

Software Development

# Buyer's Guide



# Contents

<b>Why should you read this guide?</b>	<b>2</b>
<b>The Triple Constraint: The Project Management Triangle</b>	<b>4</b>
• Customized Project Management	5
<b>Understanding the Significance of your Project</b>	<b>6</b>
• Defining Your Problem	7
• Creating an Elevator Pitch	8
• Do You Need a Minimum Viable Product (MVP)?	8
• Choosing a Supplier Who Understand Your User	8
<b>Selecting Your Supplier</b>	<b>9</b>
• Questions You Need to Ask	10
<b>Checklist for Purchasing Software</b>	<b>11</b>
<b>How to Prepare Accordingly for the Best Results</b>	<b>13</b>
• Forget the Plan	14
• Take it Step-by-Step	14
• What Goes on Behind the Scenes?	14
<b>Checklist for Best Software Development Practices</b>	<b>15</b>
<b>We're All in This Together</b>	<b>17</b>
• How to Build Sustainable Trust	18
<b>Thank you!</b>	<b>19</b>

A woman with glasses is sitting at a desk, looking thoughtful. She has her hand to her chin. In front of her is a laptop and some papers. The image has a dark blue overlay.

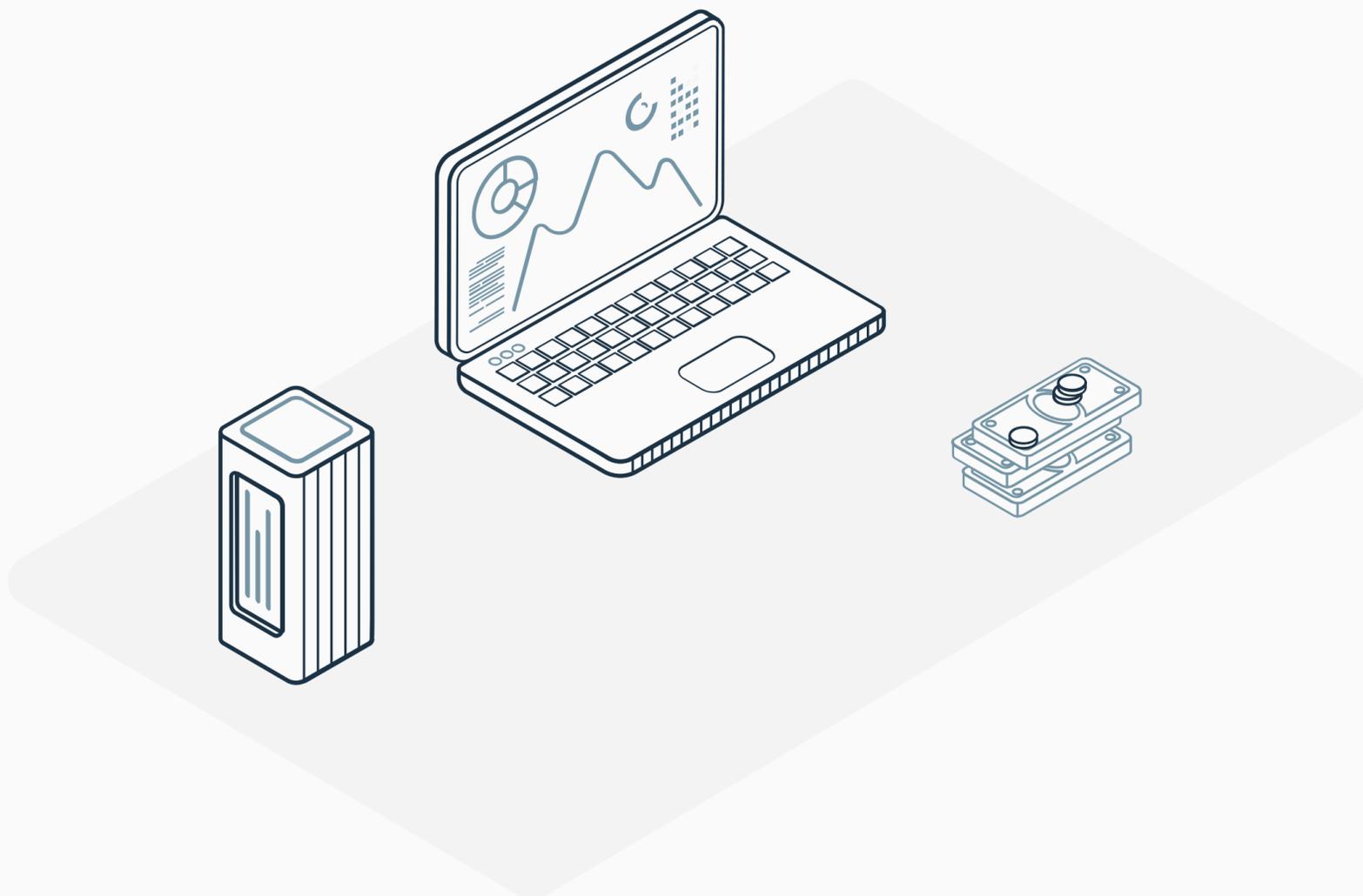
# Why Bother Reading?





