ASPER BROTHERS

ULTIMATE GUIDE

SOLVING THE CHALLENGES OF SOFTWARE OUTSOURCING

41 Actionable Strategies from 46 CTOs Worldwide

WWW.ASPERBROTHERS.COM

TABLE OF CONTENTS

WHY DID WE WRITE THIS EBOOK?

According to Statista, overall global spending on IT services in 2020 will reach 1.1 trillion dollars, and software outsourcing will become a steadily growing piece of that pie.

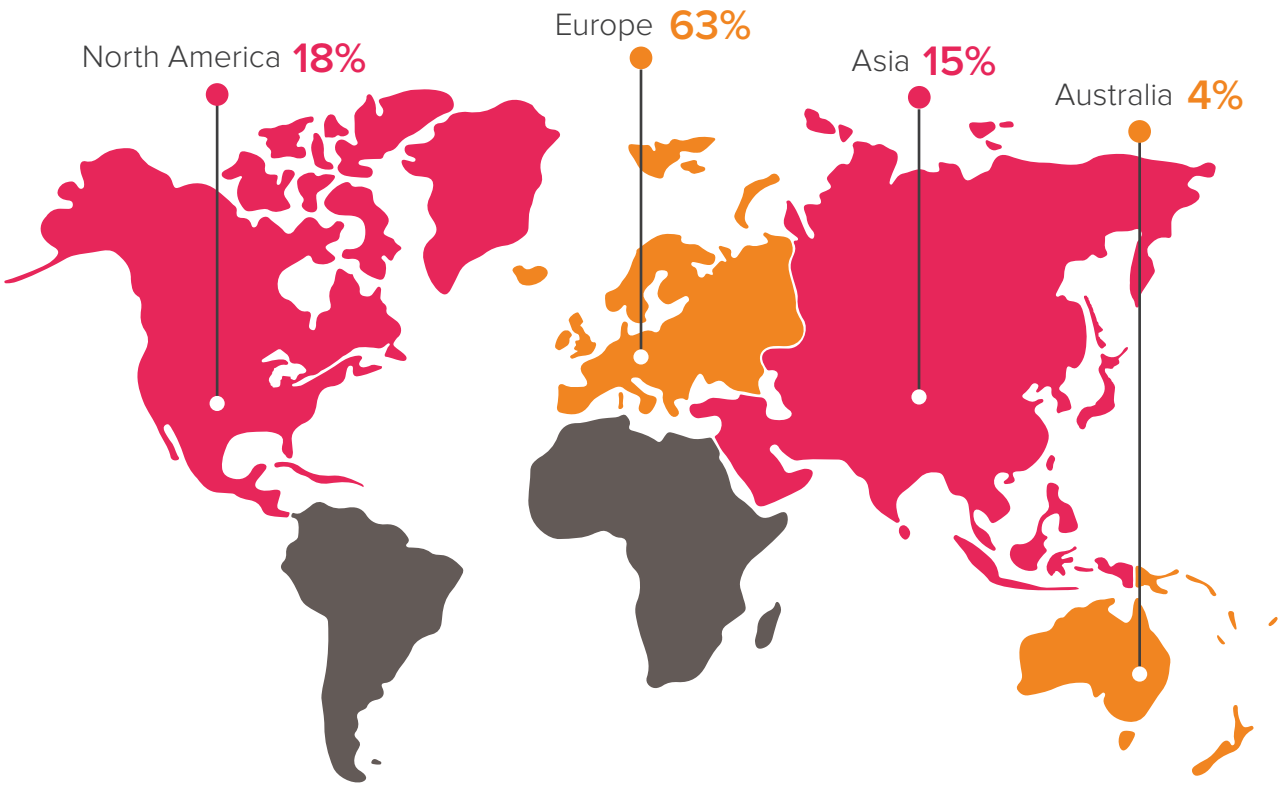
Working with international clients every day, we're no strangers to the daily challenges of IT outsourcing faced by both sides of the process. Seeing how many of these challenges tend to recur, and how relatively easy it is to get (at least some of them) out of the way of successful projects, we decided to write it all down in a helpful guide for tech leaders worldwide.

But we didn't want to rely just on our experience (though we have quite a lot of it, but still, we realize we're just a single team of experts). So we set out to ask CTOs of companies all over the world about their outsourcing experiences, traps they've fallen into, and methods they've devised over the years to deal with various types of challenges.

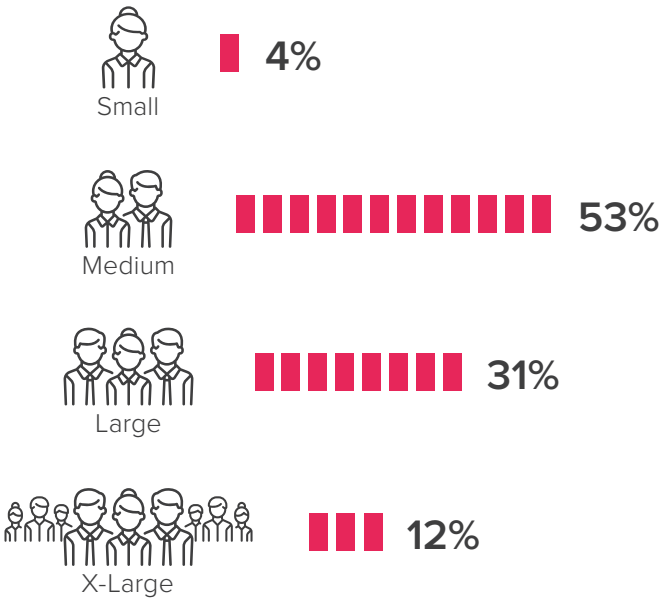
What you're about to read is an insightful mix of what they said with what we notice from a vendor's perspective. It is designed to give you practical ideas you can use in your software development practices to make outsourcing projects less of a pain (which we know they can sometimes be).

MEET THE CTOs INTERVIEWED

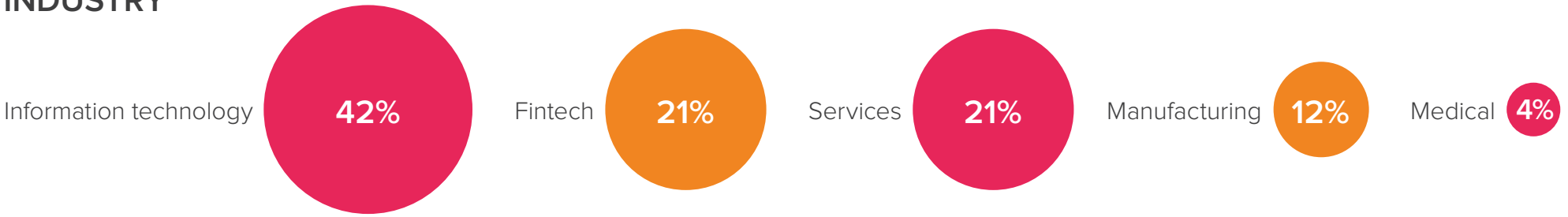
COMPANY LOCATION



COMPANY SIZE



INDUSTRY



WHAT YOU'LL LEARN FROM THIS REPORT

- What daily challenges tech leaders who outsource software development face all over the world,
- What helpful tools and methods they use to overcome them,
- Why communication between teams is so important, and what you can do to make it better,
- How to overcome cultural differences when working with offshore teams,
- The real cost of outsourcing software development,
- Practical tips to make life easier for both your in-house team and your software partner.

ABOUT THE SURVEY

We surveyed dozens of CTOs from Europe, Asia, North America, and Australia in Q4 2019. Interesting (not really fun) fact – only two of our respondents were women (both of them working in Germany), which shows we still have a long way to go when it comes to gender equality in tech, especially at the C-level.

Our respondents represented both small businesses (of up to 10 employees) and big enterprises with several thousand employees, in different industries.

OUR KEY FINDINGS

- It probably won't come as a surprise, but tech leaders frequently highlight the importance of agile methodologies in helping overcome communication challenges and syncing internal and external teams.
- Although we live in the age of tech, and there are plenty of tools out there, the most helpful way to communicate is still talking to other people (emails turn out not to be that great).
- The external team needs to know not only WHAT they're building but WHY. And it's your job to tell them.
- Paying less most often means getting less – and there's no workaround.
- Asked about ways to deal with communication differences, two-thirds of our respondents focused on the relationship, while only 33% mentioned actual software tools.

We divided the contents of this ebook into three main parts, focusing on challenges and – most of all – solutions, in the three most prominent aspects mentioned in our survey answers: communication, pricing, and tech.

If you're a CTO looking to outsource your software or improve your ongoing outsourcing projects, take the advice coming from other CTOs in the next chapters and use their experience to anticipate challenges and have solutions at hand before they actually happen.

So, take a break, grab a cuppa, and we hope you enjoy the read!

COMMUNICATION AND SUPPORT

POOR COMMUNICATION CAN COST YOU MORE THAN YOU THINK

If we were to pick one thing that comes up the most in our survey answers as key to successful outsourcing projects, it's definitely communication between the in-house and outsourced teams – the biggest challenge according to 35% of our respondents.

