Government Digital Service

Design Principles

1. Start with needs*

*user needs not government needs

Service design starts with identifying user needs. If you don’t know what the user needs are, you won’t build the right thing. Do research, analyse data, talk to users. Don’t make assumptions. Have empathy for users, and should remember that what they ask for isn't always what they need.

2. Do less

Government should only do what only government can do. If we’ve found a way of doing something that works, we should make it reusable and shareable instead of reinventing the wheel every time. This means building platforms and registers others can build upon, providing resources (like APIs) that others can use, and linking to the work of others. We should concentrate on the irreducible core.

3. Design with data

In most cases, we can learn from real world behaviour by looking at how existing services are used. Let data drive decision-making, not hunches or guesswork. Keep doing that after taking your service live, prototyping and testing with users then iterating in response. Analytics should be built-in, always on and easy to read. They’re an essential tool.

4. Do the hard work to make it simple

Making something look simple is easy. Making something simple to use is much harder — especially when the underlying systems are complex — but that’s what we should be doing. Don’t take “It’s always been that way” for an answer. It’s usually more and harder work to make things simple, but it’s the right thing to do.

5. Iterate. Then iterate again.

The best way to build good services is to start small and iterate wildly. Release Minimum Viable Products early, test them with actual users, move from Alpha to Beta to Live adding features, deleting things that don’t work and making refinements based on feedback. Iteration reduces risk. It makes big failures unlikely and turns small failures into lessons. If a prototype isn’t working, don’t be afraid to scrap it and start again.

6. This is for everyone

Accessible design is good design. Everything we build should be as inclusive, legible and readable as possible. If we have to sacrifice elegance — so be it. We’re building for needs, not audiences. We’re designing for the whole country, not just the ones who are used to using the web. The people who most need our services are often the people who find them hardest to use. Let’s think about those people from the start.

7. Understand context

We’re not designing for a screen, we’re designing for people. We need to think hard about the context in which they’re using our services. Are they in a library? Are they on a phone? Are they only really familiar with Facebook? Have they never used the web before?

8. Build digital services, not websites

A service is something that helps people to do something. Our job is to uncover user needs, and build the service that meets those needs. Of course much of that will be pages on the web, but we’re not here to build websites. The digital world has to connect to the real world, so we have to think about all aspects of a service, and make sure they add up to something that meets user needs.

9. Be consistent, not uniform

We should use the same language and the same design patterns wherever possible. This helps people get familiar with our services, but when this isn’t possible we should make sure our approach is consistent. This isn’t a straitjacket or a rule book. Every circumstance is different. When we find patterns that work we should share them, and talk about why we use them. But that shouldn’t stop us from improving or changing them in the future when we find better ways of doing things or the needs of users change.

10. Make things open: it makes things better

We should share what we’re doing whenever we can. With colleagues, with users, with the world. Share code, share designs, share ideas, share intentions, share failures. The more eyes there are on a service the better it gets — howlers are spotted, better alternatives are pointed out, the bar is raised. Much of what we’re doing is only possible because of open source code and the generosity of the web design community. We should pay that back.

www.gov.uk/design-principles

GOV.UK