Founder question list

These are the technical questions I ask a founder in a first meeting, and what I listen for. They won't replace a review, but they usually show where the real risks sit.

1. Architecture and cloud. How is the product hosted, and what would break first if you had three times the customers?

Good: They can name the host, the database, and a specific bottleneck, plus a plan to grow in place.
Worrying: A vague "we're on the cloud," or they already see a rewrite as the only way to grow.

2. Shipping changes. How do you get a change into production, and can someone else do it if you are away?

Good: Builds and tests run automatically, a person approves production, and they can roll back in one step.
Worrying: Deploys run from one laptop, and releases stop when that person is out.

3. Tests. Which parts of the product have automated tests, and which parts are checked by hand?

Good: The money and sign-in paths have tests that block a bad change, and they know where coverage is thin.
Worrying: "We test manually before each release," or they cannot say what is covered.

4. Access and leavers. Who has admin access to production, the code, and the cloud account? What happens when someone leaves?

Good: Named people, MFA required, and a short checklist they have actually used to remove a leaver.
Worrying: Shared logins, MFA is optional, or former contractors may still have access.

5. Backups. When did you last restore a production backup, and how long did it take?

Good: A dated restore test, a known time to recover, and backups stored where a single leaked key cannot delete them.
Worrying: "Backups are on," but nobody has ever restored one. Recovery time is a guess.

6. Outages. How do you find out the product is down, and who responds after hours?

Good: An alert reaches a person before customers write in, and someone is clearly on call.
Worrying: Customers or support notice first, and there is no on-call rotation.

7. Ownership. Who owns the repositories, cloud accounts, and app store accounts? Is any of that in an agency's or contractor's name?

Good: The company owns every account, the contracts assign the IP, and agencies work inside those accounts.
Worrying: An agency holds the code, the store listing, or the signing keys, and the contract is silent on ownership.

8. Key-person risk. If the person who built this were out for two weeks, what would stop?

Good: At least one other person can deploy and reach production, and there is a short written overview of how the system fits together.
Worrying: One person holds deploys, production access, and the knowledge of how it all works.

9. AI use. What in the product is actually AI, where does customer data go, and is there a spend limit?

Good: They can separate real AI from ordinary software, name the vendor, say what data is sent, and point to a cap and basic data terms.
Worrying: The pitch says "AI-powered," but they cannot say what the model does, where the data goes, or what it costs.

10. Cloud costs. What do you spend on cloud each month, and which parts of that could you cut without hurting customers?

Good: They know the monthly number, what drives it, and a few unused or oversized pieces they could turn down.
Worrying: They do not know the bill, or it has been growing with no one watching.