And, actually, it's old news. Communication is key to the success of a project and the whole organization's bottom line. Communication IS money (if only because you can measure it in saved time and higher quality outcomes that lead to more revenue). But a lot of companies still undervalue its importance and don't have the right tools and procedures in place to facilitate it.

The result?

According to project management statistics, poor communication is among the top five reasons why projects fail.

Communication is key to the success of a project and the whole organization's bottom line

PMI's seminal research (dating back a few years but still cited as the most extensive study on communication in project management) shows that for every $1 billion spent on projects, $135 million is at risk. And a whopping 56% of this (which amounts to tens of millions of dollars) is at risk precisely because of ineffective communication.

PMI's seminal research shows that for every $1 billion spent on projects, $135 million is at risk

On the other hand, companies that do recognize the significance of communication and – more importantly – act on it, have higher chances to succeed (and are 1.7 times more likely to outperform their peers financially).

For something so essential and relatively easy to fix with the right mindset and some readily available tools, all this might sound shocking. So let's look at some ideas to tackle the most common challenges CTOs mentioned in our survey.

1. INTERNAL AND EXTERNAL TEAMS DON'T COMMUNICATE ENOUGH

CHALLENGE

This is a big one and a common cause of bugs, missed deadlines, and disappointing outcomes, with products that are not up to the original expectations and the needs of the user.

Let's face it – syncing external and internal teams on progress and issues is crucial for a project's success. And working with remote teams makes face-to-face collaboration harder, purely because some members of the team just aren't there in person.

SOLUTIONS

WORKING AGILE. Agile methodologies are often cited as a solution to a lot of the problems facing companies that outsource their IT projects. Scrum forces people to align all the time through daily and weekly meetings, reviews, and other agile ceremonies.

TALK AND MEET OFTEN. Regardless of agile, most CTOs see the importance of frequent, direct communication between the teams. It helps prevent a lot of hiccups and lets everyone prepare if there are holdups or delays that otherwise people may not be aware of.

DASHBOARDS TO KEEP TRACK OF THE COMMUNICATION AND PROGRESS. It's not exactly enough just to talk – it's equally important to keep track of the conversations in a way that's easily accessed by everyone and includes all the critical information about the project. Project management tools like Jira have been frequently mentioned as helpful. What's important is that teams shouldn't be overwhelmed with too many tools, as this will be counterproductive.

INSTALLING LOCAL PMs who can serve as a direct link with the outsourced team, working hand in hand with them every day. If that's not possible, working closely with the outsourcing company's PM and visiting the team in person once in a while, or having them come over to visit your company is some of the advice given by the CTOs we talked to.

SCOTT PETRONIS
Xcentric Consulting, LLC

Leader's Advice

Leaders need to make [communication] their priority, helping everyone not only see the big picture but feel their contribution. It's also crucial to truly ***make people feel like they can make a difference with their ideas****, feedback, and creativity. There are many ways to reward that, whether it's simple verbal recognition or anything else, but it needs to be sincere.*

If you create an environment where everyone is valued, they understand their value, and they see how it all fits together, you're going to see great things happen, regardless of whether or not the team is in-house, external, or a combination.

ONLINE VOICE/VIDEO MEETINGS. Most of our respondents agree that real-time communication is key to resolving misunderstandings, which is rarely possible through back-and-forth emails or even instant messaging tools. Typing and reading are just not the same as a direct conversation.

AGREEING ON A CLEAR COMMUNICATION SCHEDULE UPFRONT — and sticking to it. It will take some time in the beginning, but CTOs acknowledge that the investment is crucial, especially for large, complex projects.

ANDREW ROBINS
PM Connect

Leader's Story

[One of the biggest traps of outsourcing is] underestimating the amount of support the outsource organization may need. In a previous role, we passed on custodianship of a complex, existing product, and did it without providing the relevant up-front support.

Fast forward to the first deliverable from the third party, and it was a mess.

The first, instinctive reaction was to rail against the third party, but on doing a deep dig, we realized that we had not only failed to support them upfront – no handover process, no training, no documentation – we had not even provided them the most recent version of the product we expected them to work with.

Outsourcing does not mean people will 'magically' just understand what they need to do.

2. EXTERNAL TEAMS DON'T UNDERSTAND THE BIG PICTURE

CHALLENGE

Another trap CTOs emphasize is that the outsourced team is often hired to "just" do a specific job but is not briefed on what's behind it. 19% of our respondents said the "quality of vision" is the main challenge they face when outsourcing. People who work creating a product don't have an understanding of things like:

- the goals you want to achieve,
- who the end customer is and what they need,
- what problems you're (and they're) solving for the customer,
- and why they are even doing what they're doing (apart from earning money for writing the code).

And the thing is, they need to know all that. They need to know as much as you can tell them. They need to "get" who you are, who your customer is, and what you're trying to achieve. The problem is, they often don't know any of that.

SOLUTIONS

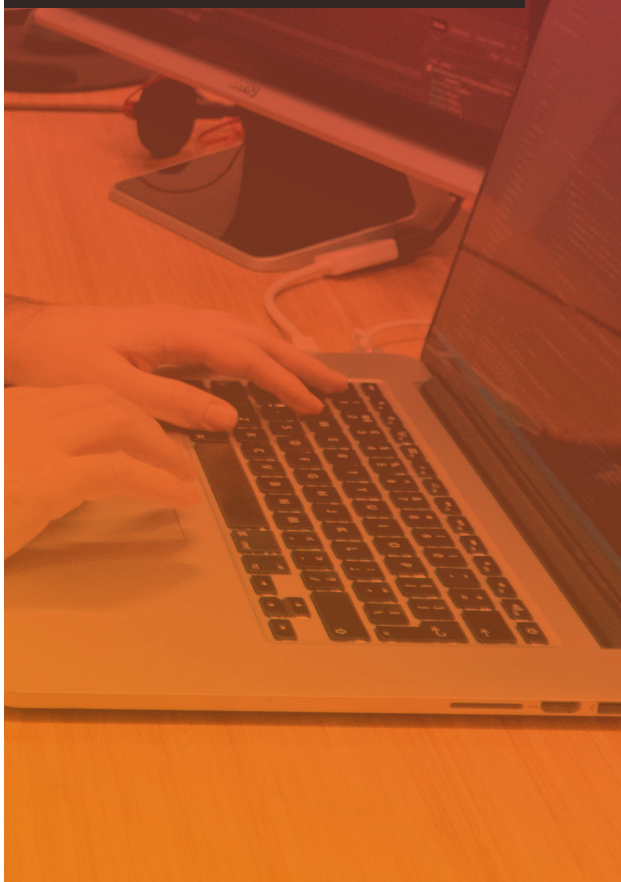
WORKING AGILE (you guessed it). Agile methodologies are great for keeping teams in the loop on your goals and helping them understand the ideas behind the work they're doing. As one of our survey answers put it, development cycle ceremonies like refinements are great tools for tactical alignment of the teams. To align teams on the strategy in long-haul projects, a good idea is periodic (e.g., bi-monthly or quarterly) co-creation events concentrating on longer-term goals.

BRIEFING TEAMS UPFRONT. In as much detail as possible. An indispensable step in project management, it will help both sides agree on project milestones and jobs to be done, while bringing in the context like: ↴

HENRY GEORGE
Expend

Leader's Says

*Clear communication of the purpose behind a task is key. Due to the language barriers, it's often easy to express **WHAT** to build, but not **WHY** it's being built, and so the smaller unsaid details are left uncommunicated, and hence unfinished.*