This guide is for anyone planning a software development project. You don't need to be a mastermind to buy software development, but you need to know the basics. It's time to start purchasing software development from a value perspective (the intended use and feasibility of the system).

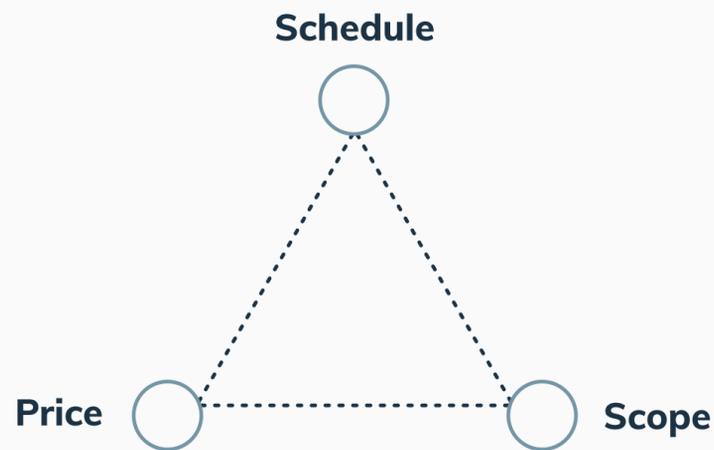
Ahead is our down-to-earth advice and practical, hands-on tools for getting your software project underway. Continue reading if you want to level up as a software development buyer.





# The Triple Constraint: **The Project Management Triangle**





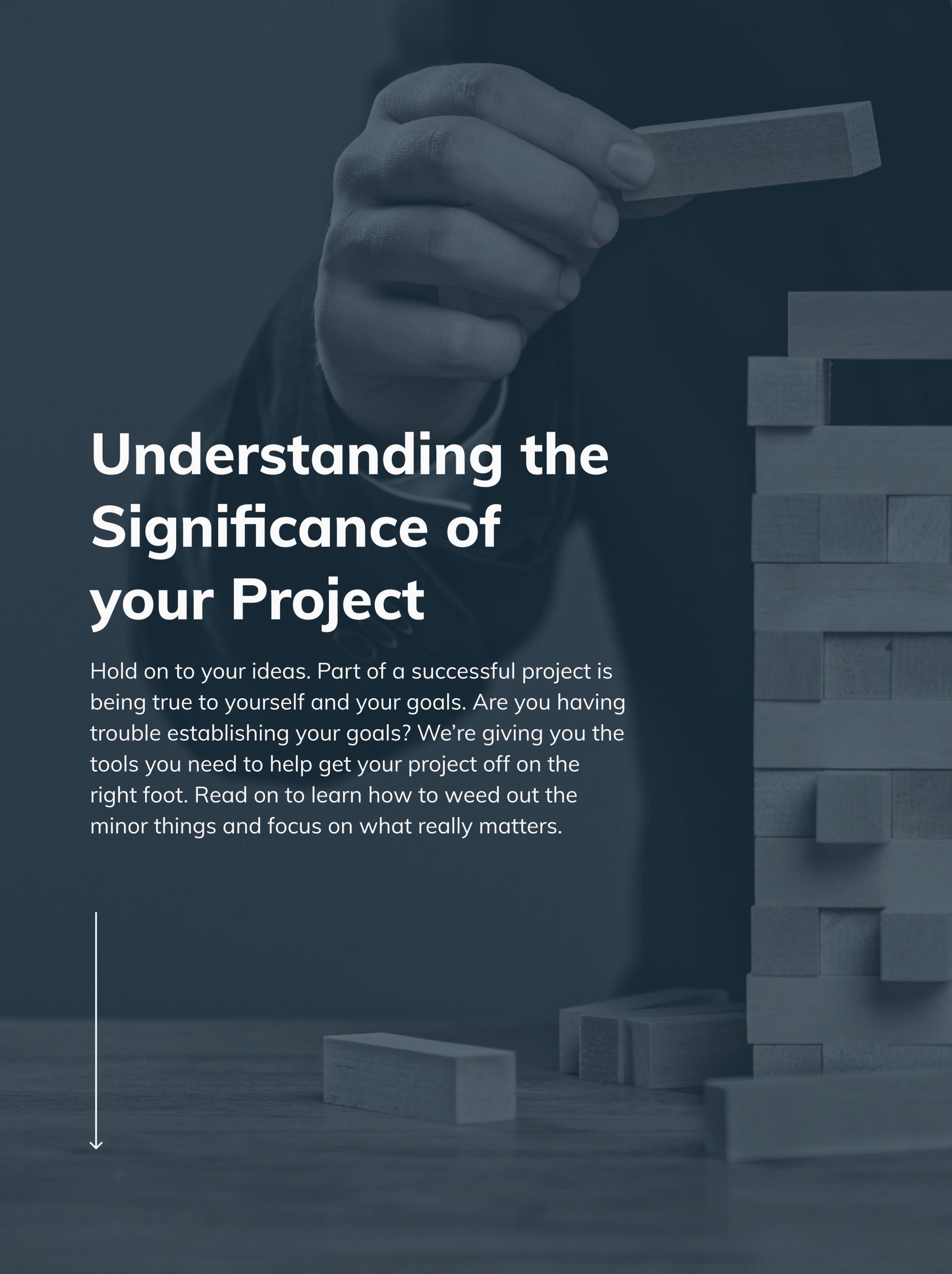
The project management “triangle” consisting of cost, schedule, and scope has been aiding project development ever since, well, the beginning of projects. Naturally, when purchasing software, people tend to look for all three factors. However, it’s nearly impossible to determine all three elements at the early stages of a project. Adhering to specific constraints from the get-go will put an investable crunch on your projects. Now, we’re not saying to forgo sticking to a schedule and budget completely, but we’re challenging you to think beyond the “triangle.”

