

10-minute read

Design

Business

Strategy

10 lessons I've learned

in 14 years of designing products



hepton

Russ Hepton · hepton.design

You see the world in a way that nobody else does

So does everybody working for you, and so does every one of your customers. Design is how you get other people to see what you see too.

Contents

02	Design isn't optional, it's business strategy
03	Most businesses think they do design. They don't
05	Businesses do one of two things. Adapt, or fall behind
07	Nothing is ever done
09	Chasing trends kills creativity
11	Design needs time for play
13	Tools don't make designers better
14	Tools can build systems and open up what wasn't possible
16	Businesses underestimate research, testing and their customers
18	AI will get more done. It will never give your team taste

I've spent fourteen years designing products for all kinds of companies, all shapes and sizes, mostly embedded straight into product teams. These are the ten lessons that come up most.

This isn't a list of tips and it isn't a step-by-step. It's ten things I know to be true about what design does for a business.

Design isn't optional, it's **business strategy**.

1

Get design wrong and you don't have a customer. You have stock you can't sell, or a service with no subscribers.

Design isn't a layer you add at the end to make something presentable. It's the thing that decides whether anybody comes back.

Customers can sense it. A product looks like a great idea, then they get it home, take it out of the box, start using it, and it doesn't work. That's the moment the trust, the connection and the satisfaction are gone, and they don't come back. The same thing happens on a website. It doesn't load, it doesn't behave, and they leave and go and find somebody else. Businesses whose customers have that experience don't tend to last.

The objection is always the same, and it usually comes from finance. Design is a cost line, not a strategy line. And it's fair, it does cost money. But if you want something quick and cheap it won't be good, and if you don't understand your customers well enough to design from what you learn about them, you won't have a customer. You'll have stock you can't sell, or a service with no subscribers, because it doesn't solve a real problem.

At the James Dyson Foundation we took box bookings up five times over and resource downloads up ten times over. That didn't come from adding features. It came from understanding who was actually using the website and building for them.

Try this

Take one product or one page and ask what it would cost you if a customer used it once and never came back. Then find out whose job it is to know that number. If it isn't anybody's, that's a red flag.

Most businesses think they do design. They don't.

2

| A hundred support tickets a day isn't a hundred problems. It's one.

There's a difference between having designers and doing design, and it shows up most clearly in how a business responds to a symptom.

The clearest tell is firefighting. A hundred support tickets a day, and the response is to get somebody to write more help pages with the copy checked by a manager. But a hundred tickets a day isn't a hundred separate issues. It's one, and it's sitting in the product. Here are some more:

- Designers having to fight for time to do research and testing
- The design team being let go the week the product launches
- No fresh junior talent coming in, so no new perspectives either
- Brand guidelines treated as finished, so nobody ever asks what the brand actually is any more
- An assumption that every other business thinks about design the same way
- No time, space or tools to start the process away from a computer

The ones who really do design right look different from the inside. Design is the driving force and the hub of the culture, and that doesn't just mean what you see on the surface. It means the whole process, from finding out what's genuinely wrong all the way through to testing whether the fix worked. It's rare, and it's hard for an older company to get to. For a startup it has to be that way from the ground up, because people expect more from products now than they used to, and the public can spot something that doesn't work almost immediately.

I saw it done properly at the Google and Avado partnership, on Squared Online. The whole thing was built around cohorts of people learning together, and every decision came back to them, from the research all the way through to what they saw on screen.

Try this

Take last month's support tickets and group them by cause rather than counting them. Then follow the route to the cause. That's the real problem that needs solving.

Businesses do one of two things.

Adapt, or fall behind.

3

Falling behind doesn't start with a bad decision. It starts with not wanting to try something new.

Four things force a business to adapt: technology, legislation, culture and competition. Most only notice when it's already the last one.

Kodak invented the digital camera in 1975. They had it in their hands years before anybody else, and they shelved it because it threatened their camera film sales. Lego went the other way. Nearly bankrupt in 2003, and the way back was going into people's homes and watching children play to get insight. It wasn't a boring offsite, it was simply watching.

Competition looks like Oral-B and Suri. Oral-B held the market and every electric toothbrush looked the same, same poor battery life, ten hours to charge, seventy quid or more. Suri looked at the whole landscape rather than at Oral-B, and rebuilt around the actual problems. Oral-B are still in business and they may well catch up, but I won't be going back.

Legislation looks like diesel engines. Tightening regulations meant bolting parts on as band-aids, so exhaust gases got pumped back into the cylinders and everybody called it a solution. James Dyson designed a cyclone that fitted to the exhaust instead, and as far as I can tell he did it because of the wider problem rather than the rule. Car manufacturers never adopted it, but that's the real difference between reacting to a regulation and going after the thing the regulation actually exists for.

And then there's AI, which is where this is happening right now. Small businesses and startups have the upper hand, because they're multiplying their output at a fraction of the cost while larger organisations sit in grooves they've been in for years. It all comes down to fear.

I'll give you mine, since it's the same story.

Two years ago there was a hard ceiling on what I could build and that ceiling was my own coding ability. Now there isn't one. In the last few months I've built a portfolio site with working prototypes in it, a brand, a photography direction and an entire design, email and marketing system. Same person, same fourteen years of experience. What changed is that I was willing to try something I didn't understand yet.

Try this

Name the last thing your business tried that it didn't already know how to do. If nothing comes to mind quickly, that's your answer.

Nothing is ever done

4

It's the end of a chapter, not the end of the story.

Ready for now is a real and useful state. Finished isn't.

This holds at every scale. A product, a design system, a single page, a social post. Any of them can be ready for now. But then real people use it, real data comes back, and somebody has a problem nobody predicted. I've never designed and shipped to forget. I ship knowing that's the beginning, and then it repeats. Research is never done and testing is never done. It's a cycle, and it should be woven into the week, the month and the quarter rather than bolted on when something breaks.

You can see what happens when it isn't. I've looked back at sites I designed a decade ago that haven't moved since. They've got stuck. And over time new problems show up anyway, because the technology moved on, the competition moved on and the legislation changed.

What changed for me was that I stopped working towards a launch date and started building systems for repetition from the beginning, so the next cycle could start before the first one had finished. It's a mindset first and a process second. The mindset starts with one person and spreads to the team, and then the process is what lets the team actually do it.

And of course stakeholders aren't wrong to want a finish line. They need one, so do designers, there's a budget that has to close. So the answer is that it's the end of a chapter, not the end of the story. Make the trade-offs, take the forks, get it over the line. Sketching and thinking are cheap. Rebuilding a bloated mess in two years is expensive. Which is also why this isn't a lesson about perfectionism, it's a lesson about accepting that perfectionism is an illusion. It's fear and insecurity in disguise. There will always be faults. We're on the seventeenth iPhone, and there's a new one every year. Apple didn't stop at the first one.

Try this

On the next thing you build, write the scope so it runs past the launch date. If the plan ends the day it ships, it was never a plan.

Chasing trends **kills** creativity

5

A convention earns its place. A trend just hangs around until the data catches up with it.

Copying what's popular makes you invisible. But some things that spread aren't trends at all, and telling the two apart is a skill in itself.

Liquid Glass is the one right now. Apple does it once, and suddenly it's everywhere. Everybody who copied it has stopped being memorable and faded into the same sea as everyone else. And it hasn't even worked for Apple. It's been fraught with accessibility problems, to the point where they've had to add a way to turn down how much of it you actually experience.

Material Design did it before that, and it's a genuinely brilliant design language, the card and layering structure especially. But I've watched teams install the components and never build on them, and everything they make ends up looking like Google made it. Which is quite sad when you think about it. All those apps and websites, built on top of an excellent system, and not one of them ever grew a personality of its own.

I've chased one myself. On my old portfolio site years ago, I wanted a splash loader, because it gave people a bit of context before the site came in. It was tasteful, I thought. It also wasn't loading anything. It was a loader for the sake of having a loader, and people were dropping off before they ever saw the site.

That's when I started asking whether I wanted it, or whether the user ever did to begin with.

Here's the other half, though. Think about a car. Two or three pedals, a wheel, a shifter, a handbrake, indicators. Cars didn't used to be like that, they were all completely different, and then the arrangement we have today won because it worked. Websites have gone the same way. That isn't a trend, that's a convention, and it earned its place. If a new idea changes what customers expect from that point on, it becomes convention. A trend is something that hangs around for a while because it looks like a good solution, and the wider it spreads the more it reads as merely fashionable, and eventually the data shows it never helped anybody and it fades.