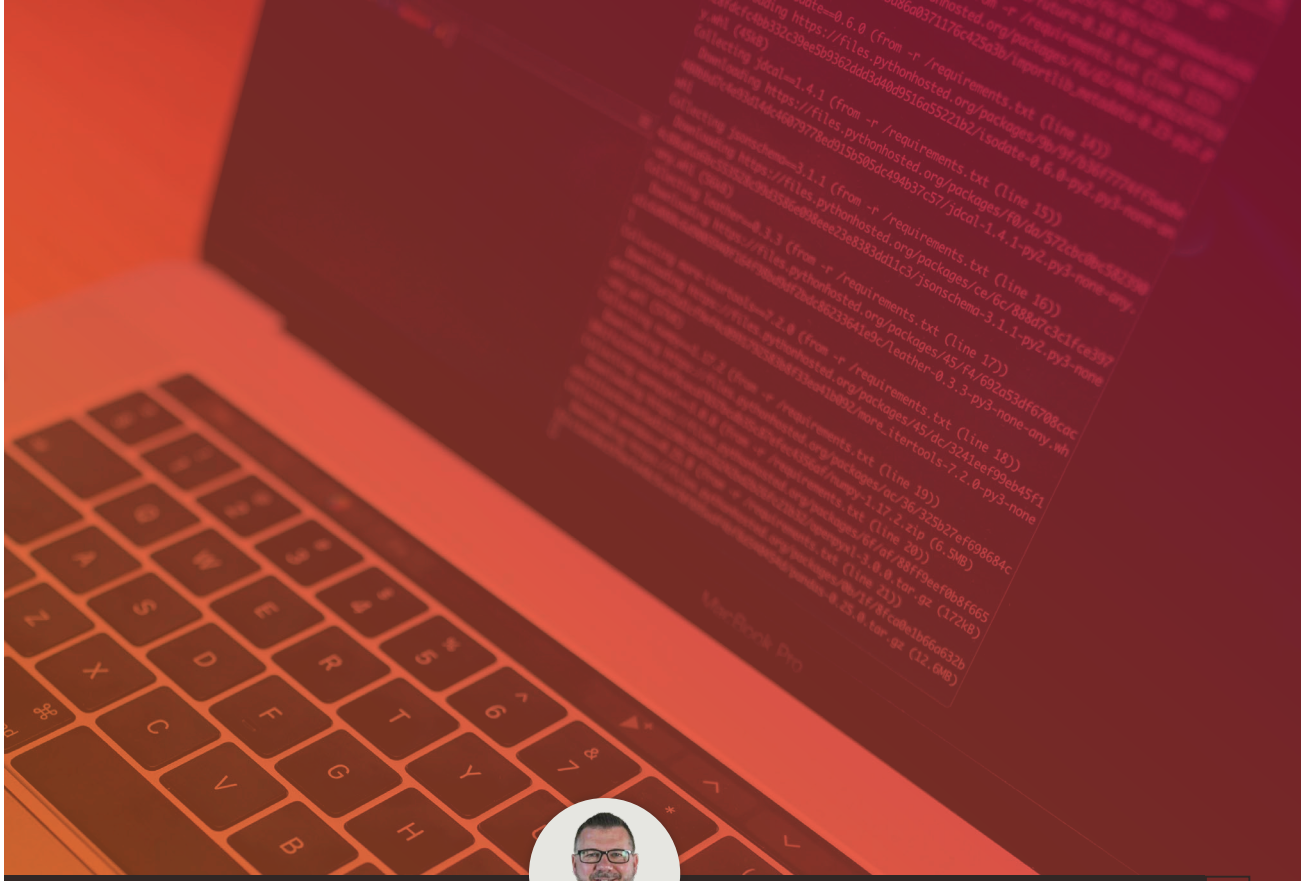


- your business goals,
- the goals of your customers who will be using the software,
- the market for the product,
- how you stack up against the competition.

Again, this does take a time investment, but CTOs agree it's worth it. It's your chance to clarify a lot of the doubts and misunderstandings before they can affect the schedule, budget, and the quality of the software.

And when you think about it, it's true for any industry – the more detailed the brief at the start of the project, the more accurate the result (and the happier the client). It's that simple.

DEFINING EXPECTATIONS AND KPIs and then continuing to align on them and track them. Defined KPIs (or OKRs, whatever you're working with) are a must to track progress and help teams understand long-term goals that the short-term goals lead to.



SCOTT PETRONIS
Xcentric Consulting, LLC

Writing code is just one part of the delivery of a great product. Everything ***needs to start from a vision****, a shared understanding of the purpose, who you're building for, what problems you're solving, the significance of solving those problems, and the expectations for quality, usability, performance, and other crucial KPIs you expect to measure. This isn't a one-time thing either.*

I find it crucial to ***remind the team*** *of these items regularly (weekly or even daily in some cases) and try to attach specific discussion topics to these measurements.*

Leader's Advice

3. CULTURAL DIFFERENCES

CHALLENGE

Outsourcing often means working with offshore companies – and that is almost sure to bring out cultural differences, including in work culture. And it's not always as simple as the stereotypical contrast between an outgoing American and a reserved Brit.

A simple "yes" might not mean the same thing for everyone involved in the project (and not only because their English is not great, which is going to happen to some extent when you outsource to non-English speaking countries). You might think you've all just agreed to what you talked about while your Asian team has only acknowledged that they have understood what has been said (without actually agreeing to it).

Then fast forward to a feature, or worse, product release and they're surprised you expected something they never agreed to deliver.

And that's just one example. There's a whole list of expressions, body language, and business rules that might give you a severe headache during the project if you don't research them first.

One commonly cited trap is when the team doesn't like to admit they don't understand something (especially if that "something" is a crucial part of the project), affecting project schedules and product functionality. It's then up to you to make sure they do understand it and know they need to voice their concerns and communicate issues as they occur.

Different regions of the world will have different outlooks on deadlines (which is a nicer way to say they might not care about them as much as you do). Some Asian cultures favor seniority and hierarchy, which might potentially cause problems when working with a younger project manager on the client side. Some teams' schedules will be influenced by their religious practices and some will work nights and weekends, around the clock.

And it's not only about geography and outsourcing projects to vendors located in a different continent. Companies in the same country can have a different company culture and management styles, which can also be a source of friction.

The experience CTOs shared in our survey shows a lot of the times leadership teams want all the benefits of agile development, but without adjusting their work style, especially at the executive management level. This can lead to conflicts and communication issues that no tools can fix.

JOHANNES DROOGHAAG
Johannes Drooghaag

What the Swedes call fika, and the Dutch call coffee-talk is a good demonstration of creating an informal atmosphere in which it becomes much easier to communicate the small issues before they become big issues. Leaders tend to be 'too busy' and prefer to say ***'give me solutions, not the problems'****, and on the other side, project teams and CTO's still prefer to explain why something is happening instead of including peers in the process of solution-finding.*

Leader's Advice

SOLUTIONS

FIND COMMON GROUND BEFORE THE PROJECT BEGINS. Learn about the culture of your outsourced team and practice inclusive communication that embraces the differences on both sides rather than making them a divisive issue

CREATE DETAILED CONTRACTS both sides can refer to in case of cultural misunderstandings. This is really important, because contractual disagreements (which we'll also talk about later) are a commonly mentioned challenge in our survey answers. Make sure you agree on the country whose laws govern the contract (and that you – or at least your legal team – know them well).

TREAT EVERYONE FAIRLY, REGARDLESS OF THEIR CULTURE. It seems pretty obvious, but we have to say it. People want to be respected no matter where they live in the world. Common national stereotypes are not helpful and they might further escalate the miscommunication, sometimes leading to discrimination or offensive behaviors. So brace yourself for the differences, but be open to them and treat them as lessons to learn.

DO ALL THE THINGS WE ALREADY MENTIONED ABOVE. Agile and frequent direct communication will help you make sure you are indeed on the same page (and of the same book). This probably can't be stressed enough – and it's what CTOs keep repeating.

MIKE BAUER
Formpipe

I never ask closed questions *such as "do you understand the solution," but ask open questions that require an elaborate answer ("Please explain the function as you understand it," "What do you think should be paid attention to when implementing this function," etc.)*

We also visit each other regularly (in the beginning it was every month, now less regularly.)

Leader's Advice

4. PEOPLE LEAVING TEAMS WITH THEIR PROJECT KNOW-HOW

CHALLENGE

Anything that's not recorded in project documentation can and will be gone whenever a member of the project team leaves the project (or their job altogether). You don't have any control over the outsourced team's internal churn. And when someone who's worked on the project for some time leaves, they take that precious project know-how with them. Unless they're in the room next door working on another project (which is sometimes the case), you usually have no way of getting that information back.

SOLUTIONS

RECORD EVERYTHING. Don't rely on oral agreements; document as much as you can. And what you can't record? (And let's face it, not everything can be recorded). Go back to the first challenge and the solution – frequent, direct communication will help you stay on top of what's going on, making it much easier to pass on project knowledge to new team members. Simply because you'll know about a lot of the things first-hand.

5. PROVIDING FEEDBACK AND ACCEPTING IT

CHALLENGE

Feedback is a strange beast. A lot of people are not great at giving it, and another lot are lousy at accepting it. And the thing is, without feedback, there's just no progress – or it's heavily limited.

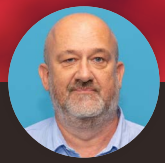
Teams that don't receive feedback don't know what to change and what to keep doing. And by not giving feedback frequently enough, you're missing the chance of getting the project back on track and getting the outcome you expect.

Of course, this is true for constructive feedback where things could (and should) be improved. But what's equally important – and mentioned by the CTOs we surveyed – is providing positive feedback to encourage the team to keep going and keep up a positive team spirit. Which is inherently challenging when you're working with remote teams – and all the more important.

SOLUTIONS

PROVIDE AND ENCOURAGE FEEDBACK – AS FREQUENTLY AS YOU CAN. Make it a natural part of your communication. Meeting people often helps solve a lot in this case – because by design you'll talk a lot to your team. You'll focus on the pressing issues, and reporting issues will become a part of regular discussions instead of turning into a blame game.

CELEBRATE PROGRESS WITH THE REMOTE TEAMS. This is something a lot of companies forget to do with outsourced teams, but it can make a huge difference. Recognize the team's (and its members) contribution, don't make your offshore team a bunch of anonymous coders.



ANDREW ROBINS
PM Connect

KEY Lesson

Tech tools are helpful, e.g., Slack, Skype, Hangouts, but do not underestimate the benefit of ensuring the right people are talking – a functional, ***clear human relationship*** *between the tech point of contact in both organizations is absolutely essential, it will help overcome all the other problems.*

CTO ADVICE

SCOTT PETRONIS
Xcentric Consulting, LLC

Leader's Advice

People need to know that it's not only okay to communicate the good, the bad, and the ugly, but it's expected. Team members need to understand how to communicate negative information in a way that's inoffensive but also that relays the severity. This skill is not easily mastered nor is it easy for anyone to hear negative feedback and not take it as personal criticism.

I find it best to establish these expectations upfront, provide examples of some of the "what" and "how" and then individually discuss with team members to understand each person's style. Then, in daily stand-ups, ensure that items are being brought up and to coach, as needed, early on.

My expectation is that everyone in a team becomes adept at providing and accepting both positive and negative feedback and that everyone provides both in a respectful way. I also tend to tell stories about all of my own screw-ups along the way, so people understand that we all experience this.

THE CTO TOOLBOX: COMMUNICATION

- Scrum, Kanban and everything that comes with agile methodologies, like sprints, daily, weekly, and monthly meetings
- Jira, Trello and other project management tools
- A solid QA cycle and code reviews
- Dedicated Slack channels and plugins (like https://slack.com/apps/A454FNE64-stand-up-alice), WhatsApp groups
- Video conferencing tools like Google Meet or Microsoft Teams
- Collaborative video tools like PukkaTeam or focusmate

HOW TO GIVE CONSTRUCTIVE FEEDBACK TO YOUR TEAM

- If you're still using "the sandwich technique", stop. It's predictable and even if it works once or twice, that's about it. After a couple of times, it just feels like a game everyone's playing.
- Instead, be honest – whatever the feedback is. People value honesty and they respond well to it (surprise!).
- Yes, people will sometimes play manipulative games, like blaming everyone around. Learn how to see through them and respond with honesty.
- Strike while the iron is hot. Don't wait with feedback (both positive and negative) and give it as soon as you can, while the situation that prompted it is still fresh in everyone's minds.
- Stick to the facts. Talk about the situation, not about the person. Use descriptions instead of judgements.
- Actively listen to what the team is saying. Don't just assume you know better.
- Don't focus on the blame-game. Look for solutions. Ask the team what solutions they see.
- Ask what they need from you to fix the problem – without fixing it for them.
- Avoid asking "Why" questions that relate to the past (like Why did you do this?!) – they usually trigger defensive responses. Steer the conversation towards finding solutions.
- Give credit where it's due. Positive feedback is just as important.

COMMUNICATION MAY SOUND LIKE AN EASY THING

But it's not. It's a complex and delicate matter. Especially when outsourcing.

Agile, at least when well performed, guarantees excellent communication in good remote software projects, and this is also what most CTOs recommend as a solution.

But agile ceremonies won't work unless developers believe in them. And that's not always the case. Ideas like "Async Communication" (check, for example, asyncmanifesto.org) are gaining ground. And of course, we understand some of its assumptions. It's counterproductive to force developers away from their work and waste their valuable time on unproductive meetings.

But we're seeing more and more developers thinking that agile is just some strange business game, and their daily status calls are discussions about all things except the main topic. Businesses need to work to change this mindset.

CTOs and Scrum Masters around the world should keep agile rules in mind and be very strict about them if they want to benefit from them. They need to prove why agile is beneficial for everyone involved, especially developers. Agile teaching and coaching should be the responsibility of the outsourced party. So check how well your future team knows its principles.

> *To establish a healthy environment you need additional time, dedication, traveling, human touch, and leadership skills*

If you want to establish a healthy environment when outsourcing software projects, you need additional time, dedication, traveling, human touch, and excellent leadership skills. But if you can master this practice, you'll recognize your outsourced team as a precious asset.

Overtooling won't replace the human touch, and we recommend you show it to your developers regularly. You need to know people. Some of our best projects were the ones where we had a personal relationship established first. When we'd had the opportunity to work together in the same room (even if just for a few days), seeing and hearing each other, understanding personalities and emotions, drawing together on the whiteboard, and going out for dinner and drinks afterward.

We want to demystify outsourcing thought of as working with robots and not real people. Approaching it from a human perspective lays the ground for building context, avoiding misunderstandings, understanding "why" and not only "what" we're making. As a result, you have collaboration, time savings, and excellent outcomes for your project.

Mike Jackowski, Pawel Jackowski,
ASPER BROTHERS founders

DIRECT CONTACT:

Pawel Jackowski

paul@asperbrothers.com
pauljackowski

FUNDING AND CONTRACTS

2

ESTIMATING GOOD PARTNERSHIP

Although – as a recent report on outsourcing created by Deloitte shows (*source*) – priorities have shifted a little in recent years from cost cutting to disrupting and innovating, for many companies one of the top reasons why they outsource IT is still a common belief that outsourcing is cheaper. Which is partly true, at least for some projects.

But the challenge comes with the popular mistake companies make, which is looking for the cheapest vendor while expecting high-quality results. 15% of the CTOs we questioned mentioned this as the biggest trap they've fallen into.

For many companies one of the top reasons why they outsource IT is still a common belief that outsourcing is cheaper

We'll leave the quality bit until the next part, but let's talk about the pricing, budgets, and contracts, sources of some major challenges for both development companies and their clients.

1. ESTIMATION ACCURACY

CHALLENGE

This is a common problem for a lot of companies with almost any budget. A lot of things can go wrong – or just differently – in a project, even in-house (I mean, who's never heard of exceeded budgets?). And when you outsource projects, things may become even trickier, because it's not you who does the actual job, so you're losing some degree of control over the budget.

SOLUTIONS

MAKE THE REQUIREMENTS SUPER CLEAR FROM THE VERY BEGINNING. This is something our respondents mentioned a lot. Of course, it has to do with communication, which is the recurring theme in all the answers our respondents gave. But it also has to do with the time investment that needs to happen if you want accurate estimations.

List all your functional requirements (the features your software should have) in a spreadsheet. The more detailed requirements at this point, the more accurate the quote you can get – and the fewer surprises later on.

Separate your must-haves from nice-to-haves and wish lists. Decide what's absolutely necessary for the MVP (minimum viable product). What can come next? What's great but won't hurt if you can't have it?

CREATE BALLPARK ESTIMATES in the beginning to provide a high-level overview of the project costs. Break the project down into phases, starting with the discovery/analysis/design/scoping phase. Then prepare a much more accurate – and detailed – final budget with your client (which is what we do at ASPER BROTHERS, by the way).

AVOID FIXED-PRICE PROJECTS. Most software development companies are moving away from fixed price, especially for complex, long-haul projects that need a completely different approach. If you're not outsourcing small, straightforward projects, use time and material estimation, which is much more flexible.

PAUL SASSELLA
CHU Underwriting Agencies Pty Ltd

Leader's Advice

*It's simple: set ambitions end date/ timeline according to your requirements (which needs to be properly documented! Including your goal/ expected outcome), and account for some deviations (due to sick days, vacation, capacities constraint, requirement adaptations – usually **+25%** of your originally allocated time).*

*Make **EVERYONE** in the team aware that the budget is the budget, if amendments to the scope have to be made, other requirements (of less importance) need to make room for it. **Do NOT** let developers experiment with "maybes", "nice-to-haves" – sorry devs out there, I know you get excited, but there is a time and place – and this project phase is not the time to do it.*

FIXED PRICE VS. TIME & MATERIAL

CHOOSE FIXED PRICE	CHOOSE TIME & MATERIAL
When you have a limited, fixed budget	When you're flexible about the budget
When you have a strict schedule, especially if it's short	When you're flexible about how long the project will take
For a small project with limited requirements	For a complex, long-term project whose requirements might change
When you want to leave most of the project management to the software development company (which we don't recommend)	When you want more involvement in project management – and more control
When you use traditional project management methodologies like the waterfall model	When you use agile project management

2. LOOKING FOR SAVINGS WITHOUT SACRIFICING THE QUALITY

CHALLENGE

This happens everywhere – companies that outsource projects (and not only in IT) look for the best price. It's natural, right? But limited budgets rarely go hand in hand with high quality.

SOLUTIONS

DON'T MAKE THE PRICE YOUR PRIORITY. When looking for vendors, take a long-term look beyond the price: what's the track record of the company? How do they work?

Use the RFP process to ask the right questions and to hire the right people. If you're just looking at the price, you might end up with lower quality/less functionality/slower development (and eventually, even higher costs overall).

LOOK FOR OFFSHORING OPPORTUNITIES. Hourly rates in Eastern Europe can be even ten times lower than in Western Europe or the U.S.. On top of that, research shows it's, in fact, much easier to find skilled IT experts in countries like Poland or Ukraine.

Of course, the same thing applies – the price shouldn't be the primary factor in making your decision, and you should do a solid background check on the team you're hiring.

GET SKILLED PROJECT MANAGERS. Strong in-house project management skills can help you avoid overspending – among other things. Some of the CTOs who took our survey also recommended a hybrid approach to outsourcing. The developers are outsourced, but they're managed just like an internal team with in-house project management. This can help you both cut costs and avoid a lot of the other outsourcing challenges, including most of the communication challenges.