The Project Triangle can’t recognize software quality or value as project goals. And we think it’s safe to say creating value is likely one of your goals. So instead of confining yourself to this in-the-box approach to project management, we encourage you to create a triangle that works for you.

## Customized Project Management

Rather than boxing your project’s success based on the amount of content, we think you should focus more on the value your project can bring to your business or brand. For example, a good project shouldn’t be categorized as “good” simply because it meets a specific feature quota or was completed under budget, or was done within a particular time frame. Honing on cost, schedule, and scope can impact the overall quality of the finished product. Why not create a triangle that works for you?

If you assess the success of a project by the value it can add, you’re more likely to have a better end result. So we’re giving you various tips and best practices to help ensure you choose a supplier who understands the value of your project while helping you plan and prepare for what to expect during the process.

A hand is shown holding a single wooden block above a tall, partially constructed Jenga tower. The scene is dimly lit, with the hand and the block being the primary focus. The tower is made of several layers of rectangular wooden blocks, with some blocks missing or slightly offset, suggesting a delicate balance. The background is dark and out of focus.

# Understanding the Significance of your Project

Hold on to your ideas. Part of a successful project is being true to yourself and your goals. Are you having trouble establishing your goals? We're giving you the tools you need to help get your project off on the right foot. Read on to learn how to weed out the minor things and focus on what really matters.





## Defining Your Problem:

So, what's your issue? Surely you're inquiring about software to create a positive change. But what are the problems you intend to solve? Figuring this out is the first step. Regardless of your niche, what is the need your product is addressing? After you have a clear-cut understanding of your product's intention, communicate this to your supplier. The best way to organize your thoughts and properly share them is with an elevator pitch.

## Creating an Elevator Pitch:

An elevator pitch is a short presentation of your idea or product explaining the concept in a clear, easy-to-understand way. A good pitch will come in handy when you initiate a conversation with a potential supplier. Trust us; this will save hours of your (and your suppliers) time. Simply handing over a list of required features and the desired completion date isn't the most effective way to open communication with your potential supplier. Instead, why not start with an elevator pitch to give the supplier critical information from the beginning?

This way, the supplier can assess their capabilities and schedule to determine if they can meet your needs. A good elevator pitch answers these key questions:

1. Who is your target audience?
2. What issues or needs are your target audience experiencing?
3. What issues or needs do your project intend to address?
4. What category does your project belong to?
5. How does it benefit its users?
6. What makes it extraordinary compared to similar products or solutions on the market?
7. What are the key differences between your product and others?
8. Why should users use your product over others?



## Tips to Keep in Mind:

An elevator pitch should last no longer than 20-30 seconds (a short elevator ride). Make your pitch interesting and succinct. And always remember to showcase the value your product will offer clearly.

