very clear from the beginning. So, when someone goes to a site, they normally know why they're going there. It might not be that it's the right site, in which case, let them go somewhere else. You've got to say, this is what I'm offering, here it is, up front. This is the evidence that you need to make an assessment about whether this is where you wanna be. You've got to try and get that information to users as quickly as possible.

So sites that contain very different designs with different areas are very confusing for users. So stick to your CSS template, that's the point of having a CSS template. You'll be surprised how many big name websites make this mistake. You go to one part of the website, it looks like one thing, you go to another part of the website, it looks like something else. And it's been designed by different people, and there's been no thought about how they interact. You may as well not have the same website if you're going to do that. But not understanding user experience is the cause of many major mistakes in almost all forms of software development. And software is often written with no clear idea how it's going to be used. And that's worse than not doing anything.

## 2.1.5 User Experience First

- Both these approaches are about as opposite as you can imagine, but they share one important thing in common :
  - **USER EXPERIENCE must come BEFORE any code is written**
  - **USER EXPERIENCE MUST BE CONSISTENT**
    - Sites that contain radically different design in different areas are very confusing for users. Stick to you template
- This is the cause of many major mistakes in most kinds of software development
  - Software is often written with no clear idea how it will be used. This is **worse than not doing anything.**

If you're not sure how the software's gonna be used, don't even bother writing it, right? Figure out what the use is for your website, get it clear. It doesn't matter whether you're Edward Tufte or whether you're Human Factors' researcher, that's the primary idea.

So the next concept, it leads on from that, and it's

- WTF. What's this for? When presenting information, you need your visitors to grasp what the information is quickly. You can judge a good design by how long it takes you to understand what it's about, all right?

So if you're looking at a website, and you're not really sure what it means, that's a problem for the site. It might look nice, it might seem presentable, but do you really know what it's for? That's the most important thing. The user visits the website looking for content. You've got to understand and clearly signpost that content. And I can't underline this enough, that's a fundamental principle. What's it for? Why am I here? Give a readable signpost. The second major guideline

## 2.1.5 Obfuscation Vs. Simplicity

- The worst kind of designs are those that look like 'corporate annual reports' (Tufte).
- Confusing content padded by generic imagery.
- Not designed to be read by anyone.

I want you to follow, obfuscation versus simplicity, okay? The worst kind of designs are those that look like corporate annual reports, according to Edward Tufte. And what he's basically saying is that it's confusing content padded together with generic imagery. So if you're looking at something and you've got a lot of information, there might be a lot of text, but you're not sure from scanning it what it means, then maybe it's padding. And then maybe there is nothing. It looks like it's fine on the surface, but when you just start to read it and you're not sure what you're doing, this goes back to this user experience idea. That user experience, it's challenging. The user doesn't really know, even though it looks like there's content, even though it looks nice enough, it looks professional, but they might not have any idea what it is. And that is a major problem. And that's what happens when you look at, according to Edward Tufte, that's what happens when you look at a corporate annual report, yeah?

- It's something, somewhere in here is important, but you're not sure where, and the images are there to remind you that it's all professional. People wearing suits with glasses and nice out of focus office backgrounds.

Beware of these kind of websites, because you don't really know what they're telling you. They're not designed to be read by anyone, and they're not really targeting you in the way you think. So I've got one good example of that I want to show. Simpler websites are better. Again, what's this for? What is it? What is the content? A news item, a product, a download. That's the first thing the user's thinking. What is this? The second thing is, where can I get it? The minimum number of clicks to get to content. If content is not on the front page, I want it in one click as soon as I know what it is. If I have to wait longer than that, there's gotta be some corporate reason. Some kind of business logic why you're delaying that transaction. Most websites don't need that kind of delay.

## 2.1.5 Obfuscation Vs. Simplicity

- **Simpler** websites are **better**.
  - What is it?
    - A news item, a product, a download.
  - Where can I get it?
    - **Minimum number of clicks to get to content**
  - Why is this here?
    - If it's not helping users to do the above, cut it out. That includes colour! Why are you using colour? WHY? Because it looks nice? What kind of reason is that?

If you want someone to use the site, get the information to the user quickly. This is what it is, here it is, get it, grab it, go. That's the experience people want. So, what is it? Where can I get it? Why is it here? So if it's not helping users to do the above, get rid of it. That includes all the color. I mean, everyone says color is great. Put color on our website and make it look colorful. And maybe that's true because color is inexpensive on a website. But why use those colors? This is the golden rule. I've mentioned this before and I'll always mention this.

- If you can't think of a reason why the color is there, don't use color, just use white.

Pick something very simple. Only build things in that make some kind of sense, so that there's a coherence to the overall philosophy of the experience. Colors bring with them confusion, use them sparingly. Pick very simple color palettes to begin with. Slowly build from that. There's nothing wrong with black and white. Black and white and red, that's the BBC website. All right, very simple. And that's, I think, one of the top 20 websites in the world.

- So why use color? Because it looks nice? That's not a good enough reason.