HENRY GEORGE
Expend

Leader's Advice

*If you pay less, you get less quality; it's as simple as that. People think of outsourcing as **1/10th** of the price of a quality in-house developer, but that's not at all true, **you get what you pay for**. In the past, we've sought out the cheapest, but in reality, I believe the right mark is around **50-70%** of in house developer costs.*

3. CONTROLLING THE BUDGET

CHALLENGE

Tracking your budget and schedule is closely tied to estimation and defining the requirements. Spend too little time initially to break down the project into details, and you'll have serious problems making sure the budget's on track.

Companies that want to outsource as much as they can and minimize their involvement might find this part especially difficult. And not tracking the budget is particularly dangerous and might cost you much more than anticipated.

SOLUTIONS

GET INVOLVED. It's the simplest way you can make sure the budget is on track – both on the part of the client and the development company. Agile methodologies help, and Scrum was again cited as one way to combat the problem (which makes Scrum and other agile methodologies look like an antidote to most of the software development outsourcing problems).

GET PREPARED. Deliverables, timelines, budget, responsibilities, goals, and results – if these aren't well-understood, documented, and communicated continuously, the project will simply not live up to expectations because expectations change. So we're back to formalizing requirements and accurate estimations. It's all interlinked, and neglecting one piece of the puzzle will affect the entire project.

JACOB THOMAS
Iconic Live

Leader's Advice

I have 20+ years of experience in breaking down large tasks into sprint size tasks which helps me create very reliable time tables and budgets for my clients. Most companies struggle to accurately estimate what the headline and high-level tasks mean in real work. Time estimation and budgeting is more art than science. I perform all-time estimation and budgetary controls on a project basis, with my team managing task level estimation.

*We blatantly ignore the **Agile "Story Points" estimation method**, using this to estimate effort in hours based on the developer/PM's experience and requirements. This allows the dev or Project Manager to calculate effort and budgeting in a single pass. We also build in a QA/Test/Documentation as a percentage of all tasks to allow for the estimation of non-development tasks. We also have ensured that our time estimation & budgeting processes are completed by people who are actually technically competent, having development experience in what they are estimating.*

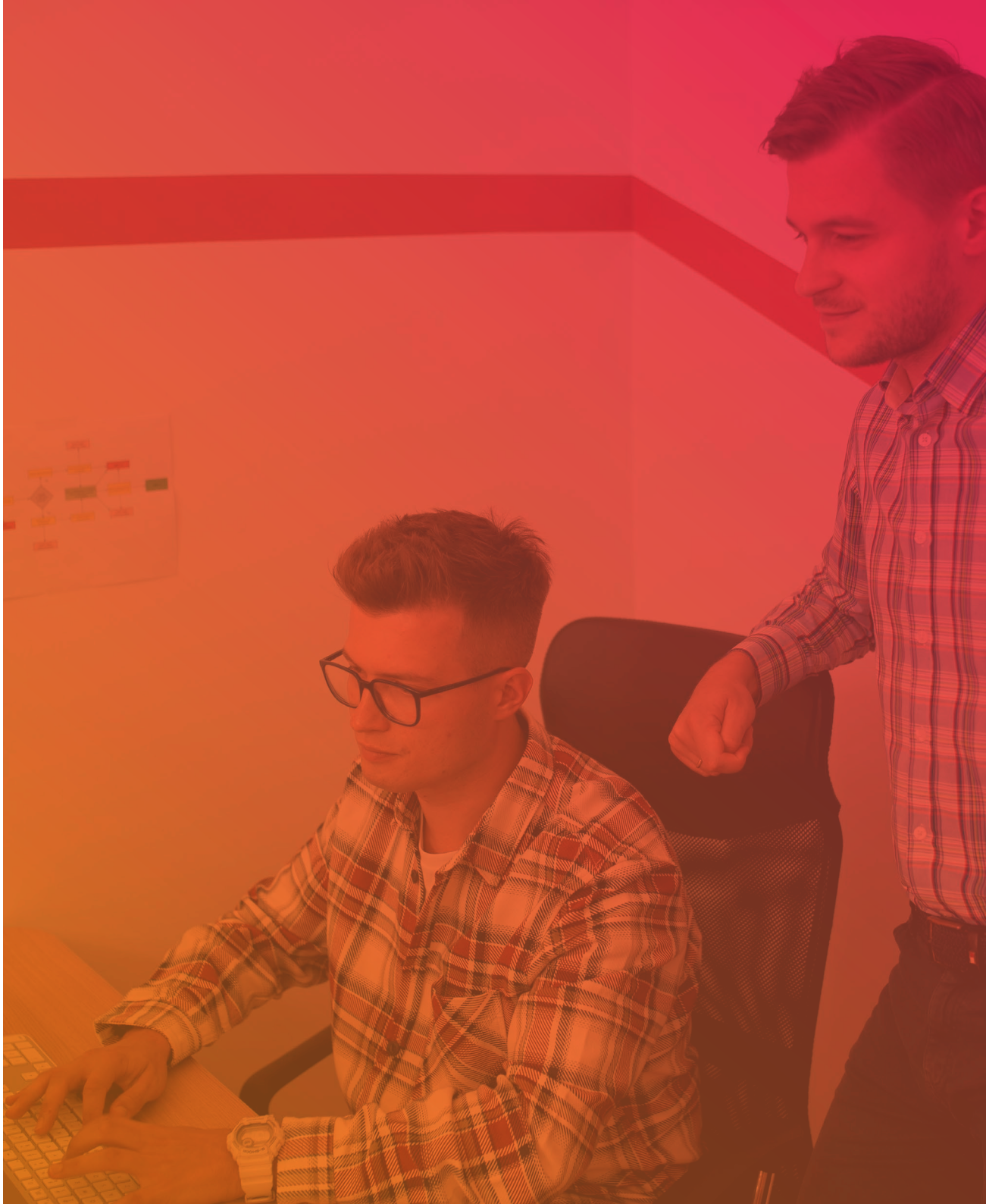
*I also personally refuse to have **non-technical program/product managers**. They are an absolute waste of space as they just push paper and cause friction between technically competent teams.*

FOSTER HONESTY AND ACCOUNTABILITY IN THE TEAM. People on both sides of the project should feel the work they do is essential in keeping deadlines and budgets. Using spreadsheets or even expensive budget tracking software will do nothing if there's no feeling of ownership and accountability for everyone. So make this clear from the outset.

People – especially in different cultures – might hide the fact they won't make the deadline or need much more time (and spend more budget) than anticipated. You have to make sure you know about this as soon as possible, so you can react and come up with a backup plan.

THE CTO TOOLBOX: BUDGET CONTROLLING

- *Scrum process*
- *Other Agile methodologies*
- *Spreadsheets*
- *Progress reports*
- *Jira & integrated software like Tempo*
- *Time and budget tracking apps*



4. CONTRACTUAL DISAGREEMENTS

CHALLENGE

Especially with different cultures – there can be a great deal of back and forth between the legal teams and different understandings of the contract. People will interpret the contract differently, and – as one of our respondents aptly put it – interpretation is "the mother of all f*ckups."

SOLUTIONS

MAKE THE CONTRACT SUPER CLEAR. Make sure the most important elements, including requirements, are not buried in the legal language that no one other than your legal team can understand. We're so used to the thought that legal stuff "has to" be complicated that we forget agreements are for people to agree – and that should involve everyone, not just your legal advisors.

COMMUNICATE, COMMUNICATE, COMMUNICATE. During the whole process. Read the contract, clear all doubts, and don't leave anything to "worry about later." If you want to avoid worrying later, that is.

TRUST SO MUCH THAT YOU DON'T NEED A CONTRACT BUT HAVE ONE.

Oh, the holy estimates and their accuracy in IT projects... Business needs budgeting and want to know how long it will take, while the development teams create #NoEstimates movements. How do you reconcile the two sides?

Working on estimates is challenging. There's a lot of frameworks and techniques, but still no perfect solution in sight. The best option is to work out a compromise among all of the stakeholders, covering as many edge cases as possible, with a mixture of well-known techniques. We suggest bringing everybody to the table to understand each other. After all, we're talking about money here.

> *Be comprehensive, clear, and detailed when it comes to requirements*

Be comprehensive, clear, and detailed when it comes to requirements. Break them down into small chunks. Yes, it takes time. But if you want your outsourcing projects to meet your requirements and expectations, you need to be open to allocating in-house resources to work with the remote team.

> *Let's also assume your developers are not very proactive*

And remember – the remote team is not around to ask questions. Small things that may sound obvious to the in-house team are not so self-evident to the outsourced team. Let's also assume your developers are not very proactive, and a lot of details are left unsaid, misunderstood, and not done.

We also often see clients don't pay enough attention to contracts. Or they think that they shouldn't bother because that's what the legal department does. You already trust the outsourced party, so why should you worry about it, right? Wrong direction.

As a CTO, you should study the contract thoroughly. A good agreement is not just the payment schedule and IP rights. It includes the way of working, performance measures, or, last but not least, an exit plan. So put a contract discussion on your agenda when you're traveling to meet face to face with your team for project kick-off.

Mike Jackowski, Pawel Jackowski,
ASPER BROTHERS founders

DIRECT CONTACT:

Mike Jackowski

mike@asperbrothers.com
in mikejackowski

Pawel Jackowski

paul@asperbrothers.com
in pauljackowski

SOFTWARE!