Which means holding out isn't a virtue either. If the trend turns out to be right then it was a convention all along, and refusing it just looks like a mistake. Holding out should mean experimenting instead, in a way that's useful and suited to your customers, and that single experiment might turn into the next new convention.

So keep asking the bigger question. Electric vehicles are already asking whether we need the pedals at all. What about us? Do we even need a user interface in 2026? What is a user interface, actually? Those are the questions that keep a business out of the time capsule and able to adapt.

Try this

Point at one thing in your product and ask honestly whether you'd still do it if nobody else was doing it.

Design needs time for **play**

6

Play isn't slack, it's exercise. We've all got creative muscles and they need working.

Play isn't rest and it isn't a perk.
It's the practice that produces the
thing no brief would ever ask for.

In my week it looks like getting off the computer and looking at the world properly. Going for a walk, which is play happening in your head. Taking something off the backlog that's never going to be prioritised and drawing three answers to it on paper. Keeping a small sketchbook for doodles and storyboards. And blocking half a day a week to try something new, whether that's a tool, a material or a medium.

The best example I've got is a new brand I built for a notebook company I run. I spent two weeks with inks, bleach, waterproof paper, brushes, water and tea, anything that would leave a mark on a page. I didn't open a computer once in that fortnight. Then I narrowed all of it down, and what came out of that mess is the logo and the brand elements. No brief would have produced that.

I don't ask clients for extra time to play, by the way. I weave it into the process and they know it's there. And whenever it produces something, the reaction is always the same. They want to push it further, because it stopped being a ticket in a backlog and became something they can see. Often that's the thing that ends up setting them apart from their competition.

So it's worth being clear that play is the opposite of slack. It's exercise. Most of us had it knocked out of us at school and then again in our first few jobs, so it's a skill that has to be relearned. It means letting go of the constraints and the tools and the rules for a while and just being human, which is the part no machine is going to do for you.

Try this

Find the ticket in your backlog that keeps getting pushed down and is never going to make the cut. Give somebody an afternoon and a pencil and let them play with it.

Tools don't make designers **better**

7

The problem was never the tool. It was that nobody wanted to do the housekeeping.

They don't speed things up either, and they don't define what you make. Buying one is just a fast way to feel like you've solved something you haven't diagnosed.

The argument I'm tired of having is about industry-standard tools. It's such a constricting idea. I don't care what tool you use. If there's a better one for you, use that. If you think the ones we've got aren't good enough, say so and we'll try something else. What matters is whether the output is any good.

I watched a client bring in Asana to fix a process and documentation problem, because the feeling was that Jira was too heavy for what we needed. But the problem was never Jira. It was that nobody wanted to do the housekeeping. Nothing was being written down before, and nothing was written down after. In the end nobody used either one and the whole thing was quietly forgotten.

So if it isn't the tool, what does make a designer better? The right mentoring. The right team around them. Time to play. The right amount of time pressure, and that works both ways, because too much time and nothing moves forward, too little and you get cheap results. Getting outside. Travelling, and spending time somewhere that doesn't work the way home works. Having a side project. And failing at things, repeatedly. Over and over again.

Try this

Before the next licence renewal, write down the problem the tool was bought to solve, and whether that problem has actually gone.

Tools can **build** systems and open up what wasn't possible

8

Tools closed my skills gap. They didn't close the gap between a good idea and a bad one.

What's genuinely changed is real.

I can produce consistent interface design far more accurately than I could in Photoshop, and hand it to developers in a form they can actually use. I can synthesise research and get to insight and recommendations in a fraction of the time. I can take a rough sketch and have a working prototype from it. And five years ago, if I wanted the thing in my head to exist at all, I had to hire a full-stack developer. Now I build it. My portfolio isn't screenshots of my work any more, it's working prototypes, actually built in code. Add the APIs and the data and they'd be fully functioning tools.

Now the honest part, because this is where it gets oversold. If AI genuinely made me faster, I'd have finished that site in one or two prompts. It took hundreds, over months. The individual jobs did get faster. The work didn't. What actually happened is that I spent every minute I saved on going further with the project, building what was always out of my reach before, exploring more directions and testing more of them before committing. Same hours, far more considered output, just me. There's an up-front cost too, because getting a tool to be genuinely useful to your business takes real time to embed it into your workflow before any of it pays back.

