TECH
LEADERS
HUB
by STX Next

Tech Leaders Hub: Management & Growth

Become a great leader by leveraging the huge experience of tech leadership experts who have spent years managing and growing teams.

STX NEXT
python powerhouse

Table of Contents

Introduction

Whether you're a tech leader, C-level executive, or (technical) manager—your job comes with a wide range of challenges.

In our experience, most of these challenges revolve around people: recruiting new hires, promoting current employees, managing team members, growing people's competencies, or scaling your teams.

We've made it our mission to help leaders such as yourself do right by your teams through shedding a light on the winning strategies of other managers and executives—which is why we started Tech Leaders Hub.

Tech Leaders Hub by STX Next is a comprehensive growth initiative for leaders with ambitious goals and a strong desire to be the best they can at what they do. We invite seasoned industry professionals to share their stories and insights, then turn that precious knowledge into useful, easy-to-read guides for you.

The following ebook is one such resource. First, a series of live interviews; now, a single source of tech leadership tricks of the trade.

We'll be focusing on solutions to problems with management and growth, generously offered to us by both internal specialists at STX Next and external guests who've worked at notable tech companies such as Egnyte, Allegro, OLX, or Acronis.

Whatever it is you're specifically struggling with as a tech leader, there's a good chance our experts have faced and overcome these same challenges. Then they told us exactly how they did it, after years of managing and growing teams for successful tech giants.

Read on to discover what they had to say and leverage their priceless knowledge to grow your business faster and lead your teams better.

How to Scale Tech Teams as a CTO

As your business grows, it becomes necessary to expand the teams within your organization to maintain productivity at a high level.

New teams and sub-teams may be formed, as well as new employees hired to get the job done. In addition, growing teams require different chains of leadership inside the company.

Being a tech executive in a growing organization, you'll need to hire and manage the tech leads and coordinate the team to achieve the goals of your business. This is quite a lot to figure out independently and requires a unique approach.

Luckily, we've teamed up with Alvaro Moya—an experienced CTO who is passionate about tech leadership—to share some of his expertise on scaling tech teams with us.

Read this chapter of our ebook to tap into Alvaro's insights if you're looking to create a high-performing team of leaders who will make autonomous decisions, hire great people under them, and maintain your company culture.

Introducing Alvaro Moya, a tech leadership guru

Alvaro Moya's professional journey clearly shows his passion for entrepreneurship, innovation, and leadership. He holds a first degree in Telecommunications Engineering, but he fell in love with computer science and development, pursuing a tech career instead.

He started his career as a Computer Vision Researcher at his alma mater, Universitas Carlos III de Madrid, where he researched, designed, and developed computer vision projects for corporate organizations. A few years later, he joined Coca-Cola as a developer.

Alvaro's passion for entrepreneurship and innovation has always been a strong drive for him. He nursed a dream and mission to lead. Thankfully, he had the opportunity to start a tech startup with some friends and filled the CTO role himself there.

Alvaro has co-founded several tech companies across different industries over the years. This has allowed him to grow in the CTO position in more companies in diverse industries. His passion for entrepreneurship has basically led him all the way from nothing to unicorn. He currently leads a remote distributed team of 60 people.

Alvaro identified the need to train tech leaders looking to scale their teams. In his words, "There are many boot camps for developers, but who is training the people who will manage these developers to enable the team to grow seamlessly and organically?" This led him to launch Lidr—an ecosystem for tech leaders where he and the Lidr mentors train the CTOs and engineering managers of the future.

This section is based on our live session with Alvaro. If you'd rather listen to what he had to say about the subject, you can watch it below:

Why scaling your tech team may be inevitable

Every organization desires growth, and with growth comes more responsibilities. For example, there may be a need to develop new products or customize existing products to meet the expectations of several stakeholders. Market demand for the company's products may increase exponentially. You'll also need to ensure that your company stays ahead of the competition.

During this phase, your organization may need to work with limited resources to maintain the growth level. There are also budget and time constraints to deal with. These challenges can only be met if you have the right team on board.

Unfortunately, people do not grow as fast as companies do. "Employee growth will always be linear and not as fast as a company's growth," says Alvaro.

Thus, it may be difficult to handle the increased workload and expectations with your current team. Therefore, you'll need to scale your team and appoint leaders to oversee the team to solve this problem.

4 important steps for scaling tech teams successfully

Growing a tech team is not a walk in the park. It requires proper planning and establishing frameworks to ensure that the company's activities are not disrupted and the revenue is not affected. Also, you need to make sure the company's culture is not lost in the wind while scaling your team.

Here are four important steps that will help you grow your team sustainably and successfully.

1) Be aware of your team's growth

Being conscious about your team's growth will help you make better decisions when scaling your team. For starters, you'll hire employees with a goal to train them to become leaders. You'll also factor it into your budget planning.

2) Start early

According to Alvaro, the best moment to train your team is yesterday; the second-best moment is today. Growing a team may take months of planning ahead. Therefore, it's best to start planning as soon as you notice growth in the market or product usage.

3) Find out the positions to be filled

It's best to find out where your team needs extra hands or leadership. This will enable you to come up with a plan to fill the vacuum. You'll also be able to maximize time and resources and ensure that your business does not suffer during the process.

4) Identify suitable candidates

Once you've identified the positions to be filled, the next step is to identify suitable candidates for these roles. It's advisable to hire from within the organization where possible. Go through the profiles of current team members and identify the best fit for the role.

How to spot potential leaders in your team

When scaling your team, it may be hard to decide who gets promoted to leadership positions and who doesn't. You don't want to make the wrong decision, and you'll need to justify your decision to the non-tech top management.

If you're at this crossroad, here are signs to look out for when looking for team members to fill leadership positions.

1) Go for self-leaders

To be a great leader, one needs to be able to lead their life first. Self-leadership is a great quality for aspiring and non-aspiring leaders. Self-leaders often embrace leadership roles that align with their passions, interests, and goals. If you don't appoint self-leaders, you'll end up modulating the employee's skills to meet the company's requirements, which requires a lot of effort.

2) Choose employees with the ownership mindset

Another great quality to look out for in leaders is the ownership mindset. If a team member always takes ownership of projects, they may be suitable for leadership positions. Employees who take ownership of projects can deliver results with minimal supervision and guide others to achieve common goals.

3) People who go beyond the nine-to-five may be a great choice

Team members who take up informal leadership roles outside the company, or work as freelancers, or have managed their startup in the past may be leadership material. It takes someone who consciously explores other interests beyond the regular nine-to-five to manage and oversee other team members and still perform their duties.

Besides, one acquires a wealth of knowledge when managing startups or leading teams that will benefit any organization.

How to make transitioning to leadership positions easier for newly appointed leaders

Transitioning from a developer to a manager or team lead can be challenging. It's common for new leaders to struggle in their new roles for some time. However, you can support them during the transitioning phase by doing the following:

1) Have the conversation early enough

It's important to inform employees that they are being considered for leadership roles within your organization early enough. This is best done during the hiring and onboarding stages. However, you can always have this conversation when the need arises.

When informing employees of the leadership positions they may take up within the organization, it's important to lay all the cards on the table. This will enable them to make the necessary plans for a smooth transition to such roles. Also, they'll be able to decide if they want to make that move or remain in their current role.

2) Create peer leadership opportunities

The best way to learn how to become a leader is by being one. You can prepare your employees to become leaders by creating peer leadership opportunities to practice and hone their leadership skills.

Peer leadership programs give employees the opportunity under little or no pressure. This makes leadership less frightening to them and the transition to actual leadership roles smoother.

3) Promote progressively

No one anticipates receiving more than what they can handle. Therefore, it's best to ensure that employees are aware of the career path within the organization.

For instance, one can expect to go from developer to team lead, team lead to a manager, manager to VP of Engineering, VP of Engineering to CTO. This progressive career growth allows employees to be better prepared for each role.

4) Train for soft skills

Leadership requires more than technical skills. A good leader should possess soft skills such as conflict resolution, communication, negotiation, empathy, time management, problem-solving, etc. You can support newly appointed managers by training them on these soft skills.

5) Cut them some slack when they fail

According to Alvaro, giving new managers psychological safety for failure will help keep them focused. In his words, “Ditch the old-school management, sink-or-swim technique and support them when they fail instead.” This way, they’ll learn from their failures and improve.

6) Create a healthy balance of their leadership and technical duties

Leadership roles come with unique responsibilities such as budget planning, making presentations, and hiring team members. Combining these duties with one’s hands-on developer duties can be difficult. Therefore, it’s advisable to create a healthy balance between technical and managerial skills.

For instance, a team lead may handle 70% technical work and 30% managerial duties, while a CTO’s duties may be 90% managerial and 10% technical. Of course, the ratios may differ in different organizations. What’s important is maintaining a balance to ensure that the employees are not overwhelmed.

7) Empower them to take charge

Empowering managers and team leads to take full responsibility for their teams and projects makes processes easier. Thus, you should delegate technical decisions to the managers and team leads.

Alvaro recommends setting the boundaries as long as they still get a blank check to take charge of their teams and projects. For instance, you can give them the business context to work with. This will stimulate them to think and make decisions with the business in mind.

What to do when you feel robbed of the time to properly train your team

In between planning, budgeting, and executing projects within steep deadlines, it may be hard to squeeze out the time to train your team. However, you don’t have to end up with a team that is unprepared for the work ahead. You can still scale your team in the following ways:

1) Outsourcing

You can partner with consultants and trainers to scale your team. Platforms such as Lidr partner with organizations to train future tech leaders. Lidr has many frameworks and resources that have been grouped to suit the organization's needs and company culture. Do check them out to learn more about their services!

2) Partner with HR

You can also partner with the HR department to train your team. For example, you can recommend employees for training based on the identified skill gap. This should be done continuously to ensure that the employees grow in their roles.

How to tackle large-scale projects while growing your tech team

Large-scale projects usually require certain changes, such as product modifications or the team migrating to a different infrastructure, to work efficiently. If not properly managed, these changes can disrupt your team's expansion and productivity. However, you can make large projects work without interrupting your team's expansion.

For starters, learn to rely on the team during this phase. Your role should be to coordinate and support them. Focus on high-level coordinating tasks like budgeting, setting up the project team, and setting deadlines. Trust your team to handle the fine details of the project.

Also, large-scale projects may drag on if not properly planned. To ensure that such projects are completed on time, focus on planning ahead of the project start date. Estimate the time, resources, and people needed to bring the project to fruition and work with the estimates.

How to preserve the company culture while scaling your team

It's easier to pass on your company's culture when working with a small team with employees reporting directly to you. However, as your team expands, the responsibility of preserving that culture falls on the managers and team leads. And if they're not prepared to uphold the culture, your company's core values may fall by the wayside.

You can avoid this by preparing middle management employees to become catalysts of the company culture with the following steps:

1) Hire for culture fit

Whether you're hiring managers from within or outside the organization, make sure to hire people who are a great fit for your company's culture. Candidates who display a good understanding of your company's core values will be able to transfer these values to their team members.

2) Embody the company's culture yourself

Humans generally learn from observation. Newly appointed leaders are more likely to look up to and imitate their superiors. Thus, a good way to ensure that your company's culture is not lost in the transition is to reflect the culture.

3) Remind the managers of the company culture

Don't stop telling employees about your company's culture. More importantly, when employees are taking up leadership positions in the company, you should remind them of the company's core values and educate them of their responsibility to uphold the culture as leaders.

How to bridge the gap between the tech leads and non-tech top management

Non-tech top management often have the final say on company projects. As a result, it may be difficult to work on certain projects without their approval, especially non-revenue generating projects.

However, you can bridge the gap and persuade top management to see things from your perspective with the following tactics.

1) Cultivate business skills

Non-tech top management teams are inclined to evaluate projects from a business perspective. Hence, you need business skills to present the projects to the top management from a business standpoint.

As Alvaro recommends, "Engage your team in mock projects using business case studies such as building a chatbot for a business." This will help your team gain high-level business skills, understand how tech applies to the broader environment and give them insights into the needs of the consumers and company.

2) Negotiate a tradeoff

While the project may benefit the company in the long run, budget constraints, limited resources, and time may make the project less likely to be approved. Hence, you should be open to negotiating a tradeoff with top management.

With good negotiation skills, you can present the benefits of the project persuasively. As Alvaro says, the management will be left with three options: to do it now, do it later, or not do it. As long as you've presented the project's pros and cons to the top management, they'll be able to make a decision.

Final thoughts on scaling tech teams

The strategies discussed in this chapter will surely come in handy when you need to scale your team. However, it is recommended that you prioritize the people in your company over anything else.

After all, it's the people who create products and make decisions that affect the organization. Also, your impact as a leader will be determined by their impact. Thus, it's advisable to put them first when making such decisions.

How to Help Your Developers Become "100x Engineers"

What are the odds you know a 10x engineer or have one on your team already? We bet you do. Our industry got sold on the idea of having fast programmers who can quickly turn Jira tickets to features delivered.

But the tech market is constantly evolving. In recent times, consumer demands have become more specific and less predictable. If you're looking to stay ahead of the curve, you'll need to reassess your team's capabilities and look beyond speed.

You may have enjoyed the benefits of having 10x engineers on your team, but do you know how to lead that team to achieve 100 times the results of regular developers? It requires an entirely different approach and a unique mindset.

Luckily, we can show you the way—with a little help from Zef Hemel.

Read this chapter if you wish to learn key insights on leading your team members from regular software developers to 100x engineers and find out what it takes to achieve 100x productivity with 100x engineering.

Introducing Zef Hemel and 100x engineering

Before we get to the matter at hand, we'd like to introduce you to our expert on the subject.

Zef Hemel has been in the software development game for a long time. He started coding at age 9 and obtained his first degree in Computer Science. He has held managerial positions in prominent tech companies. He was a Senior Manager at Egnyte, Director of

Engineering at OLX Group, joined Mattermost as Engineering Lead, and recently became a Platform Group Lead.

Oh, and by the way, he was once one of us at STX Next. He joined our company as a developer and rose to the VP of Engineering position.

Zef has led teams of 60–100 developers and added great leadership skills to his arsenal along his professional journey. In our recent live session, he shared tips that'll help you lead your tech team to reach 100x the impact of regular software developers.

Feel free to watch the conversation with Zef we based this section on below:

(Also, don't miss the mini case study of how Zef applied the 100x engineering mindset to building a new product for OLX Group—it starts around the 22:50 mark!)

What your tech team needs to be doing to become more productive

The first step toward leading your team to achieve 100x impact is identifying what makes your team less productive and nipping it in the bud. It's easy to overlook the productivity killers when your team is swamped with several projects. But this usually leads to wasting resources and time that could be invested in other projects.

If you want a productive tech team, follow these best practices we've outlined for you.

1) Avoid software development waste

Has your team ever worked on a product for several months that ended up never being used? Or conceived and built the “perfect” product only to discover that modifying an existing product could have been a better approach?

Such mistakes are not uncommon. However, if you review the steps taken by your team from when a product request is received to when the final product is delivered to the client, you may find out what went wrong. Most times, tech teams don’t bother to pause and ask, “Do we really need to build this right now?”

Answering this question before your team starts working on any project may help you explore creative ways to solve the users’ problems without any waste. It makes it easier for your team to figure out what’s needed and focus on delivering 100x results.

2) Involve engineers in pre-software development activities

Developers are ultimately responsible for ensuring that products satisfy users’ needs. Yet, they are often not engaged in the product research, end user research, vendor selection, and other pre-development processes.

Engaging your team members in these activities is a great way to get them invested and interested in the project. It makes them feel appreciated and willing to share their ideas on how to make the project run smoothly. Also, it’ll be a lot easier for them to build great products if they’ve been looped in and consulted from the beginning.

Teamwork is essential for productivity. Therefore, you should consult your team at every stage to achieve 100x more success. As the saying goes, “Coming together is a beginning, staying together is progress, and working together is success.”

3) Don’t ignore end user needs

It’s possible to get carried away with great ideas from your tech team on what the ideal features of a product should be. You can easily hold brainstorming sessions with your team to come up with several great concepts for projects. However, your team’s views on the product may be far from what the end users actually need.

Sadly, tech teams often fail to prioritize end user needs at the pre-development and development stages, which then leads to waste and end user dissatisfaction.

Products are made for the users and will only yield expected results if the users love and use the products. Therefore, you should always direct your team to figure out the end user needs when developing new products or refining existing products.

Who is a 100x engineer?

You may be wondering what all the fuss is about with the 100x engineer, anyways. Are they worth the hype and can they actually improve productivity? Yes. The 100x engineer is worth every commendation.

The concept of the 100x engineer is closely linked to the concept of the 10x engineer which dates back to the 1960s. A team of engineers conducted an experiment aimed at identifying the difference in developer performance between programmers with access to computers and programmers working with the traditional pen and paper.

The results from the experiment showed significant differences in the execution speed and the size of the programs produced by the two sets of programmers. This birthed the 10x engineer concept which focused on the execution speed of programmers.

For a long time, the concept of the 10x engineer dominated the industry and appealed to employers as employers focused on hiring fast programmers.

Let's face it: the 10x engineer is perfect if all you need are engineers who can churn out code quickly. It's possible to achieve more with fewer engineers if one had a team of 10x engineers coding lightning-fast. 10x engineers can easily write programs using a defined concept or following the specs in a Jira ticket.

But what happens if the concept is flawed and the feature shouldn't be delivered in the first place? Who's going to figure that out? Not the 10x engineer whose major concern is delivering code. Hence, you'll need engineers with the 100x mindset.

100x engineers offer more than their coding ability. They think about the project and focus on understanding the business requirements and problems that the product is intended to solve. In the end, the 100x engineer may achieve more in less time than the 10x engineer.

So, how are 100x engineers different? Here are some of their key attributes.

1) Self-direction

100x engineers approach tasks with the ownership mindset. They are not lone wolves and they understand the importance of teamwork and respect leadership. 100x programmers manage their own actions, take charge of their work, and hold themselves accountable. They can easily make decisions that move the team closer to their goals and are not afraid to own up to their errors.

2) Innovation

100x programmers are not afraid to challenge the status quo. They come up with creative ways to achieve more with less and use various unique methods to solve challenging tasks. They don't mind narrowing the scope to meet the client's immediate needs. In fact, they may come up with several alternative ways to achieve the same objective.

3) Business-orientedness

100x engineers see themselves as more than programmers; they consider themselves partners to the business. This makes them eager to uncover the business context for each product. The 100x engineer seeks to understand the need for the product to build the best product for the team.

4) Negotiation and communication skills

100x engineers possess great negotiation and communication skills. It's not easy to convince the product manager to narrow the scope of the project or ask them to perform user testing before implementation. Effective negotiation and communication skills are essential to convince tech leads, product managers, and other team members to approach projects from a different perspective.

5) Competent technical skills

In addition to the soft skills, a 100x engineer should have the right technical skills. After all, it takes a competent engineer to apply the 100x mindset to the product development process.

How to lead developers to adopt the 100x engineer thinking

The 100x engineering mindset is desirable. However, the implementation may prove difficult. You may find that the engineers on your team would rather receive Jira tickets and code away. Also, businesses are often pressed for time and expect their projects to be delivered on time. Thus, it may be difficult to convince your team members to apply the 100x engineer mindset.

Fortunately, it's not impossible. Here are some techniques that will help you lead your team of developers to apply the 100x engineer thinking.

1) Emphasize the project objectives

Rather than dumping the scope of the project on your developers, you should get them to focus on the project objectives. This way, they'll be more inclined to think about several ways to achieve the objectives.

2) Encourage them to take risks

Developers may have ideas worth exploring, but feel reluctant to share them because of the risks. Encouraging developers to take risks will make them innovative and eager to execute projects with the 100x mindset.

3) Motivate and reward developers applying the 100x mindset

When you reward team members who have adopted the 100x approach, you encourage others to do the same. You can motivate them by providing any technical assistance they need to explore various avenues within the project.

4) Embrace diversity

You'll find engineering-focused, regular coders as well as product-focused, 100x engineers on your team—and it's good to have both. The regular members of the team can work faster with guidance and support from 100x engineers. Thus, you should embrace your diverse team.

How to hire 100x engineers

Building a team of 100x engineers extends to the hiring process. It'd be a lot easier if candidates had "100x engineer" boldly written on their foreheads. Sadly, this is not the case and a 100x programmer can be difficult to identify out of several qualified engineers. However, it's possible to spot a potential 100x programmer when hiring. Here are some tips that will help.

1) Evaluate the candidate's understanding of your business

Many candidates express enthusiasm about a role, but aren't familiar with the organization's business. A 100x engineer is equally interested in the business. They know they have the essential skills and can acquire other skills needed to perform their duties. Hence, they are curious about the projects the organization handles.

One good practice is to ask candidates about their views on the recent activities of your organization that are public knowledge.

2) Look out for the key attributes of 100x engineers

The recruitment period is a good time to identify the key attributes of 100x engineers we've discussed in this chapter. Test for such attributes as well as technical skills. Oftentimes, candidates will display more technical proficiency than soft skills. You can find out if they possess the attributes of the 100x engineer by asking questions such as:

- When was the last time they challenged a program requirement?
- Do they criticize program requirements?
- How do they conduct end user research?

3) Get to know the developers personally

The 100x mindset is transferable and cuts across different aspects of a person's life. Thus, you can spot potential 100x programmers from their personal stories. It's advisable to look beyond technical and soft skills related to the job role. Ask them questions about their hobbies, passion, interests, and personal achievements.

How to apply the 100x engineer mindset to a project in flight

The best time to apply the 100x engineering approach is when you have a new product or product refinement request coming in. It's easier to get things right from the beginning. However, you may not have the opportunity to be there at the start of every project.

If you're faced with a project that's already in progress, you can still maximize productivity by applying the 100x engineer approach. Here are some strategies that will help.

1) Conduct end user research

The best products are not necessarily the best designed or engineered products. The best products are products that users will use and enjoy using. Thus, it's important to prioritize your end users in the product development process.

Before jumping on any project, get your team to identify the target users for the product, what problems they're trying to solve, and how the product can solve their problems. This will ensure that you have a vivid picture of what is being built and why. It also spurs creativity.

You can also get your team to put themselves in the position of the end users and build something that solves their problems.

2) Understand the priorities

Generally, product requests tend to be vague and merely specify the problem intended to be solved plus the must-have features. As a 100x engineering lead, you're responsible for identifying the priorities and communicating them to your team.

Figuring out the priorities will make you consider alternative ways to achieve the expected results. You'll be able to reduce the scope, come up with a minimum viable product, and reduce time-to-market.

3) Build lean products first

Rather than build the perfect product, focus on developing a prototype to test the market. Adopting a build-measure-learn product development approach gives room for experimenting with lean products or prototypes until your team figures out what the clients want.

You can improve and update the prototype using feedback from ideal customers. This way, you'll be able to build a final product that the clients will love and use, reduce software development waste, and maximize your resources.

Be sure to check out our live stream recording for a short case study of how Zef Hemel implemented lean startup thinking when building a "Tinder for jobs" for OLX Group—using a special technique he likes to call "pop-up driven development."

4) Take advantage of usability testing

A good way to ensure that your product solves the problems of the target users is to engage ideal users to test the products. Usability testing provides qualitative data needed to refine products during and after product development. It can also increase the demand for the products and increase business productivity.

You can conduct usability testing with your organization's existing platforms or create a minimum viable product to do so.

Final thoughts on software development productivity with 100x engineering

The 100x engineer is ten times better than the 10x engineer and 100 times better than the average programmer. There are many benefits of having them on your team. Luckily, you can always guide your team to approach projects with the 100x engineer mindset using the tips we've covered in this chapter.

It's not a day's work, but it's important to start instilling the 100x culture in your team right away. You'll increase their productivity and start building products that people will actually use, which makes the effort worthwhile.

When to Promote Your Best Developers to Managers

If you have a team of talented developers, filling managerial positions in your tech company may seem like an easy task. You may think that you'll be spoiled for choice and not have to spend time and money shortlisting and interviewing external candidates.

Unfortunately, that's not always the case, since good coders don't always make good managers. Also, making the wrong promotion decision can reduce a developer's productivity and, by extension, your company's productivity. So if you're considering promoting your best coders to managers, you may need to rethink your decision.

Luckily, we had a lengthy chat on this subject with Hazel Olivier, CTO of Nimbla, during one of our recent Tech Leaders Hub Fintech series. Hazel shared invaluable insights about promoting developers to managers that might save you from making the wrong decision.

If you're struggling with this choice and want to get things right, keep reading to find out how to identify potential managers in your team and promote developers to managers without reducing their productivity.

Introducing Hazel Olivier, our technical management expert

Hazel Olivier has over ten years of experience in senior and executive management roles and over 15 years of experience managing technical delivery teams.

She specializes in building and scaling complex engineering teams across multiple industries. She is also a natural leader who motivates and supports teams to maximize performance and profits.

Currently, she is the Chief Technical Officer of Nimbla—a fintech company that offers small invoice insurance services to businesses. Nimbla protects small business owners against customer default in an interestingly easy manner.

The Nimbla platform uses algorithms to assess the risk of a customer defaulting in payment or predict if a customer is likely to claim insolvency. Nimbla's clients can link the platform to their accounting system or upload process invoices for insurance on the go. Nimbla also assists business owners in recovering sums from defaulting customers.

Nimbla has established partnerships with Barclays Bank, Starling Bank, and a couple of large brokers in the UK. Recently, the company launched an EU offering allowing UK firms to trade within the EU. As Nimbla's CTO, Hazel manages and drives the value of the technology within the organization and helps Nimbla remain competitive in its vertical.

The following chapter of this ebook covers our discussion with Hazel extensively. However, if you'd rather watch the live session, here's the video of the recording:

Why you shouldn't promote your best developers to managers

People who excel in their fields often become the go-to persons for colleagues and team leads on matters related to those fields. And when there's an opening for a higher position in their field, they naturally come to mind. But there's no guarantee they will excel in a managerial role simply because they're good at what they currently do.

As Hazel says, "Don't just promote your staff because they are good at what they do." There's a difference between being a developer, a team lead, and a department lead. Managerial positions require additional skills that the best coders in your organization may lack. Also, people may be comfortable being just coders and feel overwhelmed and unmotivated if you promote them to a managerial position. In some instances, they might even end up leaving your company.

It's advisable to ensure that your team understands that they don't need to move up the management ladder to progress in their careers. There should be flexible roles for people who don't want to be managers. For instance, they could be subject matter experts.

However, if you've identified that your best coders are interested in taking on managerial roles, you can promote them. Hazel recommends supporting newly appointed managers until they embody their new positions. Thus, you should provide them with the resources they need to excel in their new roles.

5 tips for choosing and preparing the right employees to become managers

Employees shouldn't be promoted to managers on a whim. Therefore, it's essential to set up a rolling plan ahead of the promotion date or any anticipated promotion. This way, you'll have enough time to identify the right team members to be promoted and figure out the best roles for them.

Here are some tips for identifying and preparing employees for managerial roles:

1) Learn your employees' career interests

People have different career ambitions. For example, some employees may want to transition to new roles in a different field. Some may want to explore a career in academia, and others may want to start their own businesses.

Thus, not everyone will be interested in becoming a manager, and you shouldn't work with this assumption. Instead, discuss this with your team members and figure out their career interests.

This way, you'll be in the best position to help them achieve their goals. But, more importantly, you'll find the employees who are interested in assuming managerial roles and narrow down your list.

2) Train for managerial roles

Ensure that you have a continuous professional development (CPD) plan for employees and tailor employee training to their career path.

Before developing a CPD plan for each employee, ask them about their interests. Then, when you've figured out your employees' career interests, you can support them in developing the skills they need to fit into their desired roles.

Specifically, you can focus on improving the soft skills of employees interested in becoming managers. This will make transitioning to a leadership position much easier for them.

3) Job shadowing

Another way to identify and prepare your team members for managerial roles is job shadowing. Some people learn quicker by doing things and not observing. In Hazel's words, "Some employees may get bored from watching people all the time." Hence, engaging your team members with managerial tasks may be a great way to identify potential managers.

Job shadowing is essentially assigning managerial responsibilities to team members for a short period of 1–2 weeks. During this period, you'll be able to see how they perform in the role. Also, they'll be able to decide if they want to become managers or not.

Job shadowing can also be implemented effectively in a remote setting. For instance, you can ask an employee to learn the goals of 2–3 people on their team and revert or some other exercises that force them to think and experience what managing people would feel like. You should discuss their findings from the tasks later to find out what they think about such responsibilities.

4) Set goals for your employees

Goals help us achieve various milestones in our lives. It could be managing your finances, dieting for weight loss, or aiming for a promotion. You're more likely to get the results you desire if you set goals for yourself. Also, it's been proven that people improve their performance if they know what they're doing in the workspace.

When it comes to setting goals for team members, it's essential to identify areas where the team member needs to improve and help them set performance goals in these areas.

Then, you can collaborate with them to identify key areas they can implement in a quarter, six months, or year.

Finally, you can link their goals to your company's vision. For instance, if the company's goal is to make more revenue in a given year, identify how the employee can help the company achieve this goal.

Goal setting helps employees see themselves as essential members of the organization as well as develop coordination and planning skills, which are necessary management skills.

5) Conduct periodic performance reviews

Beyond setting goals, you should also conduct reviews to evaluate the employees' performance periodically. Performance reviews help employees identify their strengths and weaknesses. Also, it enables you to know who's better prepared for management positions.

If you don't conduct reviews periodically, your team may not improve and achieve their goals. It'll also be difficult for you to make an objective promotion decision without a performance record.

Performance reviews make promotion decisions more objective. For example, an employee with consistent poor appraisals won't feel unappreciated if another employee with good reviews is promoted ahead of them.

How to decide who to promote when you have multiple candidates

Sometimes, you may find more than one person in your organization interested in a managerial role and qualified for the position. In such instances, it's advisable to come up with a process to select the right candidate.

Hazel recommends using the same process you'd use if you're recruiting externally. Thus, you can review their CVs and put them through the interview process. This way, you'll be able to find one person that stands out.

According to Hazel, the goal should be to "give them a fair shot and make them feel appreciated even if they don't get the role." Putting them through an interview process makes the selection fair.

5 common mistakes to avoid as a manager

As a manager or newly appointed manager, your activities can affect your team's performance a lot more than you know. As Hazel puts it, "If you see a team that isn't as performant, look into the management and how they are dealing with those teams."

If you're a manager hoping to find your predecessors from your team or a newly appointed manager trying to get on the right track, here are some common mistakes you should avoid to effectively manage your team:

1) Isolating yourself from your team

You'll often come across people who are more interested in being a techie and don't want to meet many people. Sometimes, they get enthusiastic about what they're creating and forget about everything else. As a manager, you can't afford to be so.

Managers are often the first point of contact for their team members. In addition, team members often relay their worries, concerns, and complaints to their managers. Thus, as a manager, you should be open to interacting with people and solving their problems.

As Hazel says, "Make your team feel like they are not just numbers, but the essence of the company." People will feel this way if you interact and engage with them often.

2) Setting the wrong example

If you're hoping to hire managers from your team, you need to set the standard by being a good manager yourself.

People often learn by observing other people. Hence, it's important to display the skills and attributes you want your team members and future managers to possess.

In Hazel's words, "Good management needs to be timely." It would be best if you led by example early on, guiding and supporting your team members to achieve the company's objectives.

3) Not providing feedback

People can't fix things when they don't know that something is wrong; they'll just keep making the same mistakes.

Most employees will appreciate regular guidance and feedback from their team leads, and if they aren't getting feedback on their work, they won't be able to develop and grow into the positions you want them to fill.

Also, great feedback boosts your team's morale and improves their productivity.

4) Not giving your team autonomy

According to Hazel, people who manage effective technology teams often give them time to grow and encourage them to use their initiatives. There will always be high-level decisions exclusively for leaders. However, you should learn to trust your team to make some decisions by themselves.

If you hire people due to their skills, you should trust them to use those skills with little or no guidance. This will build trust in your team and help your team members become more responsible.

5) Not celebrating wins

It's easy to focus on solving problems and forget to celebrate your wins. However, as a manager, you should make a conscious effort to celebrate your team's successes and manage failures diplomatically.

When your team members make mistakes, don't attack the person. Address the error instead and offer solutions to the issues. Also, try not to jump from one problem to another.

Final thoughts on promoting your best developers to managers

In the tech space, managers do a lot more than coding. Therefore, being a good coder or holding a senior position doesn't automatically qualify your team members to be promoted to managers. Managers are responsible for coordinating people and resources to deliver on the company's goals and objectives, and not many employees can do this.

Also, employees are every organization's greatest asset. That's why they need to be adequately managed to become productive and efficient managers if they choose to become those.

As a tech leader, your role includes identifying the employees with the right potential and training them for managerial positions. If you use the tips shared in this chapter, making promotion decisions and managing your team will be a lot easier.

Managing Large Teams at Hypergrowth Companies

A hypergrowth company is a dream come true for most founders and investors. But as a manager or employee, you may prefer the early days of working with a close-knit team, grabbing lunch with the CEO, and sharing the same office with other departments.

However, scaling the teams within an organization may become inevitable as your company grows. And while working in a large team may not be your strong suit, you may be assigned a new role or set of responsibilities that will require you to manage and interact with many employees.

If you're wondering what to expect when your company hits the hypergrowth level or you're trying to figure out how to manage your fast-growing team—you're in luck. We recently hosted Evgeny Sharypin, Senior Development Manager (Scalability) at Sendcloud, during one of our Tech Leaders Hub series, where he shared great insights on managing large teams in hypergrowth organizations.

Read on to learn Evgeny's unique strategy for tackling company hypergrowth, what to do when conflicts arise within your team, and tips for managing large teams successfully—even as a newbie manager.

Introducing Evgeny Sharypin, our large team management expert

Evgeny studied telecommunications with a specialization in applied cryptography at the Technical College of Telecommunications and Informatics in Moscow. During his third year, he joined Flexline-N, a small-scale internet services provider in Russia, as a network engineer and transitioned to a software engineer/system administrator role in the company within a short period of time.

Once he graduated from college, he joined Acronis, one of the biggest IT enterprises in Russia, as a software developer/build engineer working on different projects. Notably, he set up a CI/CD pipeline for one of the projects that cut the time to build from 12 hours to 40 minutes!

Shortly after leaving Acronis, Evgeny joined Sendcloud as a senior backend developer and the company's 73rd employee in 2019. A few months later, the company experienced hypergrowth as a result of the pandemic. The spike in orders forced Sendcloud to quickly grow from 73 employees to over 300 between 2019 and 2021. The backend development team also grew from 20 people to 6 teams by the end of 2020.

Evgeny is currently the Software Development Manager in charge of scalability at Sendcloud, where he manages three teams with the aim to sustain the company's rapid growth phase. His love for interacting with people and fascination with processes for systems made him interested in becoming a manager. Thanks to adopting the "process thinking" method by applying processes when managing people, Evgeny's approach to management is quite unique.

This chapter covers our discussion with Evgeny extensively. However, you can also watch the live session below if you'd rather listen to Evgeny speak on the topic himself:

How to apply “process thinking” when leading a team

Just as you adopt processes when building tools, you can adopt “process thinking” when leading a team. However, humans can be fuzzy and you’ll need to refine the processes to make them work for people. This will help your team become more productive and achieve the desired outcomes.

Here are some processes you can adopt when leading a team.

1) Lead with context

As a manager, you may be tempted to give your team members detailed briefs for all projects. While this may speed up execution, your teammates may become overly dependent on you. You’ll have to constantly be in charge, which can be quite tedious and make projects move a lot slower in your absence.

Evgeny advocates for “leading with context instead of leading with control.” Leading with context entails providing just enough details to enable your teammates to figure out the right processes to reach their goals. Training your team to think about the expected outcomes will make them come up with innovative ways to achieve them.

Evgeny recommends creating an atmosphere where employees, managers, adjacent departments, and team members think more about the outcome than the process. He also shared practical examples of leading with context, as opposed to leading with control, which we’ve highlighted below.

- **Leading with context when planning projects**

If your team needs to plan for the coming quarter, you may choose to be in control by liaising with several people to come up with the plan, then sharing the plan with other team members.

But if you want to lead by context, you’ll focus on sharing a mind map of everything there is to be done by the team and describing the expected outcomes for each task to enable the team to come up with the plan themselves using the mind map and outcomes described.

- **Leading with context for big projects**

When working on big projects, you may be tempted to break the goals into smaller pieces and assign the pieces to different team members. But you can lead with context if you help your team understand the context and expected outcomes, then ask each team member what they can do to make the project successful. This way, you'll be engaging everyone in the process.

2) Engage your team when designing processes

One mistake managers often make is designing processes that team members don't want to adopt. Employees are more productive if they enjoy working with the processes they receive. As Evgeny puts it, "If you give people a say on how they work, they will enjoy the process and be more engaged."

Thus, it is advisable to engage your team members in the process selection and design stage. A good way to do this is by telling them the problem and encouraging them to come up with processes for solving it. After the processes have been designed, the team can agree on the one to be adopted.

What to do when stakeholders disagree on the process to be adopted

Considering your team's input on processes may be challenging. Certain stakeholders may prefer one step to another, making it difficult to consider everyone's input. If you're faced with this problem, here are some tips for resolving the conflict.

1) Find out the motivation behind the stakeholders' preferences

Companies, teams, and people often fail to question their decisions. This can rub off negatively on projects and make execution slower. But you also don't want to ignore people's preferences. You need to consider dissenting views to keep everyone engaged.

A good way to do this is by asking the team members for the reasons behind their choices. You may be able to convince them to join the majority if you understand their concerns.

2) Be diplomatic

As a manager, you need to maintain positive relationships with stakeholders. One way to maintain good rapport with your team members and other stakeholders is by being diplomatic.

As a manager or team lead, you may have to reject the stakeholders' preferences, but this must be done diplomatically to ensure that no team member feels left out and make them willing to proceed with the selected process.

5 useful tips for managers and tech leaders in hypergrowth companies

As a tech leader, the rapid growth stage can be scary and interesting at the same time. There will be a lot to do with less time to execute. You may become overwhelmed if you can't rely on your team to deliver and support you in this period. Hence, it's advisable to have a plan for when things become really busy.

Here are some tips for navigating the hypergrowth period as a manager:

1) Diversify your team

When a company enters the hypergrowth stage, its talent needs to change, too. You may find out that you need more senior developers or sales executives to drive growth. Therefore, it's best to start diversifying your team based on the company's needs.

If you don't begin to diversify your team, you won't be able to handle new problems that may come up. Plus, according to Evgeny, "During hypergrowth, new problems appear faster than you can resolve older problems."

When diversifying your team, look out for candidates with soft skills, such as strong leadership qualities. Also, consider hiring people who have overseen bigger projects in the past.

2) Properly integrate new employees to the team

According to Evgeny, it may take 3–4 months for talented developers to become fully integrated into their roles, while most people need 8 months to a full year. People get up to speed with tasks quite differently and background differences may also make it difficult for people to adjust to a new environment. Thus, within the first half of the year, focus on integrating new employees properly.

During this stage, encourage the employees to figure things out by themselves and progressively assign more responsibilities to them. This will make them less dependent on you and help them become self-leaders.

3) Avoid being paternalistic

People often adopt a parenting approach to managing people. However, this sentiment is wrong because parents are directly responsible for children while children can't be responsible for themselves until they become adults.

As Evgeny says, "If you treat adults like children, they may become mere executors and feel like they have not been given much responsibility." Also, they may not fully understand the consequences of some of their actions, which will make it difficult for them to learn from their mistakes.

Avoid shielding your team members from the consequences of their actions and encourage them to admit their failures openly. This will help them become less dependent and grow into self-leaders over time.

4) Make data-driven decisions

Even when you've built the right team, it may still be challenging to deliver on projects when you have more tasks on your hands than the team can handle. You may receive several time-bound requests from different clients. In such situations, you'll need to prioritize which task to attend to urgently.

A good metric for making these decisions is data. For instance, you can prioritize the project from the client bringing in more revenue. This will help you deliver real value to the stakeholders in the end.

5) Prioritize projects that enable exponential growth

As demand for your company's products or services increases, your team may be unable to carry out all tasks in-house. You should consider outsourcing certain tasks to your business partners.

In deciding what projects to prioritize, Evgeny recommends considering projects that enable exponential growth for the company. For instance, you may decide to build an MVP to reduce time-to-market for certain products.

How to survive managing large teams as a new manager in a hypergrowth company

As a new manager in a hypergrowth company, you're faced with new responsibilities that need to be executed at the company's growth rate. This may be difficult to navigate and you may find yourself struggling in the first few months.

If you're in this situation, Evgeny shared three important tips that can keep you going:

1) Understand that you're back to being a newbie

In Evgeny's words, "No matter how many years of experience you had as an engineer, the moment you become a manager, you have zero experience." You're back to being a newbie—which is both good and bad.

According to Evgeny, it's bad because you don't know what you're doing and good because you'll need to be a bit easier on yourself when you're struggling. So don't beat yourself up when you make mistakes.

2) Know that you're responsible for more than deliverables

Being a manager is a huge responsibility. It gives you agency over people's lives and livelihoods. You may be forced to make difficult decisions that affect people's overall well-being. Hence, it's advisable to always be firm, considerate, and diplomatic.

3) Seek help

As a new manager, you'll need all the help you can get to navigate your new role easily. Evgeny recommends finding a good support network of leaders, reading books, or finding a mentor who is ideally a manager that doesn't belong to your immediate vertical, so that there's no conflict of interest.

Final thoughts on hypergrowth company large team management

If we were to summarize what leadership in a hypergrowth company requires, we'd go with Evgeny's number-one tip for managers: be kind to yourself and others.

Leading a large team during hypergrowth requires a less systematic approach. People are less likely to deliver the expected outcomes if they're treated in a systematic manner. But with the strategies shared here, you should find managing large teams less intimidating.

Why Your Company Needs Processes

In a well-oiled organization, every team member should be dealing with processes on a daily basis. It's not just an issue for managers. Everybody, from top to bottom, should be able to contribute to how processes are organized.

Processes, workflows, procedures, SOPs—they're like algorithms in code. The difference is that these algorithms are used by people, not machines.

At STX Next, a variety of processes helps us make sure that work goes smoothly. Working with different companies, we observe how they approach process management, and we use this knowledge to improve.

Even small teams have processes, although they might not think of them that way. Anytime you say, "This is how we do things on our team," you're already talking about a process.

Read on to find out which tasks to turn into processes, when to start thinking about processes, and how to introduce them in a way that will improve your team's productivity and turn your work into smooth sailing.

Or if you'd like, you can watch Marcin Zabawa, our Director of Core Services, share his process management tips during this live session:

Process management is valid, but processes have a bad reputation

Any standardized way of completing a repeatable task is a process. But people often don't think about it this way.

Procedures are often seen as a rigid, corporate-only approach to work. If you have to work with standardized processes, you're just a tiny cog in the big machine, and you can't use creativity in your daily work, right?

The truth is a bit different. A well-designed, properly managed set of standardized workflows actually enables your team to be more creative, and add more value in their daily work.

Another great thing is that processes enable your team to exchange best practices. By default, you want your process to encapsulate the best way to do something. The most efficient A-to-Z steps to finish tasks like building a feature, fixing a bug, or recruiting new people.

People in your team will need to discuss how to do their tasks, share their ideas and methods that work for them. That brainstorm leads to finding the most efficient way to accomplish a task.

And that's what processes are all about: finding the best way to do a task, and making it easy to replicate.

Two essential reasons for introducing processes

There are two organizational aspects that create the need for processes: complexity and automation.

Work, especially in tech companies, is getting more complex. If you can figure out the best way to do a repeatable task, and organize it in a simple workflow, this frees you up to think about other things.

For example, when you have a detailed process for building new features, your team can get to work quickly by following steps—they don't need to rethink how to add a new feature every time.

This doesn't mean creating a conveyor-belt type of workplace—tech companies aren't factories. Modern work is complex, especially cognitive work. In tech, pretty much all you do are complex cognitive tasks. The mental burden is high, and reducing it is crucial to improve productivity and avoid burnout. That's what processes are for. They reduce the mental burden in everyday work.

The other big need for processes comes from automation. You can automate a lot of your repeatable daily (weekly, monthly, yearly) tasks, but you need to define them in detail to enable automation.

First you have to define the input: a new feature is built and needs to be tested. Next, what needs to be done, step-by-step: extensive tests are run on the feature, the code gets refactored. Finally, the intended output: a new feature that can go to the client/project manager to be evaluated.

Defining the input, necessary steps, and output is exactly what you do when you design a process.

By reducing complexity and automating tasks, everybody on your team can be much more productive. But it's easier said than done. First you need to prove to your team why processes are important.

How to build an organizational culture that accepts processes

You don't want to run a dictatorship in which you define the process, because tech companies don't thrive in such conditions. As a manager or CEO you wouldn't be able to work out, and continuously improve, all of your company's processes. It would require an impossible amount of time, and turn out worse than if you support your team members in organizing their own work.

In order to be accepted by the team, and to stick in the long-term, processes need to be adaptive. There needs to be a built-in mechanism for improving them over time. That mechanism is a feedback loop.

The feedback loop doesn't just happen. It evolves from an organizational culture that's built around feedback and cooperation.

From day one on the team, people need to know that they can influence their workflows. That their processes aren't set in stone, and that if they have feedback, it will be heard and considered.

As a manager, you should encourage people to think creatively about processes that they're involved in. If they think that change is necessary, they should be able to voice their opinion, and you need to hear them out!

Don't direct people to internal wikis or docs. Give them time, explain the company as a living organism and show where the process fits in, and how other people depend on it.

Imagine this: one of the developers comes to you with a suggestion to improve the QA process. What do you do? A simple mental simulation is a great way to address this.

Discuss how the change would play out. Ask directed questions—how would this specifically affect other team members? What would need to change, and how much would it cost? What about edge cases, would the new process hold up?

Your team will see that you care about their opinion, and that their suggestions will always be considered. If you want a tight-knit team, people need to see that you're all about teamwork, not shut-up-and-do-as-I-say work.

Another hugely useful habit here is providing positive feedback as soon as people deliver value, make a good decision, or complete a process quickly and efficiently. As humans, sometimes we get so caught up in focusing on mistakes that we forget to appreciate accomplishments.

To create a thriving culture of feedback, you should always give positive feedback whenever good things happen and achievements are made. That's not all—you need to also include that positive feedback in regular growth review meetings. Show your team that you care about their contributions.

Apart from helpful habits, there are tools you can use to make processes easier. We use ProcessPlan. It's really simple. Process designers have a visual representation that looks kind of like a mind map. You just connect the steps and define what happens.

On the other hand, from a team member perspective, they see a task list—they always know what to do next, and have an easy way to move the process forward. You literally click a button in an email to signal that you completed another step of the process.

Then there are personal productivity tools, like OmniFocus or Things3. They're kind of like to-do lists on steroids, which you can use to standardize your own workflows, and make your daily work easier.

A big part of finding the best tool to manage your processes are templates. Process templates, to-do list templates—they make it a lot easier to iterate new versions of your project, and reuse processes that work.

There are tons of tools for process management out there, so it's best to do your own research and find the ones that suit your needs.

The difference between helpful processes and rigid bureaucracy

How do you know if processes are still helping your team, and not making their work harder than it should be?

It's easy to create so many procedures that people get lost or even paralyzed. But it can be avoided as long as you maintain a continuous improvement mindset. Feedback needs to be an integral part of your culture.

When your culture is based on continuous improvement and feedback, then your team becomes the very mechanism that will prevent processes from becoming too rigid.

Whenever a process adds complexity instead of reducing it, people will stop, rethink things, and suggest changes. If processes are like algorithms, then this is like refactoring your organizational code.

A designer would say that perfection is not when there's nothing left to add, but when there's nothing left to remove. This rule works perfectly for process management.

Which tasks are most worth turning into processes?

Most tasks that are repeatable can be enclosed in a process. If you do it regularly, you might want to think of making a process for it.

Another indicator of a task that can be enclosed in a process is when multiple people have to cooperate in doing something—especially if they don't work together on a daily basis.

Here's a variety of examples:

- **Recruitment and onboarding**

Where are the best places to look for candidates? What are the best ways to test their skills? How do you find out whether they fit your team? Once you hire someone, how do you make them feel at home as fast as possible?

With a great onboarding process, you can get a new team member ready to add value within a week. Without any onboarding process, it can take months until someone is at full capacity.

- **Giving someone a promotion or pay raise**

It involves a manager/team leader, an HR person, someone who does finance, and sometimes legal (in case of contract changes). A perfect opportunity to ensure smooth cooperation between departments with a solid process.

- **Incident handling**

When there's a security incident, it requires cooperation between the person who found it, the incident manager, an admin/SecOps person, top management, lawyers, sometimes customer support and PR.

- **Offboarding after finishing a project**

A good offboarding process will make sure that you always collect feedback from key stakeholders, and close the project in a way that there are no loose ends.

- **Quality assurance**

This is a perfect area for creating processes. Without any processes, it will always take a lot of discussion—how it should be done, at which stages, should it take place at team level (if multiple teams work on a product), or at a higher level, and so on.

- **People evaluation and team growth management**

People evaluation is an interesting case because this area calls for a relatively loose, adjustable process. You don't want to measure everyone's growth the same way. People are different.

Your process should consist of basic steps for talent management—like regular meetings, goal-setting, reviews—but the approach for each team member during these steps should be individually tailored.

Within the realm of software development, release planning and crafting product roadmaps are just two examples of activities where a codified process could pay off nicely.

We could go on with more examples. But the bottom line is that more likely than not, there are people in your organization right now who are performing tasks that should be turned into more formalized processes, to the benefit of all.

How do you reconcile Agile with process management?

We all want to be as Agile as possible, and the Agile manifesto is the golden standard to look up to. At first glance, the manifesto seems to argue against processes. One of the points states, "We value individuals and interactions over processes and tools." Did the authors of the manifesto mean that you should avoid processes?

Not at all, there's no contradiction between the Agile manifesto and process management. The manifesto doesn't say anything about *not* valuing processes and tools—it only tells you that people are more important than processes.

Developers and tech managers know this well. For example, your development team might have set up automated deployment of code to the production environment. That's the process that works. But when developers need to deploy something manually, they should be able to do it, too.

Whenever there's a situation that requires a unique approach, the process can be changed, or omitted. Common sense should always be able to trump the process.

Processes are meant to be improved on, changed, and made to fit your team. Even when you consider a standard like Scrum, every Scrum team uses their own version of it. The fundamental laws of Scrum are there, but it has to be adapted to your unique context.

Tailoring processes to your team's needs is nothing extraordinary—it's the most natural approach.

When should you start thinking about adding a process?

We work with companies of all sizes. When we cooperate with startups, we see that they often don't have standardized procedures. It makes sense. Until they start scaling up, they probably don't need a process, because the level of complexity in their organization is low.

Same goes for bigger companies that provide simple services or products. Complexity within the organization is low, so processes might not be necessary.

Complexity is the best metric for deciding when to organize a process, because it can mean different things. If you're adding people to your team at a rapid pace, complexity goes up. If your team isn't growing, but your product is becoming bigger and you're adding a lot of new features, complexity goes up, too.

As soon as complexity starts growing, the need for standardized processes arises. It's better to start organizing processes too early, than to start organizing them because you're having incidents like production faults, security breaches, data leaks, or employees leaving due to frustration.

Whenever complexity goes up, processes can help you reduce it.

Final thoughts on introducing processes to management

In any work environment, processes are all around us, whether we standardize them or not. When you get around to organizing your work in processes, there are a few key rules to remember:

1) Common sense should always trump processes, and when the process is actually making your work harder—rethink it, or get rid of it.
2) Processes are always subject to change, you should keep improving processes not just as a manager, but as any member of an organization.
3) Everybody in your organization should know that they can suggest changes to processes—you need a culture of continuous improvement and feedback.

It's about guiding actions, and not dictating them.

How Scrum Can Go Wrong in Large Organizations

Over the course of his career, Łukasz Nowak went from being a stakeholder in waterfall-managed projects, through introducing Scrum in a large organization and managing a single Scrum team as well as 20+ separate Scrum teams, all the way to his current position as Service Delivery Manager at STX Next.

He's seen what it's like to introduce Scrum in a traditionally managed environment, plus what can go wrong both in a single team and a large group of teams.

In this chapter of our ebook, we'll share the most important things Łukasz has learned, hopefully providing you with the tools you need to avoid critical issues in your Scrum team(s).

You can also watch Łukasz discuss this topic in depth during one of our live sessions instead of reading the following section if you prefer:

Problem #1: Goal misalignment between different Scrum teams

In a big organization with multiple Scrum teams, you're always running the risk you'll lose money, time, and opportunities due to mismanagement. It's very easy to invest a lot of money in IT inefficiently.

Łukasz described the greatest issue he'd ever faced as misalignment of goals, or lack of a common direction.

When you have 10, 20, 30 teams, even if they're all working properly and using best Scrum practices, there's a huge risk that some of them will choose directions that are not aligned with the main goal and strategy of the company.

Why is that? One of the main roles in a Scrum team is the Product Owner. That person decides which actions should be taken, which features to build, and ultimately how their team will influence the rest of the company.

Left on their own, Product Owners from individual teams might make decisions focused too much on their own goals, and not on the goals of the company, which are usually related to growth, boosting the revenue stream and so on.

This is why the first thing we'll talk about is how to avoid having multiple Scrum teams go in different, separate directions.

The worst thing that can happen is multiple Scrum teams going in different directions

When Łukasz was involved in implementing Scrum in a waterfall-managed company, the available literature and knowledge was limited. The Agile manifesto was still relatively new and there were few Scrum courses. There weren't many frameworks showing how to coordinate and align more than 10 Scrum teams.

So for the issues his team had to solve, they had to rely on common sense, trying different approaches, and creating a strong feedback loop that would quickly tell them what wasn't working.

His main approach was to slice up the product into different parts. The company was doing ecommerce, so it was quite easy to do. Side note: the main KPI in ecommerce is the conversion rate.

Łukasz and his team sliced up the product based on different parts of the customer journey funnel. Several Scrum teams were responsible for the first stage of the funnel, others for the second, and so on.

Every layer had business goals. The first layer was responsible for visitors converting to add items to cart, another for sign-ups, etc. The great thing was that the final layer included a KPI regarding returning customers. It created a nice loop of the funnel, which seemed crucial to Łukasz. Every layer of Scrum teams had its own KPI to grow using their own resources, and making their own decisions.

These were just the Scrum teams responsible for visible, predictable things. Another group of teams were responsible for the platform, communication between services and modules, tools for developers, all the things that make the business work in the background.

All in all, the 20 teams covered all internal and external needs of the organization. How do you diagnose the problem of misalignment between so many Scrum teams?

Diagnosing and solving the problem of Scrum team misalignment

Once Łukasz's team started seeing different features that the teams were developing, they realized that some of them contradicted others. Team A was building one thing, and hoping to achieve a goal that was quite opposite to what Team B was trying to achieve.

The main, big solution to this is to define, create, and promote the strategy of the company within the organization. Every day, in every company, thousands of decisions are made, from tiny to huge. If your team lacks a common understanding of what the main company goal is, those decisions can contradict each other, leading to lower impact on both sides.

A clearly defined strategy solves this. It tells everyone why we're here, what's the role of every department and person in the grand scheme of things, why and what should be delivered to the customer or stakeholders—or "internal clients" as it's sometimes called in the enterprise world.

Once everyone knew what should be done and why, the second step they took was something that might sound weird. In a big organization, Product Owners shouldn't have full autonomy to decide what to do. Let's not beat around the bush—we needed them to obey, to a certain extent, the decisions that were made higher up.

But it's not as simple as imposing your will on people, you don't want to be a total authoritarian. What Łukasz came up with was to match the teams in groups, and introduce a new person that would be a bridge between them. Someone responsible for a group of teams, who would coordinate and control how they spent money and time in every sprint.

He called them Product Development Managers, and each one was responsible for one layer of the customer journey funnel, along with the teams that were inside that layer. It seemed reasonable at the time, and it worked really well for them.

But it's not the only solution. There's one that's more general, one that drives efficiency in individual Scrum teams, and between separate teams: the feedback loop.

One solution for many problems: feedback loop

You might introduce a strategy and new roles within your company, but without a feedback mechanism, you won't know if it's working.

Łukasz set up his team's main goals by defining their place and share in the market. They had to measure themselves against the competition, see what the advantages and disadvantages were. But it's not enough to do it once, it needs to be done regularly.

In their case, there was a big strategy update once a year, but now Łukasz would advise to do revisions at least every 6 months. It will tell you if you're still on the right track, and show you if your decisions are still valid and beneficial for the company. Valid in terms of how the market and competition are changing.

The impact of COVID-19 on the ecommerce industry is a good example. The market changed rapidly and unexpectedly, and companies threatened by this situation had to adapt or die. Especially offline companies, which had to quickly undergo digital transformation in order to survive.

How can you learn and adapt to market circumstances quickly?

To be able to learn, you have to secure the possibility of making failures

Your company culture shouldn't support failures, but it should tolerate them. The feedback loop makes this possible.

Every mistake should bring us to lessons and new ways, new approaches to eliminating those mistakes. It's not about celebrating mistakes, but talking about them openly. Analyzing why they happened, what should be done differently.

You need to give mistakes the right amount of attention and make sure they're easy to discover. It's really hard for managers to provide information about mistakes without placing blame, without pointing fingers at the people or teams who made the mistake.

Being honest, fair, and transparent is crucial. Post-mortem documents can be created, and spread across the organization if there was a failure, so everybody knows what happened and how to avoid it in the future.

Let's move on to a few actionable tips for maintaining team alignment in large Scrum organizations.

Maintaining team alignment

A radical change causes a cascade of decisions. If you notice that your strategy has to drastically change, like a completely new direction, you need to create a new strategy, set new numbers and goals to have a benchmark, decide on new KPIs, and all that has to cascade down to different organizational layers.

It's a great deal of work in every layer to lay out the tactics for the next time period. Good questions to ask in each layer are:

1) How should we change the KPI?
2) What can we do to reflect the new strategy and direction in our backlogs?

Meaning the group backlog for the whole layer, but also every Scrum team's individual backlog. There's no place for features and ideas that don't support the new common goal.

Plus, it can provide a sense of relief to the team; cleaning out the backlog is a bit of a zen exercise.

Every now and then, there's a magical moment when you check if your decisions were correct—as long as you have a feedback loop for the strategy, and a loop for each group of Scrum teams, and individual loops in each team.

This is how you make sure that, from the bottom up, everybody is making progress towards their own KPIs, and the company's main KPI.

Loops within loops within loops. That's the beauty of Scrum, you can make a mistake as long as it's checked soon, and if you're gonna fail—do it quickly. This way you get a second chance to get it right, then a third chance, and so on.

Problem #2: Humanity at work—motivation and conflicts in large Scrum teams

The second big problem of large Scrum organizations is very basic and human. It's the problem of losing meaning in your work.

You don't want people to spend their time wondering, "Is there a real purpose for me being on this Scrum team?" or, "Is there any real purpose for my team?" An organized explanation of what your team does really helps here.

This problem boils down to motivation. People want to be seen as, well, people. As human beings who, individually and as a team, are doing something that's tangible, significant. If software were a house, they want to know which bricks they're adding to it.

For Łukasz, personally, it's important to be a part of something that's useful and fruitful. He certainly isn't alone in this, since meaning is a basic need in life and work. What can you do to ensure that people in your organization stay motivated?

Avoiding demotivation in a large organization

Łukasz thinks a good solution is to talk with people about their roles, responsibilities, and areas of influence. They might not realize how many other people rely on their work.

It also definitely helps to celebrate great decisions, and visualize the results of good work. At the same time, it's not useful to hide mistakes, or when the work they did hasn't resulted in anything meaningful.

All of this can only happen in a culture of transparency.

Celebrating victories, understanding failures, it's all crucial here. All this, coupled with the feedback loops we mentioned, and opportunities to fail quickly and try again, is premium oil for the engines of your Scrum teams.

It's not as simple as patting people on the back or punishing them—this isn't pre-school.

The key here is to translate KPI progress from percentages into something tangible. In ecommerce, even slight KPI progress can lead to thousands more products being sold every day.

For example, instead of telling your team(s), "We grew a key metric by X%," it would be better to say, "Thanks to your work, we sold Y more TV sets/wearables/pairs of shoes." You need to find a way to show how a specific change in the code, adjustment or new feature, leads to more satisfied customers that will come back.

But that's not all. There's one thing that's always going to happen in large teams, and that's conflict. Let's see how to handle it.

Dealing with conflict

Having your teams divided into layers, and having a clearly defined strategy are already good ways of avoiding conflict. Everybody knows what they're supposed to be doing, so they shouldn't invade other areas.

When there are discussions, transparency helps resolve them quickly. The ability to be open and change your mind when someone else is right is important, although like most soft skills, it's hard to teach someone how to do this.

Leadership is a big part of solving conflicts. Companies are democracies to a point. At the end there has to be a person, or a group, to make the final call. It's a matter of culture. Responsibility has to be clear, and it shouldn't be hidden somewhere between groups or people.

This is especially important if you have to drop a project or a feature. Someone has to have the power to drop something when they notice that it goes against company goals. Which means that months of work could go into the bin, but things like this unfortunately need to happen sometimes.

When they do, transparency comes in handy again. In such cases, Łukasz advocates for just saying how you feel—it's really sad, months of people's work have to be discarded, the market has changed, that work doesn't make sense anymore, it happens, etc.

It's likely the team will feel just a little demotivated after that. So how do you get them excited again? Show them what's the next big thing, the next feature, the next project to focus on, and how this new thing will influence us as a company, and get us to our goals.

Being sad, being sorry and showing compassion are important, but outlining the next steps and getting people fired up about them is even more important.

Problem #3: Technological decisions in large Scrum teams

Having a lot of teams, they can make independent technological decisions. It's good to have a framework that clearly defines limitations for these decisions.

You can choose the tech, but you have to be aware of what consequences it brings, and the variety of other technologies that can also be applied.

Missing this part can lead to chaos in your organization and make development needlessly expensive. The solution here is easy to understand, but hard to apply. You need to work with the best people on the market.

Choosing technology is comparable to deciding how you will design engines in an automotive company. Sometimes the direction you take influences not months, but years of work, and you have to be prepared.

That's why the architects and CTOs on your team need to be the brightest minds in the industry.

Now, should every little decision, every library be consulted with the CTO? Not exactly. Łukasz found that a good solution is to create a guild of people who have the most experience and let them challenge each other, discuss, and decide what direction should be chosen.

Not every decision has to go through the guild, mainly those that have major long-term consequences across the organization.

It doesn't have to be a guild, you can find your own mechanism of proposing different solutions, and having them challenged by the rest of the organization. You don't want people imposing their choices, it should be a discussion, maybe a competition of solutions, people playing devil's advocate and looking for possible issues.

To make the right decisions, you need experts with vision and knowledge, but also a place, time, and opportunity to challenge that vision and discuss other options.

There's a great saying that goes back to Confucius: "No friction, no shine"—to paraphrase. As in, you can't polish a gem without friction, same as you can't make good decisions without a bit of friction in your team.

Final thoughts on using Scrum in large organizations

To sum it up, the main problems that arise in large Scrum organizations are:

- Goal misalignment between different teams
- Demotivation and conflicts
- Tough technological choices

If you're dealing with these issues, the first thing we suggest is talking to your people, checking if their minds and hearts are still there. Verify if the goals of your company are still valid, and if everybody understands them.

If you have 20 or more Scrum teams, we're sure you've already checked all the frameworks for managing so many people. We'd use those frameworks as support, but still be sensitive to what makes your organization special, and use common sense when applying them. Keep in mind what's valuable in your culture and try to boost it.

In the end, it's about finding where the important things meet. The business goal, money, time, the human approach, people's basic needs. When you merge them, you can find the right channel to grow these things in parallel, and then you'll know that you're headed in the right direction.

How Implementing KPIs Will Help You Become a Data-Driven Business

Every company wants to run more efficiently and help their employees reach their full potential. Among others, there's one great way to make that happen: KPIs.

They're the name of the game when it comes to building a modern business, with nothing to lose and everything to gain by implementing them. We've been using KPIs for a while now at STX Next and they've really been working out for us.

We're here to help you find out what you can gain from relying on KPIs. **Keep reading to:**

- **learn all about how KPIs can change your company for the better,**
- **obtain the tools necessary to get everyone on board with the change,**
- **get a good idea of the benefits and drawbacks of KPIs.**

The following section is based on a live session with Łukasz Koczwara, VP of Engineering at STX Next. If you'd rather hear him talk about KPIs instead of reading the piece below, feel free to watch this video instead:

What are KPIs and what is their purpose?

The abbreviation “KPI” stands for “Key Performance Indicator.” In a roundabout way, it describes exactly what KPIs are—a means of monitoring performance in key areas, with a specific end goal in mind.

The main idea behind KPIs is to establish a specific outcome and measure the progress along the way, all in order to learn from the collected data and use it for future planning.

KPIs have seen a surge in popularity lately, making them a veritable revolution in business. They’re the ideal solution if you want to improve your company’s overall performance, but also inspire your employees to be more driven.

How to get the CEO on board? The pros and cons of KPIs

While KPIs may seem like a dream solution from a certain point of view, CEOs might not be so easily convinced—and their concerns are valid.

Even KPI enthusiasts such as ourselves can’t deny that there are elements to KPIs that tend to raise red flags. Data security is one such area, as it’s hard to monitor financial data without losing its confidentiality. Since transparency is one of the core tenets of KPIs, this could become an issue for some top executives.

With that said, there is one crucial aspect of KPIs that should more than make up for that particular fear CEOs have to overcome: concrete data. While disclosing metrics is a delicate subject, you simply won't get the full picture of how your company is doing without this disclosure. After all, how can you really expect to know whether this month or quarter is better than the last when you can't reliably compare notes?

In that regard, a lot of CEOs rely on their gut feeling. While that certainly is one way of doing things, it's hardly an objective metric. KPIs help you operate on factual data, not only establishing a more consistent way toward better profits, but also allowing your team to work together more organically.

As a result, it never falls to just one person to be responsible for a project, a department, or even the whole business. The effort becomes collective.

How to communicate implementing KPIs to the rest of the company?

CEOs are one thing, but your other employees may be wary of KPIs, too. Revolution has never been an easy path, and people really don't like to deal with change if they can help it. Plus, KPIs make things much more transparent and many people fear they might be misused as a tool to point fingers, pin blame, or even go as far as to incite witch hunts.

This fear can get so strong that some people may never come around to trusting KPIs; however, most are likely to become much more open to them once they realize the tradeoffs that come with the change. The same transparency they fear might serve as a gateway to bullying can actually help them get the recognition they deserve for their efforts.

Not only does it make employees feel safer and perform better when they see that their work is being appreciated, they'll also have a much higher opinion of their workplace. When used correctly, KPIs serve to create a more fair environment that treats everyone equally based on the quality of their work, not to rule over them with an iron fist. That way, you're not just increasing efficiency—you're building morale.

And while we're at it, no one really likes to have their boss breathing down their neck. KPIs are designed to focus on finding solutions rather than seeking out unruly minions to punish. Having them implemented makes everyone feel like their contribution matters.

How to choose the right KPIs for your company?

One of the greatest and most common mistakes companies make that turns them off the very idea of using KPIs is that they're attempting to use a template.

Since KPIs are all the rage these days, it’s understandable you might want to take a look at the sort of KPIs other companies in industries similar to yours implement. And this is where that critical error happens—those KPIs aren’t yours.

KPIs have to be tailor-made and business-specific. In order to make sure your KPIs work for your company, there are several aspects you should consider:

- What type of business do you run? Are you a service or product company? (This is the best question to start with, as this aspect determines your overall goals.)
- What is your service/product? (Consider how this can impact the motivations and responsibilities of your employees.)
- How do you charge for your service/product?
- What is the scope of your service/product?
- How do you operate as a business?

The reason for this particular order is that starting from the base and moving upward is a surefire way to figure out what it is you really need. This is how you arrive at the particular set of KPIs that are just right for you.

How to prioritize and implement the most important KPIs?

Once you set your goals and know where to go from there, it’s very easy to get carried away. Sometimes, people start working with KPIs and get tangible results right away—that’s great, but not a hard-and-fast rule. Taking a step back is the smarter approach.

While it may be tempting to expand your initial KPIs from 5–10 to 20 or even 40, the best course of action is to stick to your guns, at least for a while. The more KPIs you add, the more overwhelming the amount of information and less clear the course of action.

When you’re working with a smaller pool of KPIs, finding your focus becomes crucial. Good prioritization can come a long way in making your business efficient. It’s best to start by trying out a few KPIs and seeing what works. Your actions have consequences—finding out which ones produce results and eliminating those that don’t is key.

The next step you’ll want to take is evaluating the stage you’re currently at. For instance, if you’re an up-and-coming company, devoting more of the KPIs to hiring will probably make the most sense for you.

You'll quickly discover which types of KPIs are not relevant to your current situation—filter them out mercilessly. This will help you find out which metrics are actually moving your company forward, allowing you to set up your KPIs around them.

How to implement KPIs effectively?

As we mentioned before, there's no single KPI template that any business can use if they wish to really grow from it.

What you do as a company is absolutely central to the entire idea, so examining the core of your activities should be your starting point in deciding where you take your KPIs. A good way to help you stay focused on what matters the most is to keep track of what works and what doesn't using Microsoft Excel or Google Sheets.

You can also spare yourself some headaches and hire a KPI specialist to handle everything for you. It takes practice to really figure out what kind of KPIs will benefit your business, but if you're working with someone who already knows the field like the back of their hand, it'll make your life a whole lot easier.

Another relevant point to bring up here is that if you really want to master KPIs, you need to have a single, unified source of truth. Software such as Power BI presents data from everyone involved in the project using a dashboard, while also providing reports on the condition of the company.

Most importantly, however, keep in mind that KPIs are not set in stone. You don't just decide on one set of them and run with it indefinitely. It's only with a continued effort to improve, innovate, and monitor your KPIs that they can truly play their part.

What are the benefits and risks of implementing KPIs?

While we're confident KPIs are the revolution modern business needs, they might not be for everyone. By choosing KPIs, you can count on the following benefits:

- Properly applied KPIs and targets help your company embrace autonomy and self-organization, fostering the intrinsic motivation of your employees.
- Everyone's actions become transparent, so it's evident how their collective efforts make a difference, helping them grow both individually and as a team.
- With less micromanagement and more freedom, as well as a goal that unifies the entire company, there is far less potential for conflict and more ground for trust.

- You can't improve what you don't measure. KPIs show you where you stand today, so that you can move in a new direction tomorrow. No more wading in the dark.
- You don't have to give up on trusting your intuition—this is the "gut feeling" revolution. Intuition is a useful tool that comes with invaluable experience, but it's so much better when there's concrete data to back it up.

That being said, nothing worthwhile comes without risk. Here are a couple of risks you should keep in mind when it comes to KPIs:

- There's a small chance you'll realize somewhere along the way that your KPIs are all wrong and don't match your business. If you're going in the wrong direction, it doesn't matter how fast you go.
- It's easy to get carried away with KPIs without considering your limitations. Sometimes, you have to be brave enough to throw away 80% of what you're trying to measure to really make the remaining 20% count.
- Be prepared for some growing pains. A company that isn't used to KPIs can have some trouble adjusting to the new system, as it is a pretty big upheaval of standard workplace politics.

Final thoughts on implementing KPIs at your company

Companies are akin to living, breathing organisms; sometimes, all it takes to vastly improve their health is one key lifestyle change.

Like a good diet, KPIs come with a wide range of benefits: a better company atmosphere, increased safety, and more satisfying results.

While KPIs may be the revolutionary future for business, it's important to stay grounded when implementing them at your company. The results might not be immediately obvious, but with time and perseverance, KPIs can make a great difference.

With the help of KPIs, you have the potential to reach the stars—you just have to take your time preparing for liftoff.

Best Tactics in Successful Business Negotiation for Technical Managers

As a technical manager, you'll often find yourself in the middle of conversations requiring great negotiation skills. It may be asking for a raise, convincing highly sought-after talents to join your team, or asking your kids for a slight breather to snooze on your favorite TV show.

How do you handle such conversations, figure out what works best for everyone, and ensure that you don't get the shorter end of the stick?

Read on to discover negotiation strategies that will help you navigate difficult conversations as a technical manager and make negotiation at work or in your personal life a walk in the park.

Or if you prefer, you can watch Matthew Harris, our VP of Sales, discuss concrete negotiation techniques for technical managers during this two-part live session:

Why should you brush up on your negotiation skills as a technical manager?

Being a technical manager is not a piece of cake. Technical managers are responsible for building competent teams, meeting stakeholders' technical expectations, and executing complex software projects. Negotiation skills are important for these responsibilities.

It takes a good negotiator to convince talented developers to pick your organization over other attractive companies. Managing a team of software developers will require several processes, conversations, and of course, negotiation. You also need to negotiate with third-party contractors to get the best deal for your organization.

With great negotiation techniques, you can demand better pay or additional benefits without any hassle. You can also get shorter workweeks and other perks that will make your current job your dream job.

Negotiation goes beyond the workplace. You'll need to barter with your spouse or kids at home, too. For instance, you may need to convince your kids to settle for a healthy sandwich for dinner instead of pizza or plead with your spouse or partner to reschedule dates at the last minute.

If you do a quick review of your daily activities, you'll find several examples of negotiation skills at work. It's safe to say that you need great negotiation techniques to excel in your career, build solid relationships with the people around you, and live a balanced life.

4 useful techniques for successful negotiations in business and life

Negotiations may not be your strong suit, but with the right techniques and tricks, you can confidently hold down the fort at the bargaining table.

We'll focus on these 4 negotiation strategies:

- mirroring,
- labeling,
- tone of voice and body language,
- asking the right questions.

Negotiation technique #1: Mirroring

In negotiation, mirroring is essentially repeating a few words spoken by the other party in an inquisitive tone. It's a subtle technique that encourages your interlocutor to reveal further information and gives you a clearer perception of their concerns or requests.

It's a great way to highlight and clarify gray areas in a conversation. You can repeat the key terms that need to be clarified and elicit the right information from your counterpart with mirroring.

Also, mirroring keeps the conversation going and helps you reach resolutions quicker. It compels you to listen to your interlocutor and process the information gotten from them on the spot. It's a great strategy for building good rapport and strengthening interpersonal relationships. You'll also appear warm and empathetic, which is great for successful negotiations.

Mirroring may be verbal or non-verbal and both can be used simultaneously. Verbal mirroring is the repetition of words spoken by a person while non-verbal mirroring involves mirroring your interlocutor's body language.

Why mirroring is so important

Mirroring is a good negotiation strategy for anyone who hates negotiating. It's so important because it gets the other party talking and revealing vital information with little effort from you. You only need to say a few words to prompt the speaker to spill the beans.

Bargaining with reserved people can be quite taxing. Conversations end up being one-sided with little or no input from your counterpart. Mirroring is a great technique when bartering with shy people or those who would rather avoid the conversation.

It's equally helpful if you're the withdrawn person at the bargaining table, but would like to be heard and contribute your piece. You'll be able to get all the details you need to negotiate without speaking a lot.

Tips for getting mirroring right

- **Repeat only a few words**

You should only repeat key terms and keep them between 3–5 words.

- **Start with the last words**

It's easier to recall the last words spoken. It's also easier for your interlocutor to carry on with the conversation when you mirror their last words.

- **Be genuinely curious**

Using an inquisitive tone makes your opponent willing to share more.

- **Just shut up**

It may be tempting to chat away, but the key to effective mirroring is to listen more than you speak.

Let's take a look at a quick mirroring example in a conversation between a technical manager and a developer struggling to meet a deadline.

> **Developer:** *I'd appreciate it if you moved the deadline for this project. I'm still working on some additional features, and I will need to make some changes.*
>
> **Technical Manager:** *Make some changes?*
>
> **Developer:** *Yes, I noticed some mistakes in the initial code. I need to audit the initial code and run another smoke test. I need some more time to tidy things up but I'm sure I can work around things soon.*

Negotiation technique #2: Labeling

We use labels daily to track, identify and save vital information in our memories, applications, and pantries. Labeling also works perfectly for negotiation. Labeling is the art of identifying the emotions expressed by your interlocutor.

You can tell a lot about a person's feelings from the words they speak. Oftentimes, we can identify emotions such as frustration, anger, resentment, dissatisfaction, and disapproval by listening to the people with these emotions speak.

With labeling, you can voice out your interlocutor's emotions to elicit further information from them. It's simply identifying and expressing your interlocutor's emotions in a manner that gets them to think about their feelings.

Labeling is a subtle negotiation strategy that works well in the workplace. It diffuses tense or negative emotions and gets the other party to address those emotions. It also reinforces positive emotions.

Ultimately, labeling helps you establish a stronger rapport with your interlocutor.

Tips for effective labeling

- **Identify the emotion**

Watch out for verbal and non-verbal cues that portray your interlocutor's emotions.

- **Label the information**

Put a label on the feelings and emotions you've observed. Simple labels like "happy," "sad," or "disappointed" will do the trick.

- **Use preemptive phrases**

Preemptive phrases are the keys for unlocking your interlocutor’s Pandora’s box. Some preemptive phrases to take note of would be: “It feels like…” “It sounds like…” “It looks like…”

- **Just shut up**

Silence is also golden when labeling emotions.

Negotiation technique #3: Tone of voice and body language

Words are golden, but the tone of voice and body language give context to the words we speak. Professor Albert Mehrabian conducted a study that compared the impact of spoken words, tone of voice, and body language in communication. The study focused on a group’s assessment of the verbal and non-verbal components of a conversation. The result is what is now popularly referred to as the 7-38-55 rule.

According to this rule, people pay 55% of their attention to your body language, 38% to your tone of voice, and only 7% to your words. Adapt your negotiation strategy accordingly!

Content is important but non-verbal communication is key to effective communication and negotiation. Thus, it’s important to pay attention to your interlocutor’s body language and tone of voice during a negotiation. This will help you interpret and label their emotions appropriately.

Also, using the right tone of voice and body movements that compliment your words makes you a better negotiator. When your non-verbal communication doesn’t match your words, your interlocutor gets a different message entirely. They’ll most likely believe the non-verbal message over your words.

Here’s a practical example during an interview session:

> **Interviewer:** *“I’m very pleased to meet you today.”* (frowns and scrunches their nose) *“I’ve reviewed your resume and I think you’re an ideal candidate for this job.”* (looks away from the candidate and exchanges glances with the co-interview) *“We’d like to know what it’ll take to get you on board.”* (smiles briefly)

Can you think of a perfect reaction to the above scenario?

Of course, the diplomatic thing to do would be to react positively and show genuine interest and enthusiasm for the role using the right body language. But it would be difficult to believe the interviewer's opinion and assessment of the candidate by their words, as their actions portray a different meaning.

What kind of tone/body language should you use during a negotiation?

There's no hard-and-fast rule, but it always helps to pay attention to the following:

- **Eye contact**

It's great to maintain eye contact with your interlocutor. It helps build a connection and establish rapport. It also helps you convey the message firmly and confidently.

- **Tone of voice**

You should tweak the tone of your voice to reflect the right message during negotiation. You can use a high tone to express genuine interest in your interlocutor's opinions and a low tone to show empathy.

- **Facial expressions**

Always ensure that your facial expressions match your words. Smile, laugh, nod, or frown as needed. A blank facial expression makes you appear cold and blunt.

- **Postures and gestures**

Postures and gestures make conversations more personal and empathetic. They help your interlocutor evaluate your position beyond the words you speak. Keeping your arms crossed, for instance, can make you appear uncompromising, while gesturing makes you come across as passionate.

What to watch out for when examining the tone/body language of your interlocutor

Here are some non-verbal cues to watch out for when examining your interlocutor's tone or body language:

- **Inconsistencies**

Non-verbal cues are a great way to reinforce verbal statements. However, when there's a disparity between what is being said and the speaker's tone or body language, it clearly indicates that your interlocutor is not being honest about their feelings and emotions.

- **Subconscious movements**

You should watch out for subconscious movements from your interlocutor. For instance, cracking knuckles or rubbing one's ears are subconscious movements people make when they are nervous.

- **Read the room**

If you're negotiating or discussing with a group, you should pay attention to the different forms of non-verbal communication going on in the group. You may need to take some minutes to observe and interpret the different body movements before you come up with a theory.

Tone of voice and body language tips for video conferencing

Most negotiations are concluded virtually today. Just as you've adopted tools that make the remote world a lot easier to navigate, you also need to arm yourself with tips and tricks for negotiating virtually.

Using the right body language and tone of voice will make your video conferences a lot more engaging and personal. You'll also come across as professional and make a great impression on other attendees.

Here are few tips and tricks that will help:

- **Don't keep your face too close to your screen**

On most video conferencing applications, you'll find out that your screen is your counterpart's space. Therefore, keeping your face close to your screen may appear intrusive. Make sure your face takes up not more than a third of your screen.

- **Use the right tone**

Your tone is more important during video conferencing as you can barely see the other party's body. Your counterpart will interpret your words based on the tone you use. Always use the right tone to convey your message.

- **Pay attention to lighting**

You don't have to invest in expensive studio lighting. A makeshift lighting such as a second screen will work perfectly. You can use plain background colors on your second screen to further accentuate the lighting in your room. For instance, a plain white background will make the room brighter while a plain yellow background will add a warm glow.

- **Use discreet equipment**

Avoid large headphones with an attached microphone. They can be distracting. If you're using a wired mic, hide the wire beneath your shirt to minimize any distractions.

- **Maintain eye contact**

It's important to maintain eye contact during video conferences. The only way to do this on a video call is to look at the lens or webcam. Looking at any other part of your screen or camera will make you appear unfocused.

Negotiation technique #4: Asking the right questions

Another great negotiation tip is effective questioning. Negotiations present an opportunity to extract information from your interlocutor. You can do this by asking the right questions.

Asking questions during negotiations steers the conversation in your desired direction. It keeps you in control of the narrative and makes you appear warm and empathetic. It also gives the interlocutor an opportunity to reflect and share their thoughts in better detail.

While asking questions can be all you need to bargain better, it must be done cautiously. Not all questions are conversation boosters. Certain questions may be viewed as offensive by your interlocutor. It's also possible to get them worked up and defensive with the wrong questions.

Therefore, it's best to know the questions to avoid asking and the type of questions that are great during a negotiation.

Questions to avoid in a negotiation

a) Leading questions

One effective way to extend and enrich your conversation with other people is by minimizing persuasion or intrusion of your views on the other person. One way to get this wrong is by asking leading questions.

Leading questions are subjective and compel the other speaker to accept your perspective. They force the interlocutor to accept your views and opