

A Practical Guide to Outsourcing Software Development

Get actionable business results fast
by taking full advantage of the vast
possibilities outsourcing offers
to C-level executives.

Table of Contents

<u>Introduction</u>	1
<u>In-House Development vs. Software Development Outsourcing</u>	2
<u>4 Assessment Questions to Evaluate Software Nearshoring Vendors</u>	4
<u>How to Compare Prices and Assess Estimations of Outsourcing Companies</u>	6
<u>5 Holes in Your Software Development Outsourcing Contract</u>	9
<u>Fixed-Price vs. Time-and-Materials Outsourcing Contracts</u>	12
<u>How to Outsource: 3 Things You Should Prepare Before Nearshoring</u>	14
<u>What Is the Onboarding Process After Signing an Outsourcing Contract?</u>	16
<u>How Can You Manage Your Outsourced Development Team?</u>	18
<u>Moving Development Back In-House: Ensure a Pain-Free Project Handover</u>	20
<u>Final Thoughts</u>	22



Introduction

In these uncertain times of an economic downturn, looking into [software development outsourcing](#) can prove more vital to your business than ever before.

Outsourcing offers limitless, previously unavailable possibilities to C-level executives. Thanks to the internet, you can have people from all over the world working on your software project, giving you access to talent you would normally be hard-pressed to find.

But finding expert developers in limited time to meet fast-approaching deadlines is only one side of keeping your development moving forward. Amid the coronavirus shutdown, maintaining an in-house team has become much more difficult—[and far less cost-effective](#).

A remote, outsourced team will improve your cash flow and increase your efficiency, allowing you to scale up or down whenever you need without firing or hiring your team members. While your competition may struggle to survive the current financial crisis, your development will continue uninterrupted.

But before you can take full advantage of outsourcing, first you need to know *how*. The practice isn't as straightforward or intuitive as it may seem.

That is why we've compiled this comprehensive guide to outsourcing. If you wish to learn about the various aspects of the process and prepare for the challenges it can bring, you've come to the right place.

We'll take it step by step, starting with a comparison of in-house development and outsourcing.



In-House Development Vs. Software Development Outsourcing

A tale of two development methods

Let's begin by defining the most basic concepts we'll be using throughout this entire guide. Custom software development is an immensely layered process and there are two main ways you can go about it: **in-house development and outsourcing**.

In-house development has been a tried-and-tested option for many years now. Outsourcing, however, is a more recent development that has seen a significant boom recently. In 2020, outsourcing is projected to reach [a revenue of \\$220 billion](#)—that's how huge the market currently is.

But what exactly is in-house development and outsourcing, and how are they different?

What is in-house development?

In-house development essentially means using the resources you have available on hand. You build the software engineering team you need by finding, hiring, and onboarding your own employees.

The biggest draw here is that this development method gives you a lot of control over how your team members operate. You get to communicate directly with all of them and make sure each and every one is there because you believe they're the right people for the job.

That said, [in-house development is the more expensive solution of the two](#), mostly due to the very involved recruitment process you simply don't get with outsourcing. Plus, let's

not forget that your potential talent pool is limited compared to outsourcing, since you're only casting your net in your immediate vicinity.

What is software development outsourcing?

Outsourcing involves commissioning an individual or a group outside your company to handle some or all of the workload. There are many benefits to this, the greatest one being saving time and money. Since you're dealing with freelance contractors, you don't need to give them the employment terms your in-house team would require. **You also get to scale up or down faster without hurting your company's brand and improve your cash flow.**

But the real value of outsourcing is its unlimited potential; you can basically hire anyone from anywhere in the whole world. So, while you might be able to get the best graphic designer within driving distance from your company through in-house development, outsourcing gives you access to top-notch design experts who live on another continent.

Naturally, you don't get to maintain the same level of control with an outsourced team that you would with an in-house one, but it's a small enough price to pay for better results.

The two types of outsourcing: nearshoring and offshoring

Once you decide to outsource your software development, you can either go with nearshoring or offshoring.

Nearshoring involves looking for people relatively close to you—say, within a time zone or three. This gives you a greater variety of talent to choose from while still making communication relatively easy.

With offshoring, the distance isn't considered a factor. You don't care if your specialist lives on the other side of the globe and you might not be able to ever communicate in a synchronous manner—that person's expertise is all that matters.