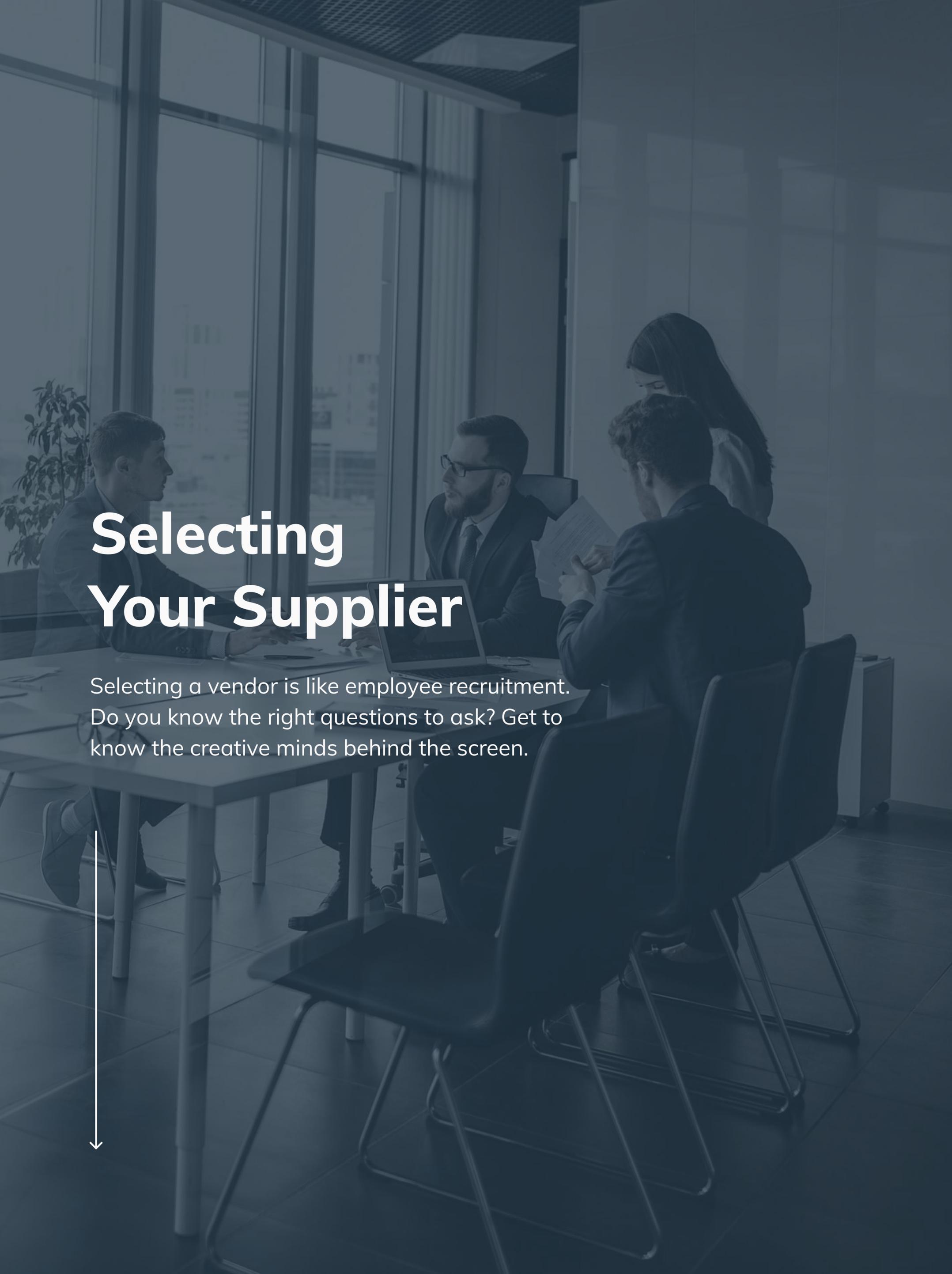
## Do You Need a Minimum Viable Product (MVP)?

Absolutely. With an MVP, your product will be developed with the bare minimum features to validate the given issue and the solution your product strives to resolve. MVPs collect (and learn from) data, analytics, and feedback (positive or negative) from real users. Plenty of the most successful software projects begin with an MVP phase. So, you might want to check if your chosen agency implements an MVP functionality. A well-versed developer should be able to craft an MVP using your elevator pitch.

## Choosing a Supplier Who Understands Your User

Just because a supplier has experienced coders doesn't mean your project will be successful. Working with a developer who wants to understand your audience is often more valuable than having a world-class development team. Users demand digital services that are fast, visually appealing, and easy to use. Make sure your provider uses both service designers and UX (user experience) designers.





# Selecting Your Supplier

Selecting a vendor is like employee recruitment. Do you know the right questions to ask? Get to know the creative minds behind the screen.





## Questions You Need to Ask

1. Ask for examples of products or services they've developed.
2. Ask if they have any experience completing a similar project to yours.
3. Ask how long projects similar to yours usually take.
4. Ask who the supplier works with. Do they have an in-house team, or do they outsource services?
5. Understand if they charge based on hourly rates or fixed prices.

”

**As a client, it's worth your while to get a feel of the supplier organization, up close and personal.**

A hand holding a pen over a laptop keyboard, with a checklist document visible in the foreground. The background is a blurred office setting.

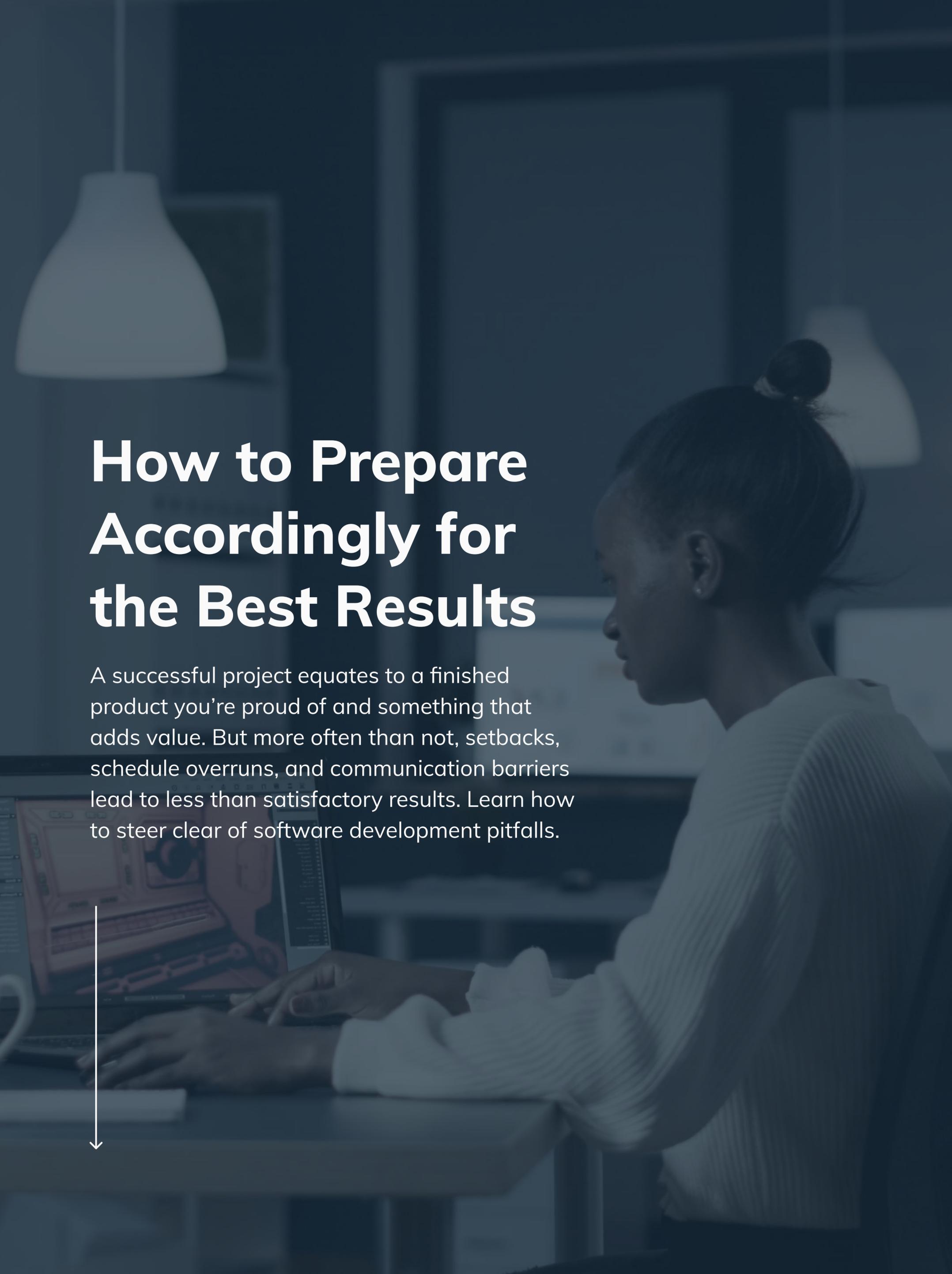
# Checklist for Purchasing Software

Use the checklist below to help guarantee you receive exactly what you ordered. The following tips will help you establish the proper framework for your project.





