20 QUESTIONS TO ASK A SOFTWARE DEVELOPMENT TEAM

A GUIDE FOR INTERVIEWING DIGITAL PRODUCT DEVELOPMENT TEAMS TO FIND THE RIGHT ONE FOR YOUR PROJECT

Created by

JMG

Introduction

Digital products are never truly done.

They exist in a dynamic environment where user needs, technology, market conditions, security threats, and regulatory requirements constantly change.

Owning a digital tool like a mobile app or custom software can be likened to owning a house or a car, where continuous improvement is essential to maintaining its lifespan.

This makes choosing the right software development partner foundational to the success of a digital project. It starts with interviewing multiple development teams and finding one that aligns with your business goals.

Comparing one digital product development team to another isn't apples to apples. The following page lists all there is to consider.

Things to Consider When Comparing Digital Product Development Teams

Expertise: Incorporates best practices, avoiding common pitfalls and speeding up development

Timely Delivery: Manages the project efficiently to deliver on time

Cost Efficiency: Manages your budget effectively, avoiding unnecessary expenses and ensuring a good ROI

Quality of Work: Ensures high-quality software meets your standards and requirements

Technical Compatibility: Aligns with your project's needs, ensuring technical compatibility and optimal performance

Long-Term Support and Maintenance: Provides ongoing maintenance and support to address issues, make updates, and add new features

Effective Communication: Maintains open lines of communication, provides regular updates, and is responsive to your needs

Alignment with Your Vision: Understands and aligns with your vision and goals, ensuring the final product meets your business objectives

Innovation and Creativity: Brings innovative ideas and creative solutions to the project, enhancing the product's features and user experience

Reputation and Trustworthiness: Has built a reputation in their community for being a reliable partner

Scalability and Flexibility: Scales their services up or down and adapts to new requirements, ensuring the software evolves with your business

Risk Mitigation: Foresees and mitigates potential risks, ensuring smoother project execution and reducing the chances of costly errors or delays

Getting Started

With everything to consider when choosing a software development team, where do you start? What questions do you ask? What is considered a red flag?

With decades of experience building digital products, the team at JMG has seen it all, so we did some of the legwork for you by compiling a list of 20 questions to ask when interviewing a digital product development team.

The questions are divided into three categories: About the Team, Development Process, and General Business.

Each question includes insights into the type of answers you may receive, yet it's important to understand that no two teams will answer these questions the same way.

This guide is exactly that—a guide for determining which digital product development team best suits your project and work style.

At the end, you'll find an interview sheet and more information about JMG, the creators of this guide.

Let's get started.

Table of Contents

Questions about the Team, Experience, and Work Style

01 Where are your developers located?

Are they overseas? Spread over the U.S.? Local? A mixed bag?

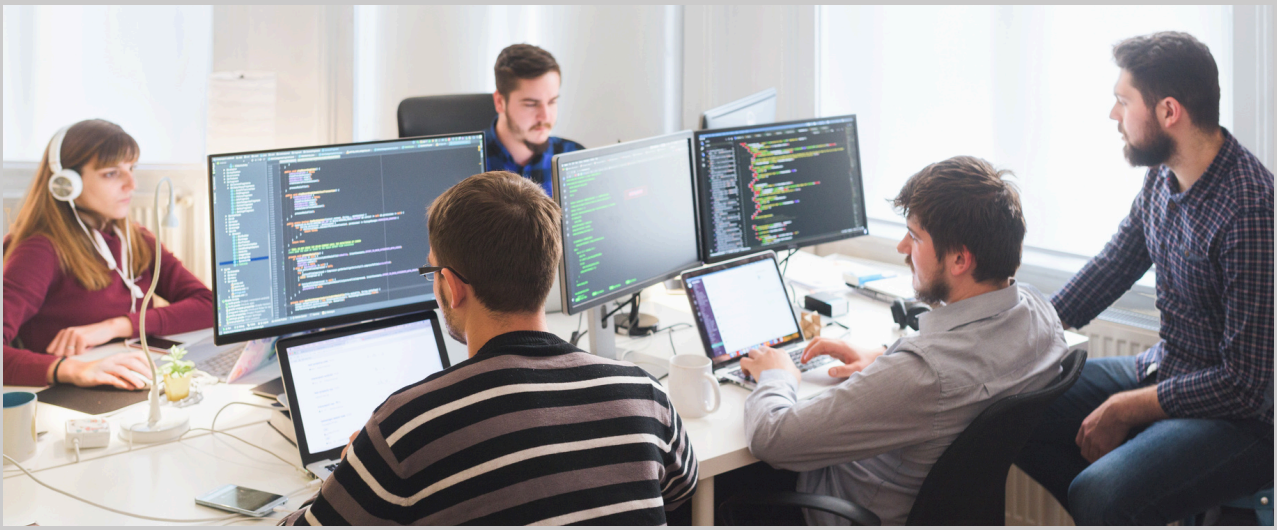
This question is arguably one of the most important questions to ask a software development team, so we listed it first.