Takeaways

It's fair to say that choosing between in-house development and outsourcing—or even some mixture of both—largely depends on your needs. However, outsourcing is the best way to cut down on costs and time, while also increasing the general talent pool available.

[Check out this article for a more comprehensive breakdown of the two development methods.](#)



4 Assessment Questions To Evaluate Software Nearshoring Vendors

Diving deep into nearshoring

We've already covered some of the most compelling reasons why it's a good idea to give nearshoring a try. But before you do, consider the following:

1) Evaluate experience

When you're trying to decide on investing in a particular nearshoring expert, the first order of business is determining whether they have the experience your project demands. No two projects are the same, after all, and if your candidate hasn't worked on anything of a similar sort in the past, they're not likely to meet your expectations.

However, we advise you to resist the temptation of picking someone who claims to be an expert in all technological fields or programming languages. You know what they say about jacks of all trades, right? Exactly. They're masters of none.

Actual experts specialize; if they claim to be great at everything, you'd be wise to take their words with a grain of salt.

What you should do is find someone who's worked with companies from your country operating within your industry. If they've covered projects similar in scope to yours, that's a good indicator they have what it takes to successfully deliver your project, as well.

2) Verify existing client relationships

It goes without saying that once you decide to work with a nearshoring expert, you'll be their client. Because of this, it's a smart move to see how their relationships with previous

clients have panned out. You can do this by simply asking them about their current clients, as well as those they've worked with in the past.

If you want to be extra sure, there are independent sites with client reviews, [such as Clutch.co](#), that should give you a pretty clear picture of the kind of nearshoring partner you're dealing with.

3) Gauge transparency

When working with someone you aren't physically in contact with for the majority of the time, you need to be able to communicate effectively. It's best to lay down your own rules here: negotiate full project transparency, the contractor's work schedule, access to the Scrum board, etc.

Aside from transparency from your partner, though, make sure you also clearly communicate everything you expect from their involvement. This will greatly speed up the entire process and ensure that your money doesn't go to waste.

4) Research team scaling speed

Scaling is a crucial aspect of today's market. If a contractor can't scale effectively, you might end up missing your deadlines, delaying the project substantially. You need to know how fast they're able to form extra teams to help with the workload.

Make sure that whoever you plan on working with has their strategies in order, so that they can make adjustments to them on the fly, as your needs change and evolve.

Takeaways

There are several bases you need to cover when choosing nearshoring, but if you're well prepared, everything should work out fine.

[Interested in learning more about assessing nearshoring vendors? Head over to our article.](#)



How To Compare Prices And Assess Estimations Of Outsourcing Companies

A crucial step in starting your software development outsourcing journey the right way is evaluating different vendors.

Hours of talks and negotiations will leave you with an extensive choice of price lists and estimations. Sometimes, their variety might shock you.

But where do the price differences come from and how to assess them?

Comparing prices: the basics

First of all, you need to keep in mind that every software project is different (at least “under the hood”) and the development process is never set in stone. Therefore, the scope tends to be difficult to predict during the first phase of negotiations. This is why you and your vendor need to be able to plan carefully and communicate clearly.

Additionally, the cost assessment should vary depending on whether you’re after team extension or end-to-end development services.

The costs of team extension

When outsourced developers join your existing product team, you’re dealing with team extension. Since this type of service tends to be based on hourly or daily rates, comparing vendors is relatively easy. However, don’t just go with the cheapest option right away; there are a few crucial factors to consider:

- **Seniority of developers**

Everyone wants the very best for their project, but there's a catch: a balanced team with appropriate competences is better (and probably cheaper) than a randomly put together, all-senior group. Don't get too wrapped up in the job titles and choose wisely.

- **Exclusivity**

Make sure the developers assigned to your project will work exclusively for you.

- **Place and time**

Some countries are just cheaper than others, which will affect the rates without affecting the quality. Also, if you plan on meeting with your team in person, consider the distance between you.

- **English skills**

It might sound insignificant at first, but miscommunication is the last thing you want.

- **Management**

The hours the developers spend on their company's matters should be paid by the vendor.

- **Assessment of technical skills**

To put it simply, it's the vendor's job to assess their employees' skills, not yours. You shouldn't have to waste time on technical interviews.

- **Pricing method**

Make sure you know exactly how the hourly rate is structured.