- ✓ Why are you seeking the help of professional software developers in the first place? First, define the issue your software aims to solve and define your goals.
- ✓ Have a timeline. Tenders need to know your preferred schedule to determine if they can take on the job.
- ✓ Have a rough estimate of your budget or your investment limit.
- Leave room for creative collaborations. While you may be dead set on something, things are updated frequently in the world of software development. So don't get too hung up on the planning from the beginning.
- You have the right to require transparent workload estimates. Look into the smallest and largest estimates - and keep in mind these are all assumption-based.
- If you can, meet and interview your developers in person. That way, you can get a better understanding of their capabilities and their personality, work ethic, and passion for their craft. There's nothing worse than an unmotivated software developer!
- Ask for examples of previous projects. That way, you can get a genuine feel for how the supplier works.
- Be sure your supplier can stick to a timeline. An excellent way to ensure you won't be left high and dry is by asking your service provider to confirm they can deliver fully finished functions regularly.
- Ask your supplier about their quality assurance. For example, what methods do they use to ensure quality?
- Ask about the warranty policy. Bugs happen!
- Demand ownership of the IPR and the source code. If you don't demand ownership, you won't be able to further develop the software with other suppliers in the future.
- Secure guarantees that you are free to use all open source code components in potential future business models.

A woman with her hair in a bun is seen in profile, working on a laptop in a dimly lit office. The background is blurred, showing a desk lamp and office equipment. The overall tone is professional and focused.

# How to Prepare Accordingly for the Best Results

A successful project equates to a finished product you're proud of and something that adds value. But more often than not, setbacks, schedule overruns, and communication barriers lead to less than satisfactory results. Learn how to steer clear of software development pitfalls.





## Forget the Plan

We know this sounds a little counterproductive. Any good project begins with a plan, right? But if the past has taught us anything, let it be that the only constant in software development is changing. When you're buying software, plans changing are almost as inevitable as it raining in London in April. Initial plans don't stand the test of time in this field. Instead, prepare yourself to keep on planning throughout the project. Changing the agenda as the project progresses is all the more valuable because you have actual data to base your decisions on.

## Take it Step-by-Step

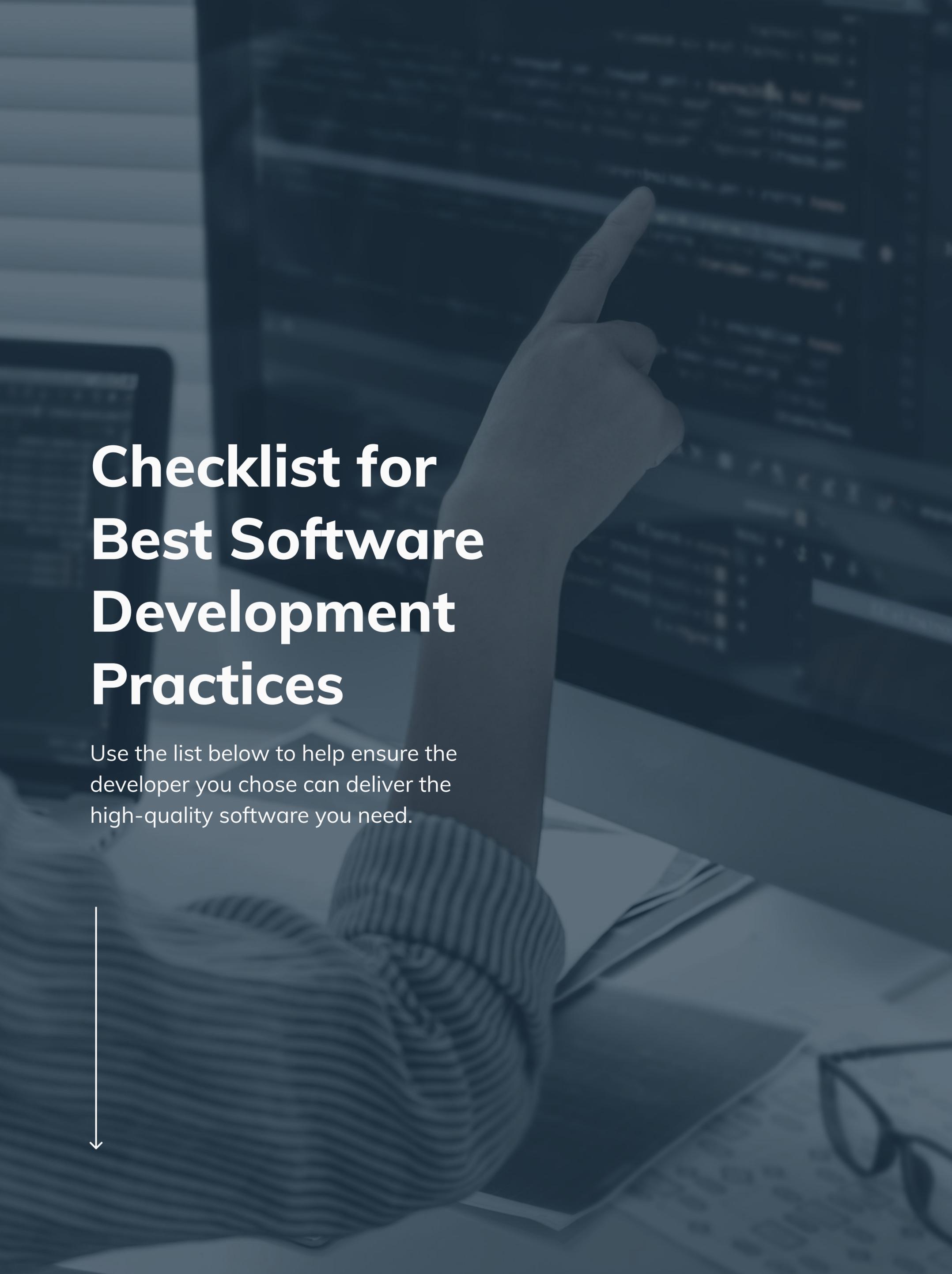
Usually, the software is developed one component at a time. For example, the database is often the first step and acts as a foundation to build on. However, we don't think that's the most effective way to go about software development.