TECHNOLOGY AND QUALITY

3

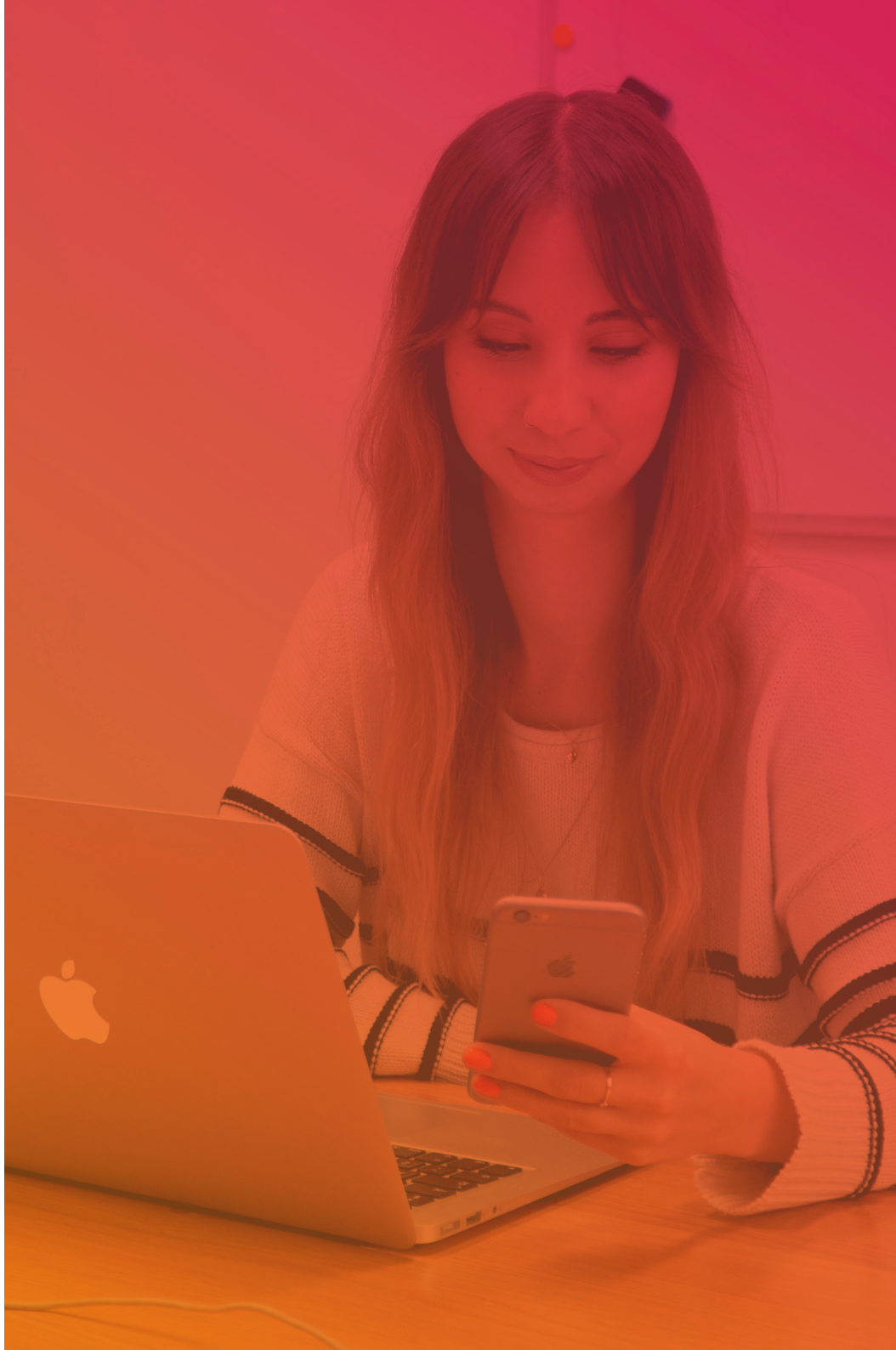
TECH CHANGES WILL ALWAYS BE AHEAD. SO KEEP CLOSE.

Challenges related to tech and product are perhaps the most complicated ones and might seem the hardest to overcome. They'll vary depending on what IT functions you outsource and what tech (maybe your own) you already have in place.

IT is one of the most dynamic industries and the future of most of the other sectors like automotive, finance, real estate, which is getting dominated by software. We live in times people won't stop talking about digital transformation and disrupting industries, spinning futuristic fantasies of the world in decades (climate change aside). Yet a lot of companies keep struggling with dated tech, stuck with patched up code, not being able to make the actual move.

Outsourcing the tech might seem additionally challenging because you're not in direct control – of the team and of the tech itself – with team rotations and skillsets possibly affecting the quality of the end product.

But it doesn't have to be the case. Take a look at the challenges CTOs mentioned in our survey and the solutions that can help overcome them, or avoid them entirely.



1. AVOIDING TECHNICAL DEBT – AND PAYING IT

CHALLENGE

Once called “the silent company killer”, it’s a growing concern for a lot of the businesses that focus on fast development and quick iterations – a standard in today’s agile teams. It’s maybe the one challenge that might be aggravated by agile methodologies, at least when it’s not taken into account – and often, it’s not – Causing time gained in the short run, with much more lost in the long term.

The cost of managing technical debt in large software companies is estimated to take 25% of the overall development time. Apart from the code itself, the debt can be related to the architecture and infrastructure, documentation, or even date back to the requirements. And, just as with any debt, if teams don’t recognize it as a potential threat in time, it’s going to grow.

SOLUTIONS

HAVE A SERIOUS CONVERSATION WITH YOUR OUTSOURCING COMPANY ABOUT TECHNICAL DEBT. The debt can be challenging for both sides, but it’s the client who will deal with it in the long term, possibly slowing down or entirely stopping further development of the product. Make sure everyone is aware of the problem and feels accountable. Knowledge of the product roadmap and long-term strategy will definitely help.

MAKE SURE THE DEVELOPMENT TEAM TRACKS AND MEASURES THE DEBT. Tech debt should be one of the standard metrics tracked during the development stage. In every sprint, time should be allocated to paying it down, bug fixing, and stabilizing the environment as part of the backlog. Along with the team, you should keep your finger on the pulse and monitor how individual sprints affect the product and its architecture.

Pay attention to code churn – if it’s going up, code is being reworked, and you may be risking higher tech debt in the future.

REFACTORING IS THE NEW BLACK. Check that it’s standard practice for the team and they understand its importance instead of sweeping it under the rug. Of course, code refactoring most probably will affect the schedule, maybe the budget, too. But – as with anything in IT – it’s not worth taking shortcuts. You’ll pay for them later, possibly much more.

CHOOSE THE RIGHT TECHNOLOGY. A solution that’s a challenge in itself (but we never promised it was going to be easy). So let’s discuss it in greater detail on the next page.

2. CHOOSING THE RIGHT TECHNOLOGY

CHALLENGE

As in other areas of business, IT managers and directors along with other C-level executives, often fall for the shiny object syndrome. It can be so tempting to pursue new tech that everyone (and especially the competition!) talks about that it may push people to focus on the benefits, without analyzing the drawbacks and the adequacy to the individual business needs of their companies.

Software development companies around the world (including us) have multiple stories to tell about clients coming back to rework projects that weren't created in the optimal technology in the first place. Don't be one of those clients.

SOLUTIONS

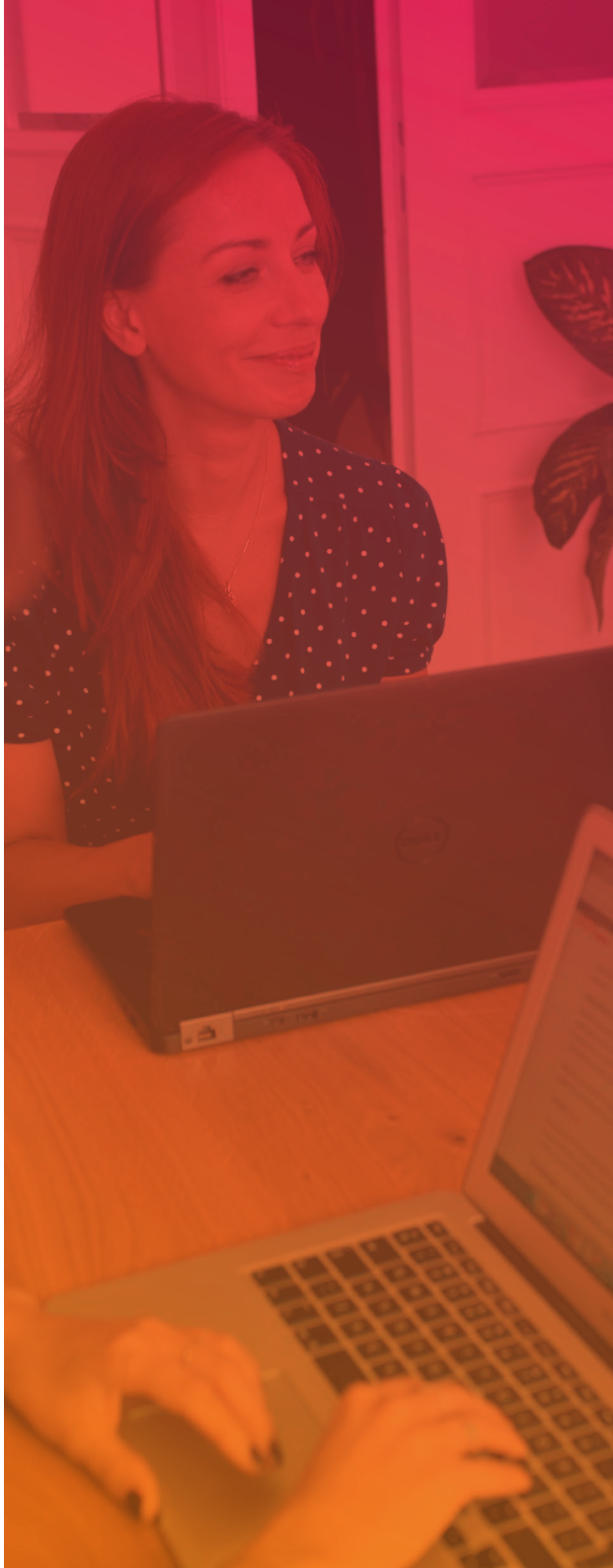
TRUST YOUR SOFTWARE PARTNER AND DISCUSS THE OPTIONS WITH THEM. A trustworthy, experienced tech partner should be able to verify your needs and advise on the right technology. If they're not thrilled with your choice, be open to alternatives. Especially if the team has had experience with similar software.

That's why including the dev team in creating the architecture and choosing the technology is critical. They may be able to suggest solutions that are not only innovative and proven but also better for your budget and product in the long run. Continue the trust throughout the development process and be open to what they suggest – or at least have continuous discussions.

MAKE DECISIONS BASED ON FACTS AND DATA. Analyze all the essential aspects like:

- scalability
- product vision
- product road map
- budget
- integrations
- and even barriers to entry for other developers who might continue working on the software in the future.

In other words, don't just go with your gut. Choosing the tech has consequences (sorry if we're Captain Obvious), including creating technical debt we just talked about a few paragraphs earlier.



3. NO CLEAR GOALS OR PRODUCT ROADMAP

CHALLENGE

This is a common problem with a lot of organizations with a particular impact on the outsourced team's productivity. Without a clear roadmap for a product, it's much more difficult to logically divide the development work into manageable stages – and then control them. It can create chaos for the in-house team, let alone a remote team, who has no insight into the client's plans.

It can also make the software development company look inefficient – mostly because they don't know the subsequent steps. And it'll affect the tech stack choices that might not be scalable or easy to integrate with other software or features in the future.

Again, you're risking rewriting the code at some point. Or a constant struggle with changing requirements. No clear roadmap might mean ad hoc project scope changes, which might make the project longer and more expensive.

SOLUTIONS

PLAN AHEAD – AND COMMUNICATE YOUR PLANS. This one's simple (at least in theory). When you outsource the IT development of your product (or its features), make sure you have the roadmap clear and approved on your end before you do it. A good product roadmap has to come from your product strategy and the overall vision.

As you've already seen from this report, most CTOs see the lack of a clear vision as one of the major roadblocks. But that vision has to be translated into an ordered roadmap with feature sets and their goals.

Plus of course, you need to communicate it to the software development company – to the extent that they need to be able to look ahead and anticipate possible issues. Don't just let your dev team know what's happening in the next few sprints, with backlog filled up basically on the go and no regular refinement sessions.

It'll make their work easier, and your product better.

AVOID PIVOTING WITHOUT DETAILED TECHNOLOGICAL ANALYSIS FIRST. Software by its very nature is dynamic, and so is business. If you're tempted to scale your business up or down, don't overlook the technology. It will most likely be much harder to change than your business model.

DOUGLAS AUSTIN

Special School of District of St. Louis County

Leader's Advice

*We use **story points** and **core objectives** for the project outcomes. We try to tie all iterations and deliverables to a key objective for the project. We track value added as well as task completion during our sprints. We also add periodic check-ins with the team to voice concerns or pose questions about the overall goals. There is still a lot of room for **misinterpretation of the goals** in this process, but we try to identify them as quickly as possible.*

Have a detailed damage control plan to mitigate risks and be ready to implement the necessary changes.

GET SPECIFIC. It's not enough to have clear goals – you need specific KPIs to measure them. We've already said it, but we'll say it again. You need to start with a vision and problems you're solving, expectations for quality, usability, and performance – and the dev team needs to understand them. Each of the areas needs to have clear KPIs, including those that are:

- strategic for the company (like customer satisfaction or revenue)
- project-based (like deadlines, budgets, quality and performance)
- task-related for specific parts of the project.

Use concrete numbers to track them – you'll be able to monitor the progress more effectively, while the development team will likely be more productive.

MIKE BAUER
Formpipe

Leader's Advice

It's easy to fall for the appeal of a shiny MVP, but actually, the important thing is the data model, page load efficiency etc. I recently had one that looked great, but there were all sorts of shortcuts taken so that the system simply didn't scale in a normal browser context and I had to rewrite a lot of it myself.

***'Just Working'** isn't good enough: **Code** and **network operations reviews** are really important. And don't forget to specify those upfront! Wireframes won't get you clean JavaScript!*

4. POOR USER EXPERIENCE

CHALLENGE

User experience is one of the aspects of your software you should track – but it's a separate challenge in our ebook because it's so important. And it's not just important because the two letters in "UX" are on everyone's lips in the C-level execs' offices. Getting high quality software is not enough – it has to be created with the user in mind.

There are tons of studies (*source*) showing that a well-designed user interface can triple a website's conversion rate, and better overall UX design can boost it by up to 400 percent.

With the current competitive landscape, software that's complicated, illogical, difficult to use just won't cut it for your customers. Sure, we have large corporations still using old school software monsters out of habit (because they're most certainly not using them to save time or money). But if you're building software designed to disrupt the market, or at least keep users using and enjoying it, you can't neglect UX.

SOLUTIONS

HIRE A TEAM WITH USER EXPERIENCE EXPERTS. It's slowly becoming a standard for software development companies, but make sure the team's serious about UX and knows how to design user interfaces that are not a drag to use. And check if they know UX is not just about UI (because the two often get mixed), but also deep knowledge of user psychology and needs.

TEST WITH USERS. Never assume you know what your users would do, unless you've verified it with them before. Brief the dev team on the needs of your customers, so they understand who the end user will be. Agile user stories are an effective way to tackle this. But we're back to square one – for the product owner or team members to be able to come up with relevant user stories, they have to have the right knowledge of important aspects of the project.

SCOTT PETRONIS
Xcentric Consulting, LLC

Leader's Advice

If building a solution that consumers are going to rely on daily, make sure everyone understands what it's like to be in that person's shoes.

What's the difference between 5 seconds and milliseconds? How willing would you be to use this if it gave the wrong answer 25% of the time? How would you like it if you needed to go through 6 steps to get an answer rather than 2 steps?

It's crucial that ***expectations are spelled out early*** *so the team understands what's worth spending time on and what's not.*

5. A PRODUCT THAT DOESN'T LIVE UP TO EXPECTATIONS

CHALLENGE

The low quality of the product might stem from a lot of the issues we've already talked about like:

- making the price your main factor when choosing a vendor
- insufficiently communicated requirements
- poor (or lack of) communication during the project
- the team's poor skills
- an unrealistic schedule
- or – conversely – projects that drag on for too long
- an inefficient quality assurance process.

The latter is a big one with product quality. CTOs often don't have enough time or resources on their end to test individual product iterations and don't do manual or integration tests. And if you think about it, it misses the point of agile development completely – because thorough testing is still done at the end of the development process, exposing the product's shortcomings at a very late stage. Which just doesn't make sense.

Luckily, there are numerous ways to fix this.

SOLUTIONS

TEST MORE – AND MAKE SURE YOUR SOFTWARE PARTNER DOES THAT, TOO.