You might think you're working with a software team based in the United States (often called "onshore" in the software industry) because the sales team or project manager you've been communicating with in the initial phases is based in the U.S.

Then, as the project picks up steam, you're communicating with a developer in a timezone 12 hours ahead of you.

This isn't necessarily a red flag. We've contracted with some great overseas developers on a few projects, but it's best to ask the question initially rather than be blindsided by the answer later.

02 How many developers will be working on my project, and how skilled are they?



Staffing a project isn't one size fits all.

Timelines, integrations, and the number of platforms are all factors that determine the developers needed.

One of the most successful and efficient strategies is to have mixed experience levels on the development team. While the most experienced developers architect the platform and manage its progress, the rest of the development team carries out the day-to-day development tasks.

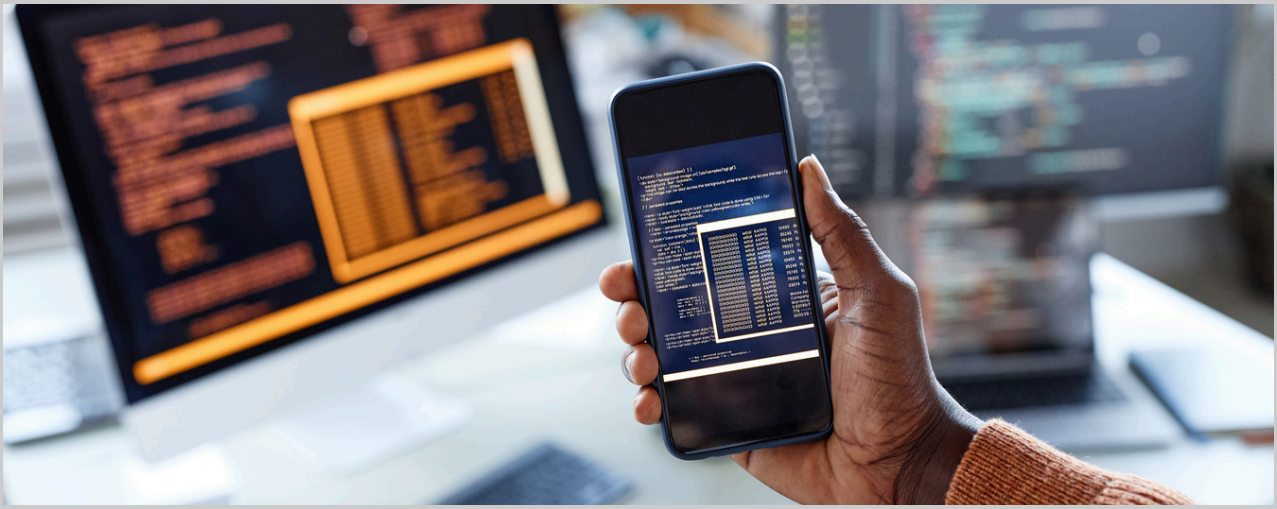
03 What projects are you working on right now? How will my project be prioritized?