We encourage you to work with a supplier that produces one feature at a time, including both the back-end and front-end implementation of that feature. Another reason to take things step by step is that you (the buyer) can

provide more concise feedback during the development phases, rather than waiting for the later phases of the project. In most cases, your supplier will have to make changes to the finished components based on your feedback. The longer you wait, the more likely the developer is already working on another project - ultimately leaving your project toward the end of their priority list.

## Take it Step-by-Step

No one expects you to be well-versed in software development language, but it is helpful to have a general idea of what happens in the front-end and the back-end. Front-end refers to the parts of the software that you can see on end-user devices like web pages and mobile apps. Back-end refers to the database, the service's business logic, and integrations to other systems. The front-end communicates with the back-end. All web applications require a server of some sort. Servers can be in a server room or on the cloud. Ask your supplier to help guide you when choosing the best service options. They should have enough insight on production support and service monitoring to guide you in the right direction. (If they don't, that's a pretty significant red flag)



# Checklist for Best Software Development Practices

Use the list below to help ensure the developer you chose can deliver the high-quality software you need.





- ✓ **A backlog** helps software developers organize and prioritize any given project. You want to ensure the developer has a clear understanding of the most critical aspects of your project and can deliver the task within a specified time. In addition, the backlog serves as a shared understanding between you and the developer of what the project will entail and what the priorities are.
- ✓ **Version control** allows developers to document changes made to the source code. It comes in handy when maintaining several development branches simultaneously.
- ✓ **Code reviews** are critical. They help reduce defects. Developers should review all lines of code continuously.
- **Unit testing** validates that each unit of the software code performs as expected. Unit testing is essential for quality control.
- **Integration testing** tests software modules as a group to ensure they work properly together.
- **Corridor testing** is when a developer will ask a colleague to test the product and offer their feedback.
- **User experience testing** tries the system on real users. User experience feedback determines the best way for a system to interact with the target audience.
- **Continuous Integration (CI)** requires developers to integrate code into a shared repository throughout the day after a change. The whole software is tested after each new change to the system to detect bugs immediately.
- **Continuous improvement** involves regular meetings to review the project, implement changes, and make improvements to create a better end result.
- **Regular deliveries** will help you stay on track with the progress of your project. Regular deliveries at short intervals allow you to check in with the developer and offer your feedback. They also help the developer learn more about your needs and requirements.

# We're All in This Together

Trust, transparency, good vibes, and positive interactions generate successful collaborations. Keep communication channels open before, during, and after the project.

As software developers, we come across all different types of clients. Some are less personally invested in their project, while others nurture it like a newborn child. But regardless, a lack of trust leads to futile work.

We've found that an excellent way to increase your project's chances of success is to select developers with a passion for openness and transparency.



