



# Table of content

## 01

### Choose the Type of Your Partner

Software House vs. In-house vs. Freelancer .....

## 02

### Start an Effective Research and Investigation

What Is a Discovery Call? ..... 07-09

Discovery Call - What Question You Should Expect ..... 10-12

## 03

### Make the Right Choice

How To Choose a Software House For Your Next Project ..... 14-16

How IT Project Estimation Should Look Like ..... 17-21

Software House Proposal - What You Can Expect ..... 22-24

## 04

### Create Safe Agreement

What Software Development Contract Should Include ..... 26-28

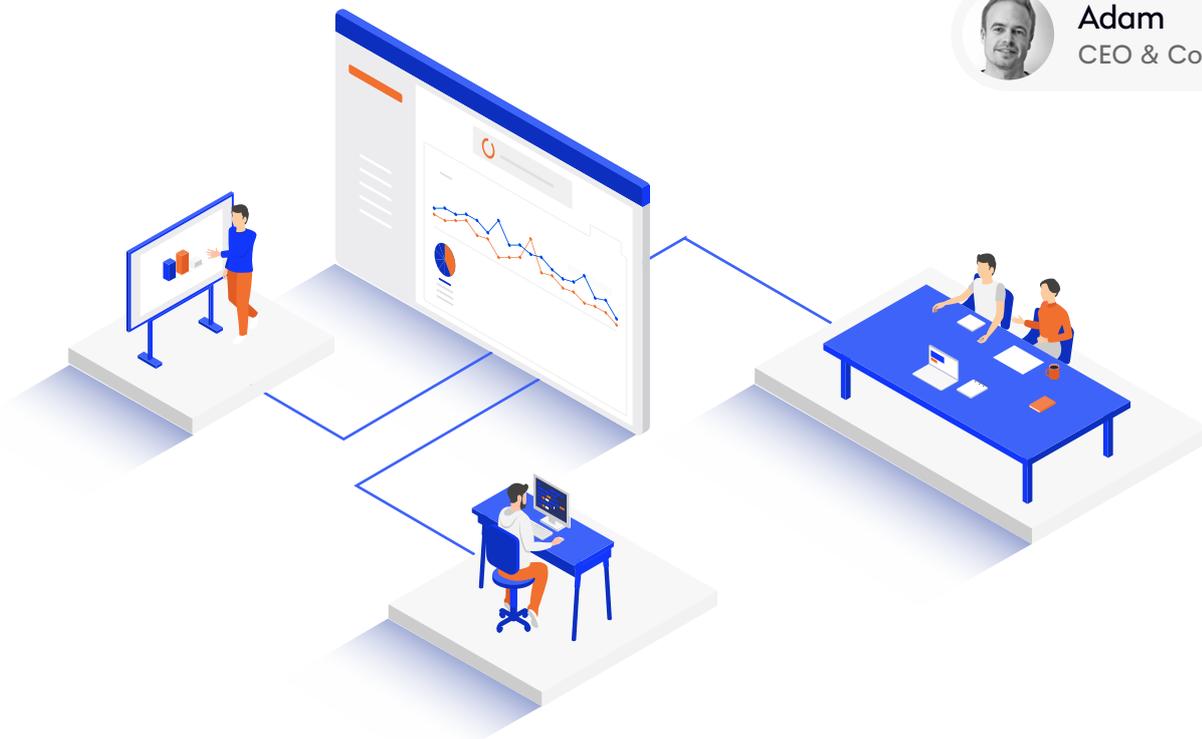
## 05

### Prepare For the First Kickoff Meeting

How to Conduct a Perfect Kickoff Meeting ..... 30-32

# 01

Choose the Type of Your Partner



# Software House vs. In-house vs. Freelancer

Looking for the perfect team for your next IT project? Is it better to outsource or build in house? What to consider before making a choice? Learn the pros and cons of working with freelancers, a software house, or building a project in house.

You have a product idea and a plan for development, so it's time to find the perfect team that will build your project. But which one to choose: a freelancer, a software house, or the in-house team?

Different projects require a completely different approach, so you need to determine the scope of the project before you choose a solution tailored to your needs. Learn the pros and cons of each solution and make a wise decision.

## Working with a freelancer

- + Hiring a freelancer works well for small projects, but only if you don't plan to develop them in the future. So it's better to hire a freelancer for one-off projects, such as making a landing page or a simple static site.



- + Moreover, it's relatively easy to find freelancers and **they can usually start right away**. They can also quickly finish a certain project and charge less than a software house or a full-time employee.
- Once a project is finished, the freelancer's job is done. **If you expect maintenance, bug fixing, or support, you will need to handle them yourself**. Even if you find a full-stack developer, it's hard to expect one person to cover all elements, such as backend, frontend, QA, UX/UI design, etc. And managing several freelancers can be tricky.
- Plus, freelance developers sometimes work beyond working hours, which can result in poor communication. They usually have more than one client, so they often work on multiple projects simultaneously. **This means less commitment to a project** and when they receive a better offer, they might drop out in the middle of the work.