The costs of end-to-end development

Before starting your cooperation, you and your outsourced partner will need to agree on a ballpark estimate of the overall cost. It's wise to always consider your absolute financial limit, since estimates tend to change throughout development.

The usual components of a ballpark are:

- a feature categorization of the project,
- timeframe for each feature,
- team details,
- budget,

- additional costs.

But what are some other things you should pay attention to in order to pick a reliable vendor and ensure smooth cooperation?

1) Your vendor should know all your requirements not limited to the list of features.

They have to be aware of what you expect the final result of your cooperation to be. Are you after an MVP or a scalable, fully ready product? You also need to agree on the exact deadlines and budgets. Without this knowledge, you're heading for a communication disaster.

2) Make sure your estimation is accurate.

If your vendor sends you a ballpark before a proper business talk, it's a major red flag. Analogically, don't expect your vendor to make assessments based on pre-existing products that aren't yours—no two products are the same, so the estimation won't be reliable at all.

3) Check carefully which non-coding activities you're expected to pay for.

At the same time, you need to realize that some of them are necessary. Which additional type of work should you expect to see in an estimation? Environment setup and application deployment, team meetings, testing and reviews, spikes, production monitoring, and the involvement of a Product Owner should all be expected—and welcome.

Takeaways

Don't just go for the cheapest option. Carefully consider what the potential costs entail and make educated decisions. Remember about the importance of the non-functional requirements listed above, and always insist on a realistic, flexible estimation. Also, don't rush things if you don't have to. It's advisable to always run a solid background check on your future vendor.

[Need more detailed information on how to properly compare prices and assess estimations? Check out this article.](#)



5 Holes In Your Software Development Outsourcing Contract

The importance of a solid contract

We've already mentioned one of the key benefits of outsourcing software development: the fact that you don't need to hire people as full-time employees. Instead, you can rely on hired contractors, which can significantly cut the costs of your project.

However, you'll still be signing contracts with your outsourcing partners, and IT contracts can be quite tricky. Navigating them is no easy feat, and if your legal department isn't up to date with all things IT, even they might miss some important details.

We've identified five main areas where outsourcing contracts frequently tend to cause trouble. Being aware of them should help you avoid glaring holes in the documents you sign with your contractors.

1) Jurisdiction matters

In the majority of cases, outsourcing involves signing a contract with someone living in a country different than your own. This is one of the main reasons why outsourcing can be so cheap if you know where to look, but it comes with its own share of problems.

Since you and your contractor live in different areas of the world, by default you'll each have a different competent court and governing law over you. You can decide which law system applies to your project in the contract. It's ideal for you to use your own country's law, but that's only possible if the supplier agrees to it.

When that isn't an option, there are other solutions that should prove beneficial to you. Try meeting your outsourcing partner halfway and using a law system foreign to both of

you—British law is particularly trendy right now. You can also opt for your local court while accepting your contractor's national law, for instance.

2) Getting your staff organized

The contract you sign with your staff will determine various aspects of their work, including transparency, the team's size and composition, as well as the notice period.

These are the aspects you have to specify to ensure that your project progresses exactly the way you want it to. This is also when you have to account for staff rotation.

3) Copyright

Make sure the contract details exactly that the rights to the product are yours if you want them to be. However, it's important to note that some of the components and libraries used by your contractors may be open source.

If you're not careful with this, your software may end up being open source itself, which can be quite disheartening if that isn't your intention. For better results, it is generally agreed that using a given software house's own components may be beneficial, as long as you keep in mind to get exclusive rights to them.

You can also simply forbid your contractor from using open-source components—just make sure it's there in the contract.

4) Source code and object code

Do yourself a favor and ensure that the contract clearly states you will receive the source code, since this will allow you to modify the object code at your leisure. If you're legally considered the owner of your source code, creating backups becomes effortless.

5) Documentation

No matter the scale of your project, it's important that you document *everything*.

For starters, you can use technical documentation, intended to help developers know the inner workings of the code.

User documentation, conversely, is meant for the end users, teaching them how to use the software.

There's also project documentation, which helps make the work on the project more manageable.

Takeaways

As long as you pay attention to those important areas, you should have no difficulty preparing and signing a contract that will take care of all the legal needs of your cooperation with an outsourcing partner.

[Read our article to get more detailed information on holes in outsourcing contracts.](#)



Fixed-Price Vs. Time-And-Materials Outsourcing Contracts

Sealing the deal

With all this talk of contractors, we also need to consider the exact kind of contract we want to sign with them. Once again, the choice is something to ponder for a second.