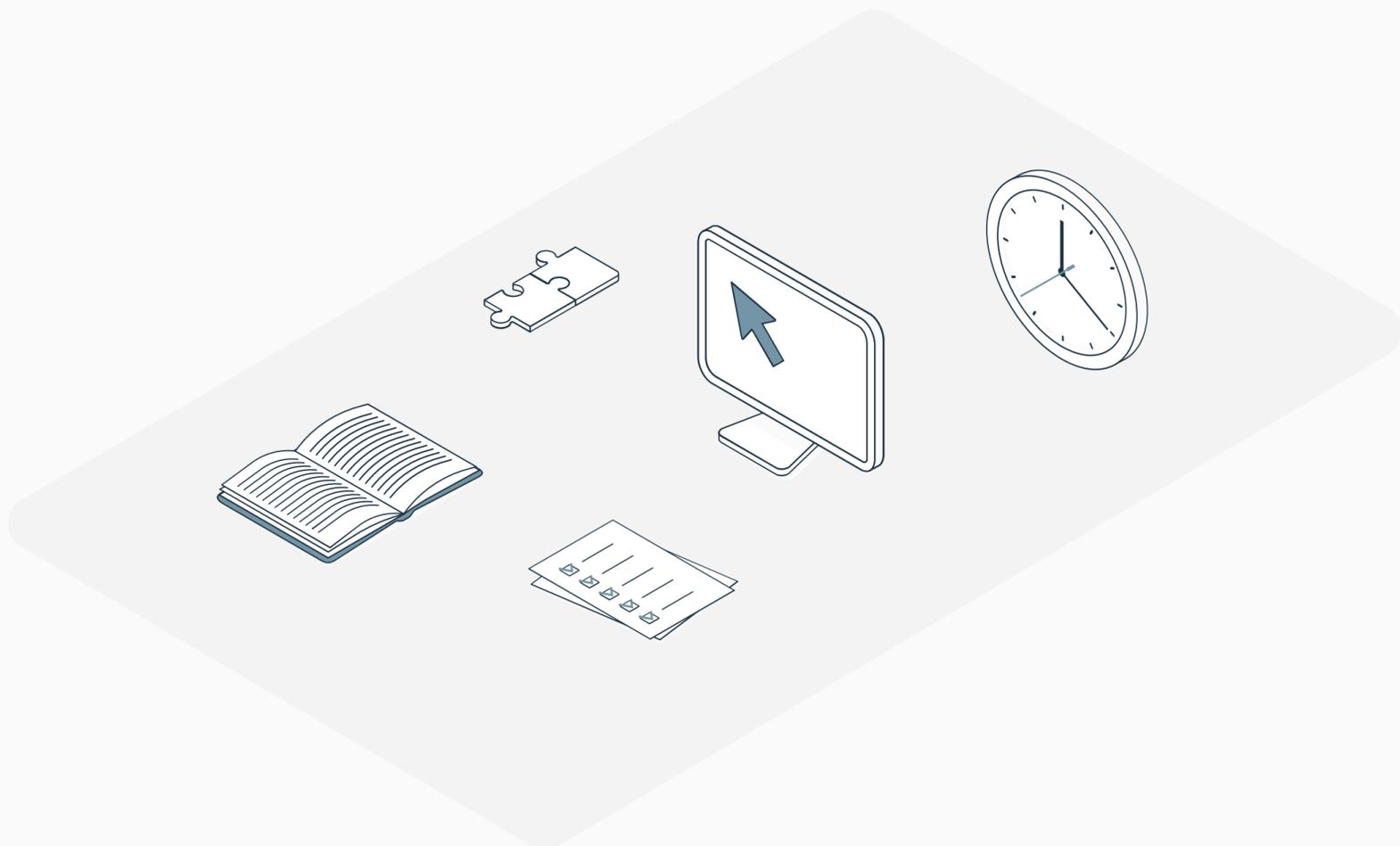


## How to Build Sustainable Trust

The best bonds between a client and a supplier are formed during face-to-face interactions. However, with today's current climate, in-person meetings are becoming more of a novelty than a reality. If you can meet in person, we highly recommend it but if it's not feasible,

be sure to use electronic devices to make communications easier.

Check in often with your supplier, set up regular face-to-face (or face-to-screen) meetings to keep the lines of communication as open as possible. This will help avoid any snafus down the road.





# Thank you!

On behalf of the team at Digital Solutions, we would like to thank you for your time and attention. We hope this buyer's guide inspired you, taught you something new, and provided practical tools to approach your next software development purchase from a value perspective.

Feedback is always welcome.

Please feel free to contact us:

**[hello@digitalsolutions.dev](mailto:hello@digitalsolutions.dev)**

**(949) 379-1714**