And if you're about to buy something, start with a diagnosis rather than a fix. You'll always see the symptom before the cause. Long, expensive handovers where what gets built doesn't match the design are a symptom. The cause might be that the design tool doesn't give developers what they need, so they're eyeballing it and spotting the difference. But even after you buy the right tool you'd still have to redesign how a handover happens. The tool is rarely the last step in the solution.

Try this

Next time a tool gets proposed, ask for the symptom and the cause written down separately before anybody signs anything.

Most businesses **underestimate** research, testing and their customers.

9

Without testing you don't know you've built the right thing.
You just hope you have.

The front half of the process is the first thing cut and the most expensive thing to skip. It's the part where you find out what's genuinely wrong before anybody designs anything.

Glacier Drop is the clearest example I've got. One of the largest token distributions there's been, four and a half billion tokens across eight million wallets and eight ecosystems. The first version was all over the place, because it was being designed around what the technology could do at the time, and what the business wanted rather than what users were actually trying to achieve. The design team knew it had problems, and so did the CEO. We just couldn't prove it or be precise about what the problems were, and without proof we couldn't move the other teams.

So after fighting for time to test, we tested it, and testing showed exactly what we'd suspected. Too many elements, too many buttons, too much cognitive load. We rebuilt around one clear path to the one thing people had come to do, instead of trying to cater for a thousand edge cases. The tech team were sceptical. We launched, and we went past every number we'd predicted.

The cheapest useful research I know is secondary desk research paired with primary validation at scale. Get a feel for where the market actually is, then ask your own questions of real people based on what you found. And when somebody tells me there's no time or budget, I don't argue, I put three options on the table:

- 1 Minimal research, some desk work to sanity-check the assumptions. Low confidence, high risk.
- 2 One round of primary research or testing over a few days. Some validation, medium risk.
- 3 Multiple rounds, synthesis, prototypes and testing. Full confidence, better product, lowest risk, most expensive.

Then they choose. It becomes an executive decision rather than a design one, which is where it belongs. And whichever they pick, I'll do my best work inside those constraints.

Try this

Put your next design in front of five real customers before anybody builds it. Five is enough to change your mind.

AI will get more done. It will never give your team **taste**.

10

AI gave me a hundred versions.
Knowing which one to keep is the skill.

AI raises your output. It doesn't raise your judgement. And taste is a word that gets thrown around, so here's what it actually means without using it:

- Knowing what good looks like
- Knowing the difference between one attempt and a hundred attempts
- Knowing the difference between something that works and something that looks nice
- Knowing that if it looks nice and doesn't work, it instantly becomes ugly
- Being fussy enough to say this won't do
- Putting your own personality into the work rather than whatever everyone else is doing
- Looking at nature and the real world and bringing that back into a product
- Understanding people well enough to actually take away something that hurts

Here's how I know a machine can't hand you any of that. I've been art directing a photography project, it generated hundreds of images. Every one plausible, every one confident, and almost all of them wrong in a way the model couldn't see. Same with the animation work. It took refining and crafting, over and over, until I reached the one I wanted. The failure mode isn't that it gets something wrong once and you catch it. It's that it will keep handing you wrong options forever, and somebody has to know when to stop at a good one.

It shows up structurally too. When I was building my own site, it was happily producing pages that weren't updating components globally, because it couldn't see past the foundations it had suggested at the start of the project. I had to change the stack and build a proper design system. It didn't know that. It would have carried on building with no real components indefinitely.

And it has habits. It reaches for the same fonts and the same layouts, as though it's addicted to whatever's fashionable. Which brings this back round to lesson five, and to the reason none of the previous nine stop mattering just because the tooling got good.

Try this

Generate ten versions of something, keep one, and write down why you kept it. That paragraph is what you're paying your designers for, and it takes years to get good at.

If you only take one of these, take the first.

Design isn't optional.
It's business strategy.

Question is, are you designing something people need?

I'm not asking you to restructure anything. Take one of these into next week and test it. Pick the one that made you slightly uncomfortable, try it on something small, and watch what changes. I bet something good happens.

And if you'd rather have somebody in the room while you do it, I'm always on the lookout for new and exciting projects. I'd love to hear about you and yours.

[Book a call](#)

hepton