## Perhaps you should hire a software house

- + When you need a full package of services managed by an experienced team and want to develop a product from scratch, a software house is the answer. Such companies put huge effort into completing **teams that consist of a vast range of experts who specialize in certain technologies and industries** and provide quality products and security. They are also familiar with up-to-date technology and trends.
- + You should consider working with a software house **if you want to build a bigger app (with a bigger budget) with complex logic** or if you run a large offline business that needs IT systems to improve its performance.
- + It's also a good idea for companies that build online products, have their own teams but with a limited processing capacity, or if you need to scale the team for some time with highly skilled people.
- One of the biggest concerns that you might have when outsourcing your project to a software house could be time zone differences, but it's often solved with the right tools and strong communication skills.
- Another thing is that working with a software house comes at a **higher price than working with freelancers**, but you get higher quality based on years of experience. Plus, sometimes you **have to wait several weeks to start a project**, but when you choose one of the market leaders, it's worth waiting.



## In-house team



- +** Building a product with in-house developers could be the best decision for companies that build complex IT products. Especially **if the project requires processing confidential data and strict access control**. Outsourcing such projects can be risky and it's more secure to keep it in house even if it's more expensive.
- +** In-house development is also a **good idea for large technology companies that offer products with commercial success**. In such a case it's worth investing in building your own team.
- +** The in-house team is familiar with the product, thus can smoothly run long-term projects with complicated use cases. They can also make real-time fixes and react as soon as anything happens.
- If your team is small, you may need to hire more people. But, **assembling a new team can be costly and time-consuming**, as it's not easy to find the best specialists. Make sure you can afford it.
- Costs related to hiring developers are not only about paying salaries, but you also need to provide equipment, tools, office space, employee benefits, etc. **Add the costs of staff turnover, onboarding and training, cover for sickness, raises, dismissal management, and many more...** It really costs money.

## Over to you

It's time to decide.



If it's a small project with minor upgrades that don't require maintenance and there's no strict deadline on the horizon, hiring a freelancer should be a good idea (and will most likely be cheaper).



If you need a bigger app with a complex logic that requires project management, think about a software house or building a project in house. Consider your team's skills and make sure it will have time and resources to complete a project. If not, a software house would be the best option. See our services and feel free to contact us!

# 02

Start an Effective Research  
and Investigation



# What Is a Discovery Call?

Your business needs a web application, so you've spent the time to find a software house that you consider a good fit. But how can you be sure that the vendor will understand your business needs? The answer is a discovery call.

As a client you have hopes that the software house will help you achieve your goals. Naturally, before you sign any contract you want to make sure that the vendor will meet your business needs, objectives, challenges, budget, deadline, and many more things. Is it possible to determine all the things during one call?

It is, but only when it's more than just the first conversation. And it is hidden under the term discovery call.

## What is a discovery call?

A discovery call can be the most important conversation, as you only get one chance to make a first impression. Both you and the software house can assess whether you're a good fit for each other.

So, what should the perfect discovery call consist of?



### Getting to know each other

First things first, it's time to learn something about what you do. **Help the software house understand where your company is now and what you want to achieve with their know-how and experience.**

Bearing this in mind, expect questions that will help the software house to know your company inside out. The more precise you are, the easier it will be to take concrete steps toward your goals.



## Understanding your concerns

Understanding your problems and requirements is what helps to see how the software house can potentially help. **Addressing top priorities, the biggest concerns, and main challenges before starting the collaboration gives the software house a broader perspective on your project** so that they can estimate potential risks and evaluate expected benefits.

By asking probing questions a software house gets the full scope of your hopes that their solution will help you achieve. This is why you should give much longer answers than “yes” or “no”. Open questions bring more value and success depends on the quality of both the questions and the answers.

If you both agree that you are a good fit for each other, then you can discuss the next steps.



## Addressing your needs

Another step is discussing how a software house can help you address each of the pain points you’ve mentioned before. Now it’s time to determine what the software house can do to solve your problems and meet your expectations.

Based on the discovery call, the software house aligns its services to your needs, so you’ll be assured that they’ll work smarter and be able to address your pains.



## Moving onwards

Expect the vendor to guide you through the next steps, help you evaluate the project, and discuss the general terms of collaboration.

After this part, you should know the details of the product based on the extra information that you provided to the software house.

## What is a discovery call for?

The purpose of the discovery call is to see whether you and the software house share goals.

Speaking on the phone or having a video call is far more effective than email correspondence. It’s an opportunity for you to verify the software house and talk about prospective collaboration.

You can ask for a relevant use case if you want to make sure that the vendor has experience with similar projects. The same goes for the industries and markets the software house operates in, the technologies they use, etc.

Perhaps you're looking for a company that will provide you with comprehensive services and manage your complex business scenario that requires both web and mobile applications and maintenance. You need to make sure that they have solid experience in applying leading technologies to be a reliable technological partner.