This question tests the development team's workload and how they manage it.

Sometimes, development teams (typically overseas ones) prioritize projects based on which brings in the most revenue.

Every project's needs ebb and flow based on its current development stage, and it's important to know that a development team won't halt progress on your project when something bigger comes in their door.

04 What have you developed that we can test out?



Just like you would test drive a car before buying it, playing around with an app or visiting a website developed by the team you're vetting is a great way to see if they can do what they say they can.

Building an app?

Ask which apps of theirs are on the app stores, then download and check them out.

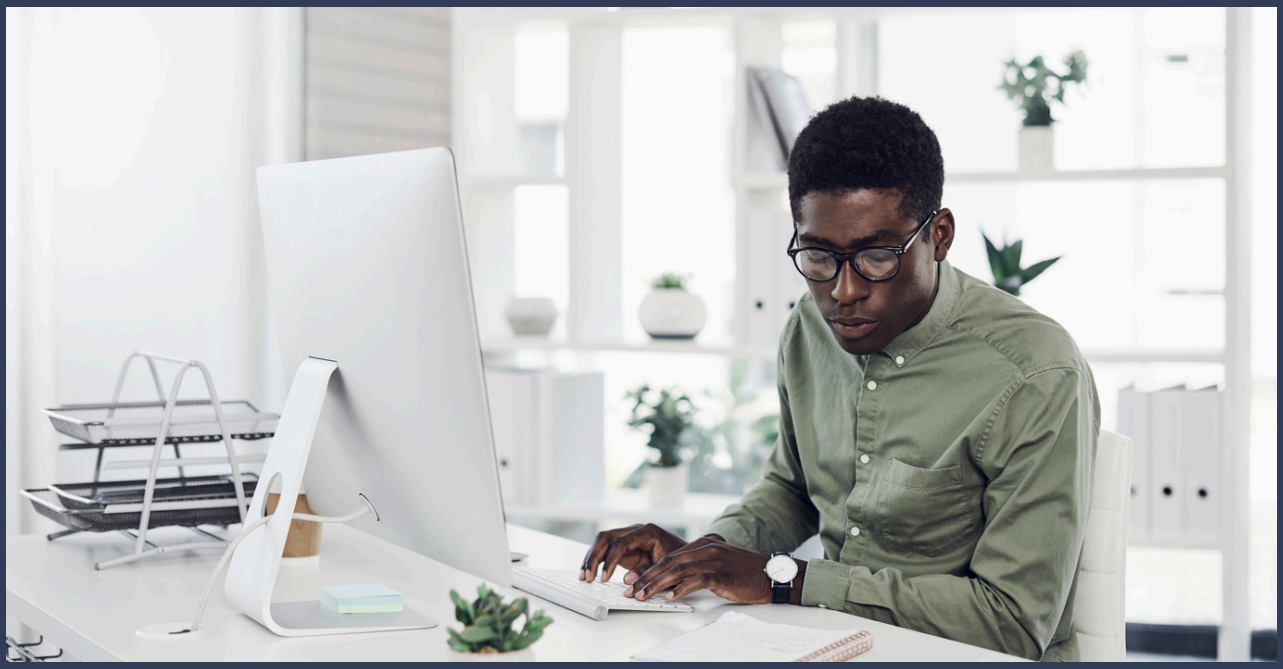
Building a website?

Research the projects they've worked on and see how their websites flow.

If you're building a backend platform like an employee portal or custom CRM, they may have a demo environment available to test.

In any case, if something isn't publicly available to review, ask for case studies and contact their references, which leads to the next question.

05 Can you provide references?



All reputable development companies have references ready to go and are happy to share them. If you have to ask for these more than once, it's not a great sign.

