# Choosing a technology partner: 20 questions to ask

The questions we would ask ourselves before trusting anyone with a business-critical platform. Use them with every provider you talk to, us included.

PROVIDER _____ DATE _____

A platform that runs your HR, your sales or your finance is not a website you can redo next year. The partner you choose will shape how your company works for a long time. These twenty questions help you find out, before you sign, whether they can carry it.

Print this page, share it with your team, and use it with every provider you talk to. Us included.

## Do they understand your business?

☐ *1* **What problem do you think we are solving?**
A good partner can explain your problem back to you in plain words, before talking about technology.

☐ *2* **What would you not build?**
Someone who agrees with every idea is selling hours, not judgement.

☐ *3* **Which parts of our process should stay as they are?**
Software should follow the way your company works, not force a new one for no reason.

☐ *4* **What does success look like in six months, and how will we measure it?**
If nobody can name the result, nobody will be accountable for it.

## Who will actually do the work?

☐ *5* **Who exactly will work on our project, and can we meet them?**
The people in the sales meeting are not always the people who write the code.

☐ 6 **How senior is the team, and who makes the technical decisions?**
Ask who decides on architecture, and how those decisions are written down.

☐ 7 **What happens if a key person leaves?**
Look for documentation, code review and shared ownership, not heroes.

☐ 8 **Which languages and time zones do you work in?**
Live conversations during your working day prevent most misunderstandings.

## How do they build?

☐ 9 **When will we see working software?**
Weeks, not months. Ask for a prototype of the riskiest part first.

☐ 10 **How often will we be able to try a new version?**
Short, regular releases keep surprises small.

☐ 11 **How do you test your work?**
Look for automated tests and code review on every change, not a testing phase at the end.

☐ 12 **How do you plan for growth?**
Ask how the architecture will handle more users, more companies, more countries and more languages.

## How do they keep it safe?

☐ 13 **How do people sign in, and who can see what?**
Single sign-on, roles and permissions should be part of the design, not an add-on.

☐ 14 **Where will our data be stored, and who can access it?**
This should be agreed in writing before any work starts.

☐ 15 **How do you know who changed what?**
A server-side audit trail records sensitive actions where nobody can quietly edit them.

## What happens when it’s live?

☐ *16* **Who watches the platform after launch?**
Monitoring and alerts should tell you something is wrong before your users do.

☐ *17* **How are backups and recovery handled?**
Ask for the written procedure, not a reassurance.

☐ *18* **How do we ask for changes, and how fast do they ship?**
The process for small changes matters as much as the first release.

## What do you own at the end?

☐ *19* **Who owns the code, and what is documented?**
Ownership terms belong in the contract. Documentation is what makes them useful.

☐ *20* **Could another team take over tomorrow?**
If the answer is no, you are not buying a platform. You are renting a dependency.

### How to use this list

You will not get perfect answers to all twenty questions, from anyone. What matters is how a partner answers: with specifics, with examples from their own work, and with the honesty to say “we don’t do that”.

If you want to hear our answers, book a call. We will go through the list with you.