When it comes to software development outsourcing, there are two types of contracts you can choose from: **fixed-price and time-and-materials**. It's important that you decide on one early on, as changing the terms down the line will cost you a lot of time and resources.

Fixed-price and time-and-materials contracts each offer distinct advantages, so your best bet is to familiarize yourself with both and pick the one that's best for what you're hoping to accomplish.

What are fixed-price contracts?

With a fixed-price contract, you pay for a specific product to be delivered within an established time and budget.

The upside here is that you set the duration, price, and scope of the project in advance. The downside is that getting your project off the ground may take longer, since you'll want to spend some time on those decisions to make them count.

However, the main issue with fixed-price contracts is that, well, more than the price is fixed. This type of contract doesn't really make much room for playing things by ear.

Should your plan change—and it will—you are bound to set off a chain reaction that will lead to you requesting a change in the work of your contractors.

This, in turn, will more than likely cause delays, and you're going to have to renegotiate the entire deal every time it happens. In other words, you'll be forced to start planning pretty much from scratch and have a bureaucratic nightmare on your hands.

Since you can't predict with pinpoint accuracy what your project will demand, the rigidity of the fixed-price contract may actually turn out to be more of a detriment than a benefit. However, if you're prepared to deal with some delays and extra negotiations, the firm foundations the contract offers may be just what you need.

What are time-and-materials contracts?

A time-and-materials contract has you pay for exactly what it says on the tin. Instead of paying for the entire predicted scope of the project upfront, you compensate the team for the actual work and expertise they put into your project.

This type of contract essentially means painting in broader strokes, which may be less specific than the fixed-price option, but it also gives you a lot more freedom in managing how your team arrives at the goal you want them to reach.

Not adhering to a strict end date set beforehand means you're free to alter the terms of the cooperation with relative ease. Rather than waste time struggling through red tape, you simply send your team off to work right away.

Working on a time-and-materials basis provides an opportunity to be much more reactive to the process and make adjustments to the workflow on the fly. You can change priorities quickly and give regular feedback in response to what is actually happening, as opposed to making wild guesses based on more-or-less reliable stats.

Takeaways

Both fixed-price and time-and-materials contracts are viable legal solutions. Deciding which works better for you really depends on your personal preference and needs.

[Give this article a read if you wish to know even more about each type of contract.](#)



How To Outsource: 3 Things You Should Prepare Before Nearshoring

Getting ready for nearshoring

In a way, nearshoring is the comfortable middle ground between offshore outsourcing and in-house development. The talent pool is broader and you still have a pretty high chance of sharing calls with your contractors.

However, there's a lot more to nearshoring logistics than meets the eye. The process of ensuring everything runs smoothly can be broken down into three main points.

1) Appointing a Product Owner

It's not enough to just find a software house that will handle the work you want to outsource. A team is only as good as the person leading it.

Since the whole point of outsourcing is not having to micromanage your project so much, you need a different, smarter solution to this.

That's where [the Product Owner](#) comes in. This person is typically someone from your company added to the outsourced team in order to provide them with all the context and background information they need to truly bring your product vision to life.

[Product Owners can help](#) put your project back on track should the need ever arise and make necessary calls regarding the budget, time, and required materials. In essence, they're the expert who knows your product vision well and manages the work for you.

2) Creating the initial backlog

It's always dangerous to go it alone, so make sure the Product Owner has the product backlog on them. This backlog serves as the initial plan for your project. Your Product Owner should have one with a list of requirements for three or four sprints.

The purpose of the initial backlog is to streamline the entire development process from the word go. Therefore, it's absolutely crucial that the backlog be ready before the team starts working.

Keep in mind that the backlog is *not* a list of specifications your final product must have. Rather, it's meant to make development smoother by pointing out the possible hurdles your team may encounter along the way.

3) Establishing dependencies and gathering data

Last but not least, remember to establish communication channels you and your team will be using. Set up dependencies between the team members and your company's employees. This will guarantee a far more organic transition once the project is ready to be handed over to you.

This usually means preparing code repositories and account accesses, among others. Make sure someone is there to help handle the hosting, cloud systems, testing environments, production environments, and so on—especially if your project started out with no requirements for internal infrastructure.

Takeaways

Preparation is the key to fruitful cooperation. Setting everything up right from the get-go significantly reduces the workload as your project evolves.