Companies will give you the contact information of their happiest customers, so bonus points should be awarded if they also point you to a neutral, third-party review site (like Clutch.co) where clients leave project feedback, and companies don't have editorial control over what's said.

Questions about the Development Process

06 What does the design phase look like?

Every software company's design phase looks different, so you'll get various answers here.

> The key is to review your design-spend priorities before asking this question because costs can add up quickly in this area.

Design doesn't generally need to be the most significant cost of your project, especially if you're taking an MVP approach to building software. A reputable software development team will turn this question around on you by asking questions about your project to determine how far down the rabbit hole you need to go in the design phase.

A caveat to this is when working with overseas developers, where it's important to have super detailed designs rather than leaning on offshore developers to infer user journeys.

07 How will you keep us updated on progress?

Miscommunication is the number one reason a project fails.

The key here is staying fluid with how much communication must occur.

If you are nearing the end of a major release, you may communicate with the development team several times a day.

If your product is in maintenance mode, you might only chat every other month. It is important to find a rhythm that keeps everyone informed and comfortable with the project's current state throughout the development cycle.

Here are a few things to look for regarding project communication:

- **Weekly Updates:** At the project's onset, the development team's project manager should establish a weekly meeting of key players from both sides to review progress and solve development hurdles.
- **Day-to-Day Response Time:** As developers are in the thick of building your project and your team starts testing the builds, questions will come up that can't wait for the weekly check-in meeting without stalling progress. Establishing how these questions should be communicated (we've found Slack the easiest for everyone) and the expected response time (anything longer than one business day isn't acceptable in our book) will set the project up for a better success rate.
- **Project Status:** You should have access to the development team's project management tool to monitor feature development progress and bug status.

08 How involved will we be in the development process?

The most successful projects stem from a collaborative relationship, with both teams leaning on each other's expertise.

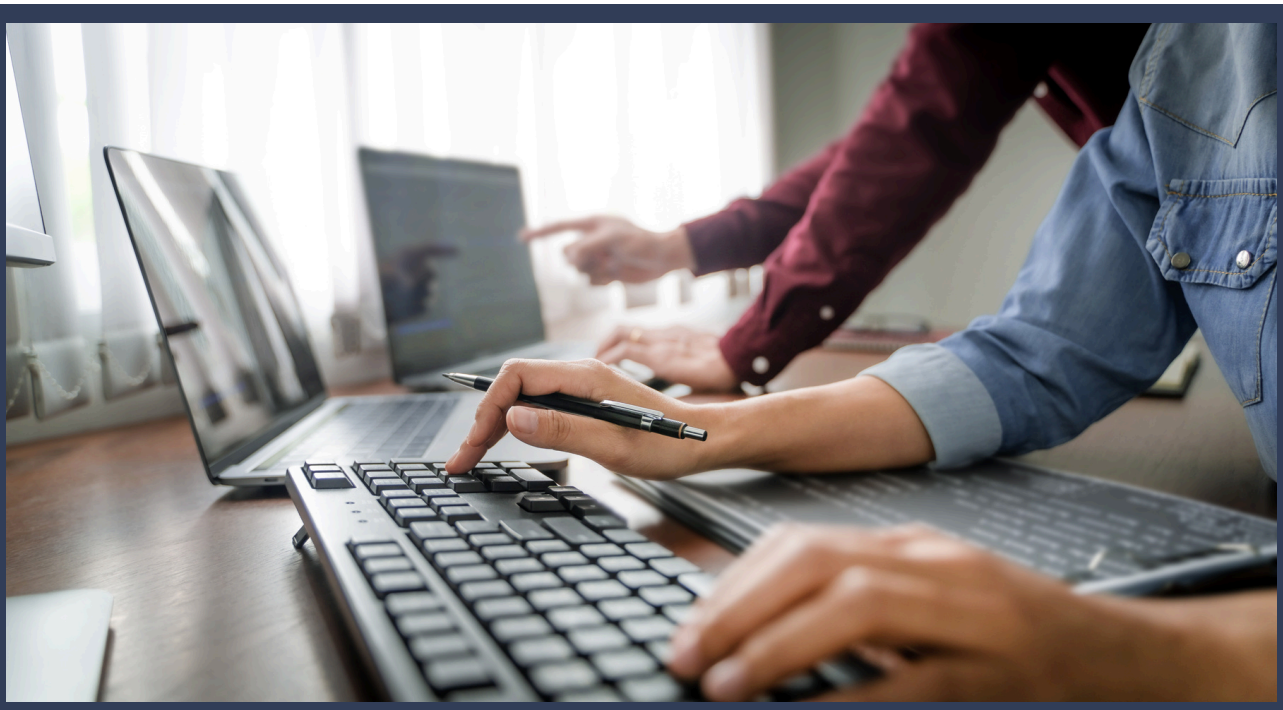
And that's the type of answer you want to receive.

Any team that answers they'll do anything asked of them, no questions asked, is what we call a "yes team." This development approach leads to over-featured products, missed timelines, and blown budgets.

Conversely, you don't want to work with a team that promises to contact you again when the product is finished.

The best scenario is a partnership built on trust between their team as the digital expert and your team as the subject matter expert.

09 How do you document your code and processes?



The best philosophy for documenting code is to leave enough comments so that anyone reading it for the first time can understand it.

> Simple, well-factored code with commenting is easy to follow, but clever code is often required for something to work correctly, and the thought process for creating that code should be included in the comments.

Software teams should mention they'll provide a "Readme" file that explains how to generate the environment, build the software, and deploy it. They should also provide documentation of user flows (diagrams of the path followed), stories (text descriptions of user paths), and test procedures.

10 Which development platforms do you use?

There are three potential answers to this question.

“We are experienced in every platform, so you’ll have your pick!”

Be wary of a “We can do anything!” answer.

Dozens (if not hundreds) of platforms exist, with new ones constantly popping up. They are continually evolving. Unless you work in them daily, you can’t become an expert dipping your toes here and there.

“We build everything in Kotlin because it’s the best!”

A single answer like this for a mobile app could be correct, depending on the context. For example, Kotlin is the most up-to-date and supported Android language for building an Android app.

There is no "right" answer for backends and web interfaces. The key to selecting these is tied more to what the platform is intended to do and the number of developers who can support it than to any 'one-size-fits-all' platform.

“We have depth in a handful of platforms.”

This is typically the answer you’re looking for, especially if you’re unsure which platform suits your project.

A good software development team will explain the pros and cons of various platforms and objectively answer questions about the task at hand.

11 How do you ensure my software will be bug-free?

A-ha, a trick question!

No digital product is guaranteed to be bug-free, and anyone who says otherwise is making a promise they can't keep.

That said, a good software development team strives to ship error-free software. They should answer this question by walking you through their testing procedures and discussing a long-term support plan.

12 How do you protect our software from getting hacked?

A-ha, a second trick question!

No digital product is guaranteed hack-proof and a good development team should walk you through its security protocols.

A solid starting approach to security is:

1

Don't collect data you don't need.
Some developers use analytic tools that ask for too much personal user information. The less data you own, the less risk you'll carry.

2

Encrypt data at rest and in transit.
When data is moving across the internet, use SSL to encrypt it. When stored on a server, use industry-standard encryption techniques to ensure sensitive data is safe.

13 How will you keep our software up to date?

Keeping software up to date requires a workforce. Time and attention are needed to maintain the functionality and security of your product, and a software development team should outline how often they check for dependency and operating system updates along with their process for taking in customer feedback to improve the user experience.

This circles back to the first sentence in this guide, which states that digital products are never done. Unless you have an internal team to support your product, your relationship with your developer should continue post-launch, and this question is an excellent opportunity to learn about the ongoing costs of supporting your digital product.

14 How do you deal with technical debt?

Technical debt happens when you take a development shortcut to get a product out the door quickly. No matter how hard you try to avoid technical debt, you are bound to run into it, especially if you take an MVP approach to developing software.

Clients typically want to see development efforts go towards new, marketable features. Nobody likes updating software with release notes that say, “Bug fixes and minor improvements.” However, these “minor improvements” keep software up-to-date and modern. If you don’t pay off technical debt early, like monetary debt, it compounds and results in more significant issues down the road.

Technical debt is a standard business issue. A true tech partner will communicate any technical debt they see and work with you to prioritize it among the other new features and improvements you want to make.

General Business Questions

15 What is your role in our business?

Before asking this question, consider the type of partnership your project needs.

Do you have a CTO and a few developers on staff and need a development team to fill in the skills gaps your team lacks?

OR

Do you need a collaborative partner to guide product development and offer strategic advice?

Setting these expectations at the forefront ensures everyone is on the same page about how integrated the software development team will be within your business.

16 Do you have project minimums?

Software development companies can have project minimums for many practical and strategic reasons, and the same can be said about companies without a minimum requirement.

There is no right or wrong answer here. Before getting too far down the road with a prospective software team, it is a good idea to know if your project's size and budget align with their clientele.

17 How much does it cost to work with you?

Software is notoriously difficult to price, and comparing pricing between software development companies can be confusing.

For instance, a company with rates of $50/hour could use overseas or junior developers and might need more time to build a project than an experienced development shop with a rate of $200/hour.

The industry also has a long way to go regarding pricing transparency. Development companies that are leading the charge have started listing pricing on their websites.

If pricing isn't publicly available, contacting them about it should be simple. A reputable company will discuss its pricing model without making you jump through hoops.

The hoops come around when it's time for a formal quote, which leads to the next pricing question.

18 How will you price my project?

Multiple conversations with a development team are essential for gathering the details required to build a quote. A development team needs to assess technical feasibility, resolve any ambiguities, and plan resources to ensure that the quote is accurate, comprehensive, and tailored to your project's specific needs and expectations.

During this process, you should feel a strong relationship building between their team and yours.

We'll break down some line items found on software development quotes. If any of the following items aren't on your project's quote, it's essential to find out why. Likely, it's because the item isn't related to your project, or it could be that the software development company doesn't provide that service.

Here are some line items found on software development quotes:

- **Project Management:** Costs related to planning, scheduling, and managing the overall project, along with regular updates, progress meetings, and status reports.
- **Requirement Analysis:** Detailed documentation of software requirements, sometimes called "Discovery."
- **Design:** Visual designs, mockups, and UI components, as well as UX research, wireframing, and prototyping.
- **Development:**
  - **Front-End Development:** Coding the user-facing part of the software, including the UI and user interactions.
  - **Back-End Development:** Server-side development, database design, and API integrations.
  - **Middleware Integration:** Connecting the front-end and back-end systems.
- **Testing and Quality Assurance (QA):**
  - **Unit Testing:** Testing individual components for functionality.
  - **Integration Testing:** Ensuring that components work together as expected.
  - **User Acceptance Testing (UAT):** Testing by end-users to validate requirements.
  - **Bug Fixing and Optimization:** Addressing issues and optimizing performance.
- **Deployment:** Preparing and submitting to app stores or other live environments and setting up beta testing environments for initial user feedback.
- **Maintenance and Support:** Ongoing support for a specified period post-launch, including costs for updates and fixing any bugs discovered after launch.
- **Third-Party and Subscription Services:** Integrating third-party APIs for additional functionalities and any costs for third-party services (e.g., payment gateways, analytics tools).
- **Licenses and Subscriptions:** Costs for any specific tools or software licenses required for development, server hosting, and maintenance.
- **Content Creation:** Creating copy, icons, logos, and multimedia content.
- **Legal and Compliance:** Drafting legal documents required for the software and ensuring it meets any legal and regulatory standards.
- **Training and Documentation:** Creating manuals and guides for end-users and administrators and providing training on how to use and manage the software.
- **Contingency Buffer:** A percentage of the total project cost to cover unexpected expenses.

