Unsplash © Shahadat Rahman

Why building software is like building a house – and how to make sure yours stays standing

Software engineering can be hard to grasp for non-technical founders, but it doesn’t have to be a total mystery. James Zhao, co-founder of Thought&Function, breaks down the tools, expertise and timelines you’ll need to create a successful end product.

If you’re unfamiliar with software engineering, it can be a confusing process to get your head around. Whether it’s knowing who you need on board or just where to start, it’s easy for the path ahead of you to look pretty impenetrable.

That’s why I like to think of it as like building a house. Both follow a clear roadmap from plans to completion, and both need a specific combination of skill sets to bring it all together. If coding feels like a black box to you, hopefully this imagery can turn it into — if you pardon the pun — something a little more concrete.

Who you’ll need

When building a house, there are certain roles you can’t do without. The same is true of building a successful tech product.

For starters you'll need a project manager to oversee the other roles and make sure everything stays on time and to budget. You'll also need an architect who will ensure that everything you're building is fit for purpose. While a construction architect will see that the designs for your house can accommodate your family's living needs, a software architect will be responsible for making sure your app or service can hold its intended user numbers.

Then there are the backend and frontend engineers who will build the majority of the software. Backend engineers handle your product's fundamental infrastructure — like electricians, plumbers and bricklayers, they fashion what goes on behind the scenes. Meanwhile, frontend engineers code what the user can see when they use the software, like a carpenter making the house's stairs and fittings.

Lastly, you'll have UX (user experience) and UI (user interface) designers. They're the interior designers of your build, who give the software the look and feel it needs to attract customers. The two roles might sound interchangeable but they are distinct from each other — UI is all about the interface's aesthetics, while UX is about how intuitive the software feels to use, a little like designing which direction a door should swing (or slide), or placing light switches in their expected position.

Sometimes you'll also need to call on third-party services to complete your software. Like buying a fitted kitchen, this just means adding ready-made services like system logs, analytics or email servers into your product rather than building it all from scratch.

Build from the ground up

Just as you can't build the rest of the house without first laying the foundations, every tech product has to start with getting the software's essential infrastructure in place.

A vital part of that infrastructure will be development operations or dev ops. These are specialist roles that look after the servers that your code runs on, as well as the networking and security to make sure the right servers and machines are talking to each other for your software to operate.

In a house build, dev ops would be the infrastructure that makes your home habitable — the roads, the sewer system, essentially what you pay your council tax for. But if you wanted to throw wild parties with upwards of a hundred people, you're essentially inviting 'traffic' to your site. You need bigger roads with good parking or a nearby station, and in the house itself, you'll need wider hallways and more toilets for your party goers to use (and for introverts to hide

in).

It’s the same with software development. The traffic you’ll attract in future, the users and functions you’ll need to support, all this informs what infrastructure the dev ops need to put in place at the start.

As for your foundations, part of that will be boilerplate code. Boilerplate is code that’s been pre-made by the developers you hire and forms the structure of what your architect and engineers will create. Rather than coding everything from scratch, it provides a blueprint to build from that’s already been proven and will accelerate development.

Once you’ve got that foundation in place, then you can start building your architect’s plan on top of it. Always remember to prioritise getting the key functionality and any necessary third party services done before moving on to making it look good. After all, there’s no point trying to hang lamp shades before the electrician has wired in the lights.

Keep up with the maintenance

When you’ve launched your product, the process doesn’t stop there. You don’t leave your house alone after moving in — whether it’s regular maintenance, refreshing the decor, or building new rooms for a growing family, there’s always further work to do.

When your product is on the market, you and your team will be constantly working to bring out new, improved iterations. Sometimes that might be fixing bugs or responding to user feedback, or it might be expanding the software’s infrastructure to accommodate planned new features.

It may also be responding to external factors, like a competitor introducing a new feature of their own. Look at social media platforms, for example — when Snapchat introduced its Stories feature in 2013, it wasn’t long before its rivals all developed their own short-form media features to fold into their services.

But the most important changes you make will always be centred around your customers and their needs. Every change you make to a house — from boiler repairs to redecorating bedrooms — is done to benefit the people using that space. If you think of your product and your users in the same way, then you’ll be on the right track.

James began his career as a software engineer – and quite a successful one – working for Barclays, KPMG, and on various projects for McLaren, Aviva and the Metropolitan police. Four years ago he co-founded Thought&Function where he brings commercial and product expertise together – along with an

understanding of the start-up process – to give start-up founders the start they need: advising on strategy, handling marketing and sales, all while still dabbling in the build process.

Article by JAMES ZHAO