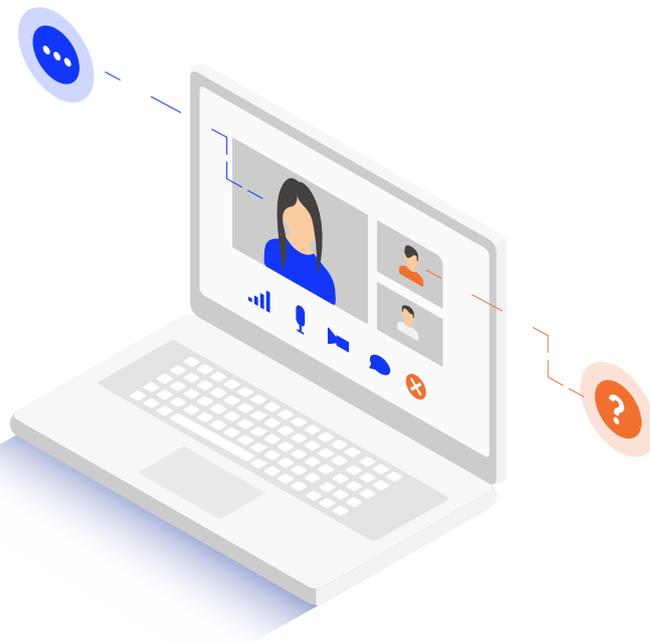
What can also be important is the methodology their work is based on. For instance, when a software house operates based on the Agile approach, it's information for you that you'll be actively involved in the process. So ask about the work culture, technology, and approach to remote collaboration.

It also helps to verify the quality of communication between you and the dev team. After all, you're going to work together for at least several months, so effective communication is what will make it smooth.

## A discovery call is a two-way conversation

As you can see, the discovery call is an opportunity to collect actionable data for both you and the software house. You can verify whether the software house you consider has trusted developers, expertise in sophisticated technology, and a solid background.

Even though it's better when it's more like a friendly conversation than an official call, it's good to be prepared. Take time to gather all the information that can help a software house to better understand your company with all its challenges and objectives.



# Discovery Call - What Question You Should Expect

A discovery call is an in-depth conversation, so it requires asking many questions to collect valuable insights. Hence honest and detailed answers really matter. Read on to see the questions a software house will ask during a discovery call.

We've explained what the discovery call is and how it can help you create a successful web app, so it's time to learn more about the process. You know that a well conducted discovery call leads a software house to understanding your concerns and challenges, so now it's time to dive deeper into the questions that you can hear during the call.

## What are you looking for?

A software house needs to understand your concerns and obstacles or challenges you experience to uncover the real problems. It's also important how you're addressing the problem today. Focus on the problems you're trying to solve and what has driven you to find the solution.

The more details you share, the better.

## Problems and pain points

A software house needs to understand your concerns and obstacles or challenges you experience to uncover the real problems. It's also important **how you're addressing the problem today**. Focus on the problems you're trying to solve and what has driven you to find the solution.



With an answer to this question a software house can determine whether it can help you reach goals. Perhaps you have in mind a product that seems to be ideal, but, in reality, needs many improvements that you're not aware of.

## Understanding your competitors

I guess you've made the industry and competitive analysis, so you know exactly who you compete with and how you can compare with them. It's also about the companies that could become competitors.

Tell a software house everything you can about the main competitors, why you think they do their job well or what they do wrong. You should also determine in what way they are similar to your company and **what you can learn from them**.

## What's your experience?

Another thing that is equally important is whether you've worked with a similar vendor before. If yes, how did it go? What was the project about? Which methodologies were used? What worked best? Did you like it? Did the collaboration go as planned? And what are the main pain points that you want to fix?



Armed with this knowledge we will get a better grasp of the business situation, so it will help us determine whether we fit your vision.

## What motivates you to choose a vendor?

It's no secret that you consider several vendors to work with. This is why one of our questions can be about what's most important to you when choosing a software house for your project.

Some of the things that you should take into consideration are a portfolio (does the vendor have experience with similar projects?), price, technology used by the software house, time to deliver, and much more. **All of these factors influence your decision making process.**

## Defining your internal resources

Knowing all the capabilities within your business makes you aware of your company's inner strengths and weaknesses. When you ask a software house for help, it's good to have a technical person on the team, as there will be important product decisions to make. And what can help to make the collaboration smooth is having a product owner on board — someone whose decisions are respected by your organization.



It's also good to know who else on your team will be involved in the project. All the information will help us to decide whether we have the resources to fill the gap.

Generally, a software house wants to know your **commitment to the project**.

## Setting a realistic timeline of goals

When you set your product plans, you possibly know **when you would like to be up and running with a solution**. So what are your expectations? How long should it take to build a test product or its live version? What about the launch? Always be realistic here.



We can then create a roadmap of the project together and set a timeline and deadline.

## How best to use your budget?

It's time to talk about the budget. You'll be asked about how much you can spend on the project. But note that it's not about how much money it will take to finish the project. A **software house needs to know your key cost constraints to help them to estimate the cost and plan the budget**.



Perhaps the budget has been the main blocker to get the project done before. Probing questions help uncover the real problems.

## Next steps — what do you need from a software house?

It's time to identify what's your main goal of the collaboration. At this stage **it's time to take a proactive approach** and suggest the next steps clearly.

## Find the right fit on the discovery call

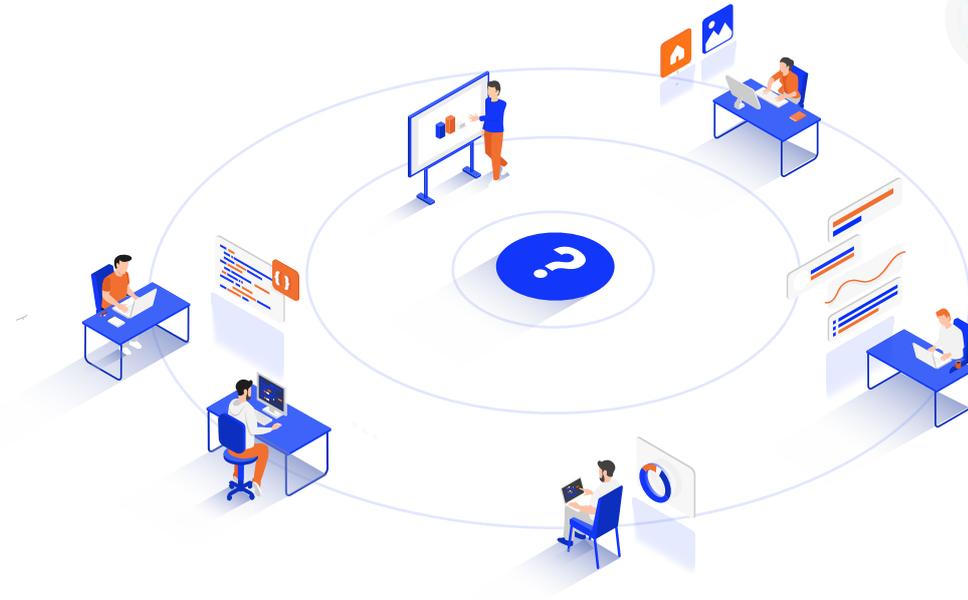
Questions asked during a discovery call are focused 100% on your needs. It's important for a vendor to identify the most critical goals, the main roadblocks, and to understand the business drivers.

You may feel a little overwhelmed by the number of questions you will face and the details that you need to share, so it's better to prepare. However, remember that it's important to create a product that fully fits your business needs. It's also an indicator of how professional is the company that you plan to work with.

The more detailed questions you get from them, the more engaged they are to create what you want. **All that sets you up for success** and helps to establish realistic expectations.

03

Make the Right Choice



# How To Choose a Software House For Your Next Project

Choosing a software development company can be a challenging decision. It should be preceded by thorough research, as it impacts the final effects of your IT product. Here's how to choose a software house for your next project.

There are some major things to consider before choosing a software house that will be the right fit for your business. It's crucial to make a well-thought-out decision, as you're going to work with a vendor for at least several months.

Read on to learn more about how to choose a software house that will make your project a success.

## A software house that understands your business needs

First, you need to specify your needs, determine your business goals, and set a realistic plan for product launch. Is your project web, mobile, or custom? Does it require a small team or a large group of developers? What technology is needed?

The more details you share with a software house, the easier it will be to decide whether a project team can answer your needs.

If you know what technologies your project requires, you can choose a vendor based on their technology stack. However, if you're not familiar with technology, a decent software development company might help you choose the best technology for your app idea.

Find a vendor that will make an effort to understand your current market situation, business goals, and objectives. You need a software development partner that will meet your expectations in every aspect concerning a project.

## Transparency at the forefront

A trustworthy company will provide you with fair answers to all your questions. **Top companies know when to say 'No' to a potential client**, as they are aware of their limits and their main goal is a successful product launch to grow your business performance.

Expect them to inform you about any issues or problems that may arise — this proves that a vendor is honest with you. You should be kept in the loop at all times.

A software house should explain the costs included in its project cost estimation, so you know exactly what you pay for. But, even if you have a limited budget, focus on the real cost. **Keep in mind that the cheapest option doesn't mean the best** and the cheapest offer may mean ending up paying even 2-3 times more for the final project.

---

🔗 When a software development company you're considering working with prepares a cost estimation in one day without thorough research, it should set alarm bells ringing. It needs to be based on a deep analysis of the project and many detailed questions asked by a project team.

---

## A decent onboarding

How will your cooperation start? Does a software house organize workshops for your team? Perhaps the company will guide you on how to prioritize tasks and how to use project management tools.

It's important to find a software house that will explain to you the differences between the methodologies used to create your project, help you understand the entire process, and give you clear guidelines.

You need to know how your project is going to be handled — **the more details a software house shares with you, the better control you have over the IT project**. Not to mention you'll feel more comfortable working with them.

Moreover, ask for a software house's approach to remote work, project management, and communication to ensure the work will go smoothly.

## Check their experience and clients' reviews

It's no question that you're more likely to trust someone with a proven track record and expertise. This is why reliable software development companies keep examples of their previous work on their websites.

**Go to the portfolio section and look for similar applications.** If the development team has experience in such projects, there are bigger chances that a company will better understand your business goals.

**Check testimonials on a software house website and look for more information on review sites,** such as Clutch (you can see our reviews here), where you can find the company's references, reviews, clients, and experience. It will give you broader knowledge of what they can do.

## How to choose a software house? There's even more

There are some other indicators that can help you differentiate trustworthy software development companies from bad ones.

Pay attention to whether a vendor asks a lot of questions. Of course, the number and complexity of questions may depend on your requirements. Note that **the more questions a software house asks, the more engaged it is.**

What can also help you in making a decision is whether a software house offers app maintenance and technical support, besides delivering the final product.

There you have it. When you base your investigation on given pieces of advice, you'll be able to select the right partner for your project that has experience in developing apps similar to your idea.

Good luck!



# How IT Project Estimation Should Look Like

Quoting the final price of a project that doesn't exist sounds like trying to predict the future. So how do you know that the IT project cost estimation is done the right way?

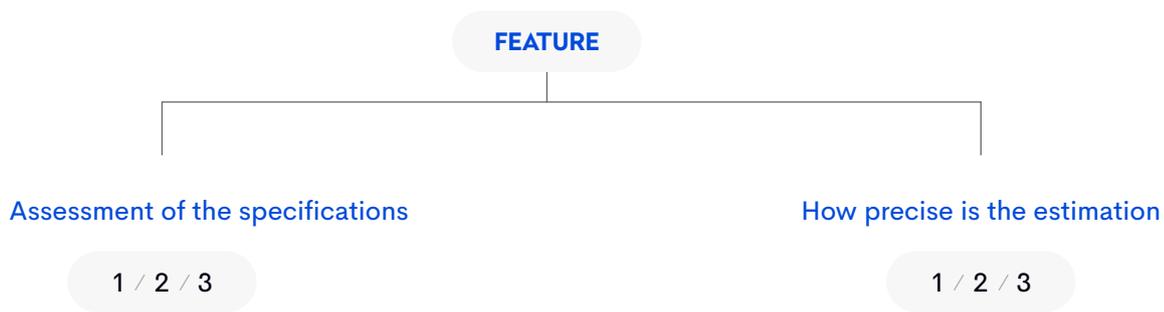
As project cost estimation comes with risks that might bring unexpected costs, it needs to be based on deep analysis of the project, related data, technology, team velocity, and many other factors.

There are various techniques for calculating the cost of software development. This article will show you how we do it at Studio Software.

## The accuracy of project cost estimation

The more details we have from you, the easier it is to calculate your software development costs and hours. Our experience shows that clients usually aren't clear about the requirements when they ask us for an estimation of the cost of their projects.

We draw a quotation based on our framework, but besides standard indicators we also include two things that matter for precise estimation



The first one is an **assessment of the specifications you deliver to us**. We use a 3-point scale. 3 is for top project specifications that make us fully understand your problems, needs, and expectations, while 1 is for specifications that lack details.

Another one is **prediction of how precise and accurate is the project estimation**. Here we also use a scale from 1 to 3 and the outcome is based on what you provided and how complex is the project. Then we operate based on our experience, so we take previous similar projects with their cost and time data to estimate your project.

And what about projects that we've never done before? We estimate the features based on our expertise, define the scope of work, and the technology that will have to be used. Then we can estimate the time we need to build such solution.

Our figures come with notes, so you always know why they have a low or high rating. Overall, it all depends on the quality of information we have from you, so we need fully outlined requirements for more precise estimates. **If we get napkin notes, you can't expect exact pricing.**

So before we create any cost estimate and detailed budget we organize a discovery call to gather key project information. Specific questions help us get deeper knowledge of your project's scope, timeframe, and expected outcome. questions help us get deeper knowledge of

Backend and frontend developers take part in the entire valuation process (as they can evaluate accurately based on their experience), and then they ask another technical person to verify their estimations.

When we're provided with all the details, we can draw up final estimates. This is why we use an Excel file to keep it all crystal clear so that you can see what the exact numbers mean.

# Cost estimation – what to expect from your potential technology partner

It's worth mentioning that you should expect a ballpark estimate rather than an accurate final price. It's because most of the features can be interpreted in various ways, especially at the beginning when the project details are only on paper.

In-house development is also a good idea for large technology companies that offer products with commercial success. In such a case it's worth investing in building your own team.

---

 **The process is called estimation, not exactimation.**

Phillip G. Armour

---

It's about predicting the resources, so these are not only the technology costs, but also time and people. Many other factors affect the final price as well, such as the time for architecture planning, bug fixing, project management, meetings, etc., so you need to determine whether the estimation is to include the following things:



## Manual or automated unit testing

You need to agree with a software house what % of total project time should be allocated to QA.



## Project management

Is it included in your project? Note that it usually takes 10-15% of the budget.



## Team meetings

They are important to keep everyone involved in the project on the same page. It takes time to set all the details, so meetings can take up to 10%.



## Setting up new environments, making project analyses, deployment etc.

Is it included in your project? Note that it usually takes 10-15% of the budget.



## Nonfunctional requirements for the project

If there are such requirements, they need to be agreed before we start the valuation. Do we need to run a load test to find problems before they impact users? Perhaps you want your application to automatically scale depending on its needs. Load balancing traffic requires adding more servers, so then we need to use relevant solutions that also cost money. The same goes for third-party requirements, etc.

Before providing the final estimation we also need to know whether the project requires unfamiliar technologies. We analyze potential risks and issues. You will receive a **custom estimate created after all the thorough research**.

Of course, we want to know your terms and budget restrictions, but the scope of a project should determine its budget. If it's done the other way, it may lead to a project that doesn't meet requirements.

So if a software house gives you the exact price, it should set off alarm bells in your head. The same should apply if you're only provided with the price for features mentioned in your project description. Giving a fixed price without in-depth analysis of the project, a number of calls, and a kickoff meeting is impossible. What you should get is range value.

## Be aware of underestimation

One of the most challenging things during cost estimation is to strike a balance between over- and underestimating. **IT projects tend to be underestimated** and it doesn't impact only costs. It's also related to time or technical complexity. PMI's 2018 findings show that only 57% of projects are finished within their initial budgets.

It's no secret that underestimation of IT project costs is a common mistake made by agencies and it is a problem occurring on a larger scale. It's mostly because of underestimating features that look easy to develop, but require hours of development, known as SMOP (small matter of programming).

This is why we put such emphasis on asking probing questions during a discovery call to get as many detailed answers from you as possible. This ensures that every estimation is near the final price.

And guess what? Every time we are about to make a IT project cost estimation, Hofstadter's law springs to mind

---

” Hofstadter’s Law: It always takes longer than you expect, even when you take Hofstadter’s Law into account.

---

## Project cost estimation is easier said than done

The exact specification is essential to reduce development costs. This is why **the evaluation should always be preceded by thorough research** based on key project information gathered during meetings. All of them help to evaluate work hours, product complexity, team commitment, technology, and more.

Big projects that come with a number of variables and functionalities require a detailed road-map. Hence you shouldn’t expect to get an exact price. But if you get an estimation, it will help you decide whether the project will be feasible.



# Software House Proposal - What You Can Expect

There are several good practices that the development team should adopt to provide you with a precise estimation. The more details are included, the better, as it determines the success of your final project. So, what should a software house proposal look like?

A decent proposal should contain a number of details, such as project management methodology, roles, budget, technology, timeline and more. Here is the key information that you can find in a software house proposal.

## Methodology

It is important to include a **methodology for establishing software process** development in the proposal. You know how the team works and what the process is going to look like, so it helps you decide whether it's something that is in line with your vision.

The goal is to create a friendly work environment that makes the process as effective and smooth as possible.

## Roles in the project

Before the final proposal and estimation are created, it's crucial to decide who of the specialists need to be involved in software development. Then, a software house should **estimate the effort required to complete the project**.

So, you will know exactly which of the team members are dedicated full-time, and which of them are needed only for specified tasks. For instance, when it's a project focused mostly on the backend, it may require two backend developers (full-time) and a frontend developer (full-time) to avoid a context switching problem, as well as a part-time QA specialist and a project manager (task-related).

## Technology stack

You should also know the details about **technical requirements and specific technologies that will be used in project implementation**, as well as the technology responsible for the communication between the client- and the server-side. Plus, details on what frameworks will be used or whether the application will be hosted on-premise or in a cloud-based environment.



The proposal should contain information about the planned technology and frameworks.

Deciding on the tech stack determines the logic behind how your application will work, so this part can't be missed.

## Functionalities

Another item that should be included in a software proposal is a list of system requirements and **functionalities detailed in terms of effort**.

Features need to be broken down into smaller bits to identify all use cases and see how much time is needed to build each function. It's also needed to determine the links between functional requirements and system, including maintenance and support. A single function may come with numerous subfunctions, so building all the features may take much longer than you thought.



The amount of features in the system is crucial for the estimation.

Of course, the more work required to build a single feature, the more expensive the project becomes. There are several other factors that affect the final price, so we put a strong focus on the real cost behind your project.

Thus, we know how many man-hours a certain function requires (including developers, QA engineer, and project manager time).

## Projected costs

Speaking of costs, you need to remember that software development is not only driven by technology costs, but also by time and people. This is why you need to know the hourly rate per role.

The final budget is calculated based on man-hours, but **you need to understand the dependencies for resources needed**. The value of man-hour can differ depending on experience level and the role of the IT specialist.

Think of it this way: a junior developer needs more time to figure out how to implement a solution while a senior specialist can build the function faster.



## Estimation details

Moreover, the estimation should consist of **the list of all the things that affect the final price**. These could be time allocated to QA, team meetings, setting up new environments, nonfunctional requirements, and more.

The last items that are equally important are all the constraints, comments, and assumptions that we followed when we prepared an estimation. Keep in mind that we need to perceive every function by identifying all possible use cases, which also matters for the final estimation.

---

For instance, if you need a web application that supports various languages, but you want to start with a single language version, there's more work to be done. This is because we need to prepare the infrastructure that makes it ready for future modification (when you decide to provide more languages), so it equals extra time on our end.

---

## Extra questions

Even though we ask detailed questions during a discovery call, the number of questions grows in line with the project complexity. So, a software house may need to ask some more questions to collect missing information. Be ready to provide more details — it's important to get a precise estimation.

## Software house proposal — what's next?

Last but not least, you should get a summary of all variables important for the project along with further steps. You can expect a meeting in order to discuss a software house proposal, as you need to understand how software development cost is calculated.

You can always discuss the proposal in detail. Keep in mind that **it's a custom estimation**, so if you feel that the budget is exceeded and need more information about the price, we'll get directly into every element to address all your doubts. We're open to discuss the functionalities, suggest possible modifications and change the project's scope.

04

Create Safe Agreement



# What Software Development Contract Should Include

You know what to consider while choosing a software house for your next project, so it's time to arm yourself with knowledge for signing a contract that includes all relevant information. Read on to learn the most important terms of software development.

## The contract that protects the interests of all the parties involved

Overall, the contract should be based on services and the project scope that is going to be delivered. However, note that **when it comes to large projects, it's nearly impossible to determine** what exactly will be done at a given time. So, it's more an outline of what will be implemented than a list of specific requirements.

Moreover, the contract needs to determine the project management methodology that will be used. So, if the software development is going to be based on an Agile methodology, the contract should include information that this method is agreed by both parties. Plus, it should mention the details of collaboration, such as the frequency of meetings and responsibility for their organization, tracking the timeline and progress.

The software development **contract should define the way your IT project will be developed and delivered**. A decent agreement includes the responsibilities and liabilities of both parties, costs, etc.

Let's get through the list of the most important things that can protect both you and a vendor from failure.

## What a client wants to include in a software development contract

There are some specific points that the contract should contain to make you, as a client, feel safe and confident that you'll get the service you expect. Here they are:

- 1 The test period**

These could be, for instance, the first two or four weeks of the collaboration when you can terminate the contract with immediate effect. It can be helpful in the event of bad practices or fraud on part of the software house.

It's important to have this point in a contract, as you can usually spot some red flags (e.g. poor communication with the project team) during the first weeks of collaboration. So this clause will save you months of collaboration with a software house you don't want to work with.
- 2 Vendor's general liability insurance**

with broad protection for the software development. It will help to cover any potential damages or legal fees.
- 3 Determining the key project roles and rates**

of individual team members. This point should clearly state who and for how much will work on your project.
- 4 Reporting on the progress**

It should be specified what and how often the vendor will be delivering to the client.

It can be, for instance, a detailed work report to be sent every week. You should be updated on how much time is allocated to certain tasks and who was responsible for which tasks.
- 5 Specified termination notice period**

This point should also address the consequences of termination.
- 6 Intellectual property rights**

The contract should clearly state that the software that is developed to complete your project, as well as the source code, becomes your property right after the payment is completed.

## Main contract points from a vendor's perspective

There are other key issues that a software house wants to address in the contract. Here's what you can expect:

- 1 The client's obligation to provide the main credentials**  
needed to start the software development, and maintain them throughout the contract period.
- 2 A key representative of the client**  
a member of the client's team who is responsible for communicating the client's requirements and vision to the dev team and for ongoing revision during the project. project.
- 3 Invoice payment terms**  
For instance, the due date can be 14 days from the date of the invoice. This clause is important for both parties, as software development usually involves large amounts. So, if you pay regularly, you can prevent debt.
- 4 The amount of money to be paid upfront**  
It usually equals an average amount on a single invoice. Keep in mind that a deposit lets a development company safely put time aside for your project.
- 5 Post-employment restraints**  
Many software houses include a clause that prevents the client from poaching its employees. It usually comes with a specific duration of the ban (e.g. 3 years) and a penalty to be paid.
- 6 Information about any settlements**  
including hours spent on programming, creating new development environments, project meetings, as well as bug fixing, which is a natural part of the software development cycle.
- 7 Disclaimer of warranty because of the complexity of software development**  
Note that even if a software house makes every effort to ensure the quality of its software, a potential bug can be caused based on third-party solutions, such as an operating system or browser. So it's natural that a vendor includes this point in a contract.
- 8 The limitation of liability clause**  
This one can be included to, for instance, set the amount of the vendor's insurance coverage.

There you have it! I hope the list will help you outsource your next software project more efficiently. And last but not least, before you sign a contract, **always read all the points carefully** to avoid confusion and reduce the likelihood of disputes arising.

# 05

## Prepare For the First Kickoff Meeting



# How to Conduct a Perfect Kickoff Meeting

A kickoff meeting is the first meeting between you and the project team. It's your opportunity to meet everyone involved in the project, discuss every project detail and, in result, lay the foundation for a successful outcome. How to conduct a perfect kickoff meeting?

The kickoff meeting takes place after you choose a software house that will run your project remotely and when you sign the contract with them. It may be organized online, but we find meeting in person more effective.

Let's dive deeper into a step-by-step guide to a successful project kickoff meeting.

## Setting the kickoff meeting agenda

As the kickoff meeting usually takes several hours, it's good to set the agenda to make it go smoothly.

So, the agenda will guide the conversation based on the time dedicated to a certain topic to keep a meeting in check. It will also help you better understand the project background and set the course for a successful project.

Note that the agenda may differ, depending on a software house but, definitely, the kickoff meeting should start with an icebreaker introduction. During this part, everyone introduces themselves and their role in the project, sharing their experience and background. It's about knowing each other better, so it will be easier to work together on your application.

## Understanding the project

To understand your project better, we need to know both your pain points and the very first idea that lies behind the project. **How it all started, and why do you need the project?** We need nitty-gritty information, so keep this section quick and clear.

When the background is set, it's time to discuss the scope and project details.

- 1 Project goals and success criteria**  
Explain the long-term purpose of the project and tell us how the product would help you achieve specific goals. Define the main problems and how you will measure success.
- 2 Your audience**  
Now it's time to better know the users of your application and what drives them to use it. Define primary and secondary users and provide their characteristics, such as age, experience with similar technology, education, etc.
- 3 Key features and functions**  
You'll also be asked for the most important features, functions, and dependencies. Define which of them are supposed to give the greatest business value (determine the frequency of the features' usage).

## Work organization

The next stage is about how we're organizing the work to get the job done. We'll explain to you the process based on our framework for managing IT projects effectively. Here's what it consists of.

- 1 Resource planning**  
We need to identify the available resources to complete your project on time and plan the workload across tasks and future projects.

2

## Discussing project management

Defining the role and responsibilities of a project manager and how he or she will run the project is crucial to keep the work on track. This is why we'll cover off-project management aspects of the project, explain to you how we're going to use Jira to manage your project and provide you with the details about project meetings and reporting. Expect the information about our project management methodology and practices our team will follow. We also want you to determine your role in the management process.

3

## Establishing communication and reporting channels

It's also time to agree on how the communications will be managed and what tools we're going to use to collaborate on a project.

4

## You'll learn the details of your everyday cooperation with the development team

We'll also guide you on how to communicate certain information and when it's better to use email or what kind of messages you can share via Slack.

5

## Discussing a release lifecycle

The next step is discussing a release lifecycle in order to split the work into sprints. So the purpose of this kickoff meeting stage is to determine how each phase will happen. When are we releasing, doing user testing, and running a bug fixing phase?

6

## Setting up meetings

As regular sprint review meetings are important for successful project management and for keeping everyone updated, we put a strong focus on scheduling them in advance. This is why we arrange project meetings along your timeline to make the entire process more effective.

## Anything else we need to discuss?

Time for questions. Keep in mind that delaying the Q&A session until the end helps to keep the meeting on track.

Ask anything that you think is crucial for your project or to clarify some misunderstandings. Tell us what we should be aware of, and what else we need from you to get started.

## Kickoff meeting — key takeaways

As you can see, a kickoff meeting is more than just a meet-and-greet. There are many details to set up before we start the project. A well-structured kickoff meeting sets the course for the entire project, as everyone involved understands the common goals and is on the same page.

You will understand project objectives, milestones, and potential risks and issues. But, the main goal is for everyone involved to share the same vision.

# Want to kick-off your next project with a software house?

We're here to help you and make your product a success. Benefit from our wide experience in developing custom software for clients worldwide. Feel free to reach us at: [hello@studiosoftware.com](mailto:hello@studiosoftware.com)

Free consultation



**Adam Mateja**

+44 20 3608 0386

+48 609 324 329

[adam.mateja@studiosoftware.com](mailto:adam.mateja@studiosoftware.com)

Find us on  
Social Media



Visit our  
website

[www.studiosoftware.com](http://www.studiosoftware.com)