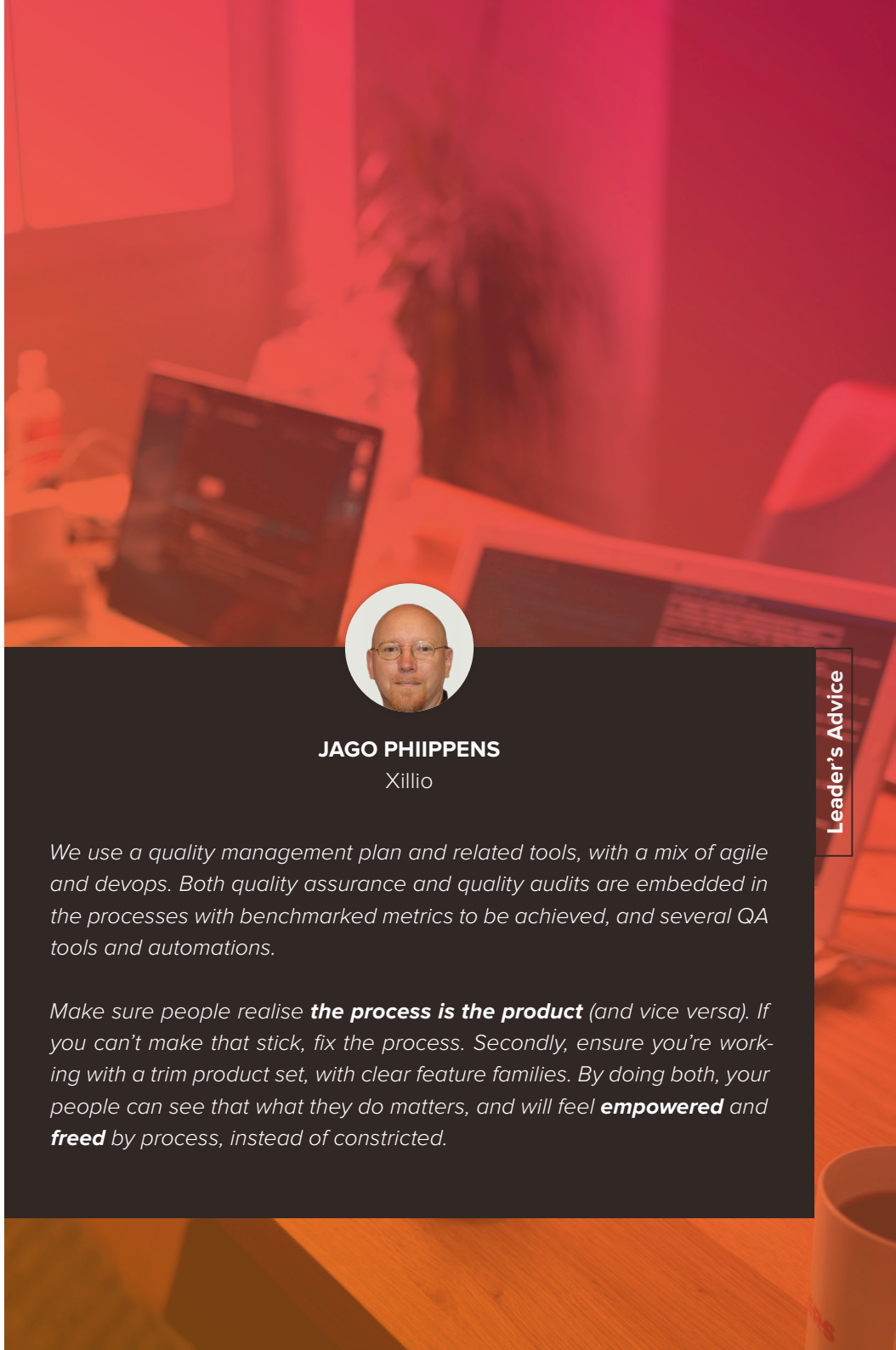
- Get a manual tester to test each iteration with different scenarios, and get feedback to the development team.
- Test the software with your existing systems to avoid incompatibility and performance issues.
- Do automated testing and make sure the code coverage is sufficient for the tests to be meaningful. A good practice is introducing automated test reviews into the demo ceremonies.

ENCOURAGE A CODE REVIEW CULTURE. Everyone makes mistakes (including top developers). If you've managed to create an open communication environment we discussed in the first chapter of this ebook, this should be its natural element. If we can share our personal experience as a software development company, CTOs are often not interested in this aspect or how it's done on the part of their outsourced team. Which is a shame.

Code review should be adjusted to the individual nature of the project, although we think the ideal situation is when it's performed daily.

DOCUMENT THE PROJECT. Especially if you're handing over a product that's been developed in-house for some time. Often there are specific nuances in the way the dev team's working that can't be standardized with libraries and IDEs. This means a basic onboarding process for a dev team takes a month instead of a week, costing the client money, too.

Check how your software partner documents their work – this will come in handy, especially if you continue working on the product in-house or go to another vendor in the future.



JAGO PHIIPPENS
Xillio

We use a quality management plan and related tools, with a mix of agile and devops. Both quality assurance and quality audits are embedded in the processes with benchmarked metrics to be achieved, and several QA tools and automations.

*Make sure people realise **the process is the product** (and vice versa). If you can't make that stick, fix the process. Secondly, ensure you're working with a trim product set, with clear feature families. By doing both, your people can see that what they do matters, and will feel **empowered** and **freed** by process, instead of constricted.*

Leader's Advice

THE RIGHT TECH CHOICES BOOST CHANCES TO GROW

It's no surprise that managing technical debt is one of the most challenging jobs of CTOs. But it's not just their responsibility – dev teams should also be aware of the impact they have.

CTOs should coach dev teams about technical debt and empower them to speak up when they know about any negative implications

And we know from experience many developers are simply afraid of your reaction.

When you trust people with starting the work, trust them when they suggest choosing the right technology or solutions to paying off technical debt as well. Because they know their job best. Same goes for your in-house team.

Our experience shows it can be tough to convince a client to have a product backlog ready for longer than a few sprints. Which – unfortunately – means they're not able to recommend the right, future-proof solutions. So if

Our experience shows it can be tough to convince a client to have a product backlog ready for longer than a few sprints

you feel like this is about you, try to make sure the backlog is filled up for the next few months, and the team is fully briefed (like through regular backlog refinement sessions). Yes, it takes time, but it pays off in the long-run.

This ebook is about how a good, trusting relationship with people can help take your project to the next level. As soon as you build trust and focus on what's really important, you'll be on your way. Good luck!

Mike Jackowski, Pawel Jackowski,
ASPER BROTHERS founders

DIRECT CONTACT:

Pawel Jackowski

paul@asperbrothers.com
in pauljackowski

Mike Jackowski

mike@asperbrothers.com
in mikejackowski

CONCLUSION

You're here!

Which means you've (most probably) gone through the whole thing. And we really appreciate it. We also hope you found some useful tips you can implement in your company in the near future. If you do, let us know how it goes!

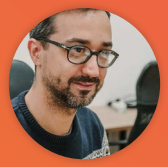
The challenges we mentioned in this ebook are pretty universal. But you probably have your own experiences with outsourcing software development you've accumulated over the years. If you're willing to share them with us, drop us a line, we'd love to hear what you think.

hello@asperbrothers.com

SOFTWARE CAN BE DEVELOPED WITH CODE

FIND MORE IN OUR BLOG

Time and Material (T&M) vs Fixed Price
asperbrothers.com/blog/time-and-material-vs-fixed-price/



Risk Management in Software Development
asperbrothers.com/blog/risk-management-in-software-development/

Software Cost Estimation – How We Do It
asperbrothers.com/blog/cost-estimation-in-software-development/

Team Extension Model or Dedicated Development Team
asperbrothers.com/blog/team-extension-model-or-dedicated-development-team/

How to write a RFP in software development
asperbrothers.com/blog/rfp-software-development/

THANK YOU!

We're thrilled to be part of a community where people share their achievements, experiences, and stories about overcoming the obstacles they encounter in software development outsourcing.

Abdul Wahab Majeed *Wirescan AS*
Alain Courbebaisse *Euronext*
Alexander Alten-Lorenz *Infinite Devices*
Amit Saraswat *FCM Travel Solutions India Pvt. Ltd.*
Andrew Robins *PM Connect*
Arshad Ebrahim *illion*
Bill Rue *BillRue.com*
Christine Martinez *Kalle*
Dmitry Valetin *Sharget*
Doug Austin *Special School of District of St. Louis County*
Eskil Skoglund *DolphiTech*
Glenn Kramer *xCures*
Farina Steinert *Technische Hochschule Lübeck*
Henry George *Expend*
Homayun Zahidi *Reelmetrics*
Jacob Thomas *Iconic Live*
Jago Phiippens *Xillio*
Jean Christophe Almira *KONE*
Johannes Drooghaag *Johannes Drooghaag*
Jon Geater *Jitsuin, Inc.*
Jon Walden *Blue Prism*
Magnus Eklöv *ftrack*
Marcel Dumont *OctoBoost*
Mike Bauer *Formpipe*
Minh Le *Meey Land Group JSC*
Murat Matur *Flen Health*
Paul Sassella *CHU Underwriting Agencies Pty Ltd*
Pinni Raveh *International Flavors & Fragrances*
Robin Rodgers *DEX*
Rubén Sopeña Vallina *Kabel*
Scott Petronis *Xcentric Consulting, LLC*
Srinivas Reddy *SSBA Innovations Private Limited*
Stig Olav Settemsdal *Siemens*
Viren Italiya *The Adinath Co-Op Bank Ltd*

ASPER BROTHERS

LET'S SOFTWARE

CONTACT US

ASPER
Spółka z ograniczoną odpowiedzialnością Sp. k.
Jana Pankiewicza 1/3
00-696 Warsaw
POLAND

hello@asperbrothers.com

asperbrothers.com/contact/

asperbrothers.com/blog/

REVIEWED ON Clutch 12 REVIEWS

WWW.ASPERBROTHERS.COM