[We discuss Product Owners, backlogs, and data gathering at length in this article.](#)



What Is The Onboarding Process After Signing An Outsourcing Contract?

Getting everyone on board

Before work on your outsourced project commences, you need to make sure everyone's on the same page regarding all the project requirements. To that end, client onboarding is an indispensable step.

During the onboarding process, both sides detail the conditions of their partnership, establishing the right course of action and setting expectations. No outsourcing project should begin without thorough onboarding.

The importance of client onboarding in the outsourcing process

Cooperation with your outsourcing partner will go smoothly from day one if your onboarding process is both comprehensive and flexible. It lets your partner know exactly what to expect from you and your company, strengthening your work relationship and making it much easier to establish the logistics, budget, and timeline of your project.

Having said that, there's no one-size-fits-all method to successful onboarding. Make your process bespoke and take into account everything you want your partner to know.

Consider how you'll handle the budget and paperwork, how long you want the onboarding to take, and all the remaining formalities with that specific partner in mind.

What to discuss during onboarding?

How much ground you should cover throughout the onboarding process largely depends on the degree to which you're going to be relying on your outsourced team and whether they're going to be starting work from scratch.

Especially in the latter case, it really pays off to include the software house you're partnering up with in the design process. This will allow you to come up with better solutions together.

How long should the onboarding process take?

Onboarding can take anywhere between a day and a couple of weeks. This depends entirely on the complexity of your project.

How to prepare for onboarding?

The goal of the onboarding process is for the team members to acquire all the information necessary to understand the goals of the project. Your job is to make sure they have everything they need.

Since you're the client, you have the deciding voice on the general flow and structure of the project. You can make things considerably less chaotic by introducing the outsourcing partner to your company's business context.

Any technical documentation you're willing to share—backlog, source code, development environment automation—is welcome, since the extra data will make the job of your outsourcing partner much easier.

You should also be ready to address any concerns your own developers may have, if the outsourced software house serves as an extension of your in-house team. Tensions are not uncommon and it's up to you to make sure the members of both teams manage to overcome any potential differences they may have and work together as a single entity.

Takeaways

A sound onboarding process is in the best interest of every single person working on your software project. For your outsourcing efforts to be effective, it is invaluable.

[Hungry for more onboarding tips? This is just a taste of what we offer on the subject over here.](#)



How Can You Manage Your Outsourced Development Team?

A degree of control

With outsourcing, you give up some of the control you normally have over an in-house team, but that doesn't mean relinquishing all of it. Just because your team is far away doesn't mean you can't communicate with them to ensure that all your expectations and requirements are met to your satisfaction.

Here are some of the aspects you should pay attention to in order to properly manage your outsourced development team.

1) Levels of contact

Contact is the single most important tool for managing outsourced teams. You need to establish how closely you want to work with your team and how often you wish to communicate with them. We recommend staying in touch on a daily basis, although that really depends on the overall flow of your project.

There are five main means of contact you can choose from. Naturally, whether you opt for just one or even all of them is entirely up to you.

a) Chat tools

Tools like Slack are perfect for frequent real-time communication.

b) Catch-up calls

Usually lasting around 15 minutes, catch-up calls serve as quick progress updates.

c) Backlog refinements

These are regular meetings meant to clear up any difficulties and evaluate new developments in the project.

d) Sprint reviews

In Agile software development, which is divided into Sprints, Sprint reviews take place after each Sprint to see what went wrong or right, what could be improved, and the like.

e) Face-to-face visits

These may be the most difficult to organize, especially if your team is located *really* far away from where you are. However, such visits are worth arranging every once in a while to build rapport instead of just following up on details all the time.

2) Manager calls

Aside from staying in touch with the development team itself, you should also consider contacting the Service Delivery Manager regularly. They're the person charged with making sure your project is carried out to the highest possible standard. As such, they're also the point person for nurturing your outsourcing partner's relationship with you.

Manager calls typically happen every two to four weeks, unless your project requires some additional attention in the meantime; for instance, when there's a change in the team's composition, a deadline fast approaching, and so on.

Takeaways

While the specifics of your relationship with your outsourced team will vary from team to team, it's almost always a good idea to keep in touch with them frequently or at least have the means to contact them quickly should the need arise.