19 Who owns the code when the project is finished?

Bottom line: You should own the code. All source code falls into one of three camps:

1. Code you brought to the table.

This should be self-explanatory. If you wrote code, you own that code.

2. Code written custom for your project.

Custom algorithms, configurations, proprietary algorithms, and business flows should be your property unless a Master Service Agreement (MSA) or Non-Disclosure Agreement (NDA) states otherwise.

3. Code that is provided by someone else (usually via open source)

Internal libraries

A digital product agency's primary value is delivering custom software quickly. One way they can do this is by reusing chunks of code that are standard components of most software.

Think of these code chunks as Lego bricks. Each brick is necessary for the entire project to function, but it's not like your red 4x2 Lego is different from the next guy's red 4x2 Lego brick.

In software terms, features like "profile picture camera" are like a Lego brick. Unless you build a custom camera app, the selfie camera view isn't proprietary to your app. You don't care whether the code is written from scratch or a framework is used. What you *do* care about, however, is that the selfie camera code is something you have the right to use and modify later on if you choose to do so.

When it comes to code like that, most development companies own the code and grant customers the right to take that code and do whatever they want with it. That way, both parties own the code and can continue to modify it without worrying about infringing on someone's rights.

External libraries

It would be difficult to find any software of even minor significance today that does not incorporate at least one piece of open-source software (OSS). Like the "profile picture camera" example above, open-source software is useful when it performs a task that isn't critical for your digital product to have written from scratch.

An excellent approach to OSS is to use it sparingly and in areas that don't need to be written from scratch.

20 How will you measure the success of our project?

Measuring success is different for each project.

Some projects are successful based on how much revenue they generate for the business. Others are based on the number of monthly active users.

Your development partner should work with you to identify the metrics you need to hit to call the project successful and discuss the necessary tracking tools for measuring them. This could mean adding a third-party analytics package like Kissmetrics or Firebase Analytics or creating a custom dashboard for tracking values.

Interview Sheet

Questions about the Team and Experience

1. Where are your developers located?
2. How many developers will be working on my project, and how skilled are they?
3. What projects are you working on now, and how will mine be prioritized?
4. What have you developed that we can test out?
5. Can you provide references?

Questions about the Development Process and Work Style

6. What does the design phase look like?
7. How will you keep us updated on progress?
8. How involved will we be in the development process?
9. How do you document your code and processes?
10. Which development platforms do you use?
11. How do you ensure my software will be bug-free?
12. How do you protect our software from getting hacked?
13. How will you keep our software up to date?
14. How do you deal with technical debt?

General Business Questions

15. What is your role in our business?
16. Do you have project minimums?
17. How much does it cost to work with you?
18. How will you price my project?
19. Who owns the code when the project is finished?
20. How will you measure the success of our project?

About the Creators of This Guide

JMG was formed during the early days of the App Store when having an app was the "cool new thing."

During that time, everyone wanted an app for the sake of having an app, but over a decade later, the market is much different. Every company is now a technology company, and businesses are leveraging digital tools like mobile apps and custom software in order to be agile, innovative, and responsive to market trends.

Since 2012, JMG has been the go-to partner for companies that recognize the need to improve their bottom line with technology. Our reputation is bound to the success of our clients, and we focus on meeting companies where they are to inform and facilitate the creation of the digital tools they need to grow their business.

This includes being an educational resource for the community. We aim to dispel the myth that you need an engineering degree to understand technology.

You can find our blog, podcast, newsletter, and other resources at https://jmg.mn.

The views reflected in this guide are based on the experiences of the JMG team. We recognize that the software industry is constantly changing and strive to provide accurate information with as much bias removed as possible. Feel free to reach out to us with any questions or feedback.

JMG

https://jmg.mn
info@jmg.mn
612-208-3872