[Take a look at this article for a deep dive into managing outsourced development teams.](#)



Moving Development Back In-House: Ensure A Pain-Free Project Handover

Passing the torch

No matter the reason, there may come a time when you'll want to move the development of your software back in-house. This is by no means unheard of; sometimes, clients only turn to outsourcing to overcome specific challenges at a certain stage of development.

Making the move back in-house can be the most optimal choice for some companies. However, it's a delicate process that takes thought and preparation to ensure a smooth and pain-free transition.

Here's what you need to do to make it happen just so.

Identify the right moment

If you make the move back in-house at the wrong time—too soon or too late—you may create new problems for yourself instead of solving old ones. Usually, the best moment to start the handover process is simply once the work has concluded.

Sometimes, though, the smarter play may be to do it earlier. This can happen when the development process slows down and shifts to a more stable maintenance status, for example.

Basically, if you feel you have both your feet firmly on the ground, you're probably ready.

Making the software project handover painless

There is no single, universal way to make the handover process painless, but there's a high chance one or several good practices will do the trick.

You can meet your contractors face to face and work through the process of handing work over to your in-house team step by step, so that everyone is on the same page. Alternatively, you can do the same remotely—the internet makes pretty much anything possible (including recording things, something you should rely on heavily here).

Make sure that all the proper documentation is taken care of, since you don't want to be left with a piece of code that doesn't come with any manual. Many software houses opt to provide this in the form of a wiki, which is a very practical way of going about it.

One last tip: scale the development down before you start the handover. For rather self-explanatory reasons, it'll make the transition much smoother.

How long does the handover process take?

The question is a bit difficult, but not impossible to answer. As we've stated several times here, it all depends on the project. The more complex your system's architecture is, the longer handing it over to you will take.

In most cases, though, it's pretty safe to assume it'll take about a month or so to complete the handover process. This estimate actually holds true for both a full handover and scale-down, so keep that in mind.

Takeaways

A software project handover might be just what your project needs when it's fully or nearly fully developed. Don't be afraid to go down this road if you've made the necessary preparations.

[We highly encourage you to read our full article on moving development back in-house.](#)



Final Thoughts

Well, here it is: all the essential information you need if you're thinking of outsourcing your software development.

Follow the advice we've outlined for you in this practical guide, and you should have no problem getting fast business results with outsourcing.




There's a whole world of possibility out there—you just need to reach out for it.

We've been in the software development outsourcing business for almost 18 years now, with well [over 750 projects successfully delivered](#) under our belt.

Feel free to [reach out to us](#) if you believe there's any way [we can help you out](#) with your outsourcing efforts.

Hire an exclusive Python development team







Accelerate your software project with Europe's largest Python software house.
For companies with big projects and fast deadlines.

 Team Extension	 End-to-End Development	 Consulting & Expertise
Additional developers or experts supporting your development efforts within 14 days	Full development team taking your project all the way from discovery to deployment	Solving your problems or improving your product with the help of subject matter experts



Over 400 developers

Ready to empower any project with well-reviewed code and a results-driven Agile process

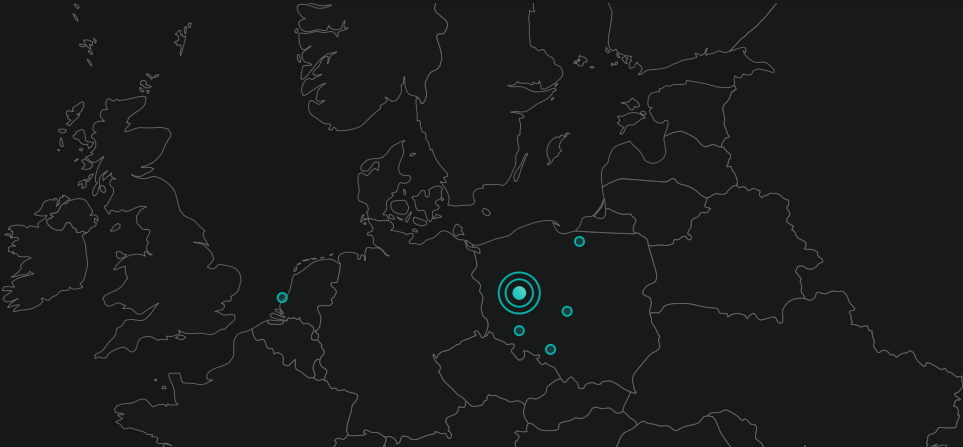
 17+ years market experience	 750+ projects delivered	 3.5+ years average partnership
 300+ clients served	 550+ professionals on board	 6.5+ years average experience of our developers

Locations

Poznań (HQ) ●

Mostowa 38
61-854 Poznań, Poland
+48 61 610 01 92

- Wrocław ●
- Olsztyn ●
- Katowice ●
- Łódź ●
- Hague (Netherlands) ●

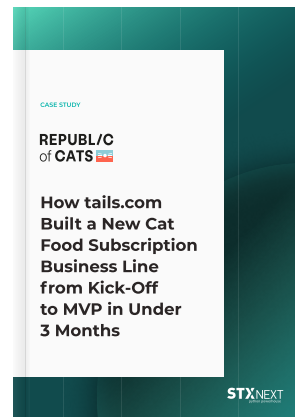
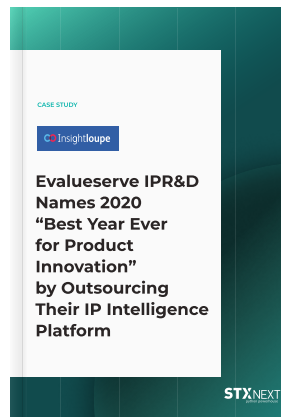
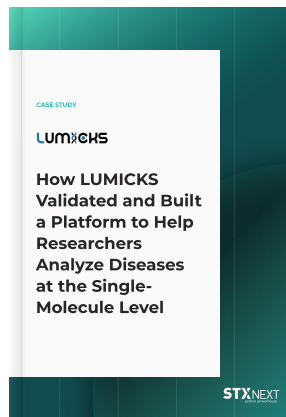


Resources

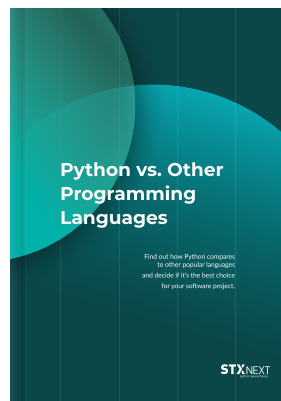
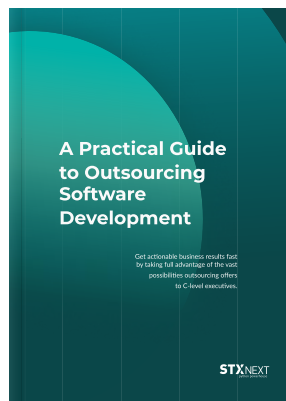
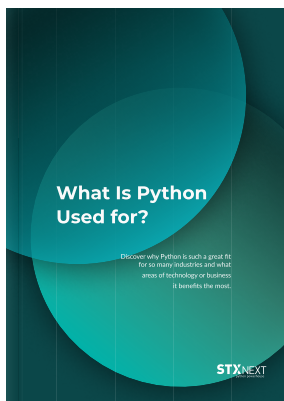
Arm yourself with the expert knowledge you need to successfully deliver software projects.

Get free in-depth resources, templates, and checklists—all based on 17+ years of software development experience.

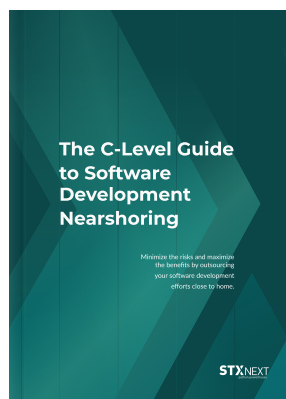
Reports and case studies



Guides



Ebooks



DISCOVER MORE →

Services

Your project is all that matters. We'll build it like it was our own. Whether it's team extension, end-to-end product development, or expert consulting you're after, we'll do everything in our power to meet your needs.



Python Development



JavaScript Development



.NET Development



Web Development



Software Testing & QA



Mobile Development



Django Development



Node.js Development



React Native Development



Fintech Development



Machine Learning



Data Engineering



DevOps



Product Design



Discovery Workshops



Tell us about your project

Speed up work on your software projects and outpace the competition.

HIRE US →



Marta Błażejewska

DIRECTOR OF SALES

marta@stxnext.com

+48 506 154 343



Sebastian Resz

HEAD OF SALES

sebastian@stxnext.com

+48 690 433 578

Follow us

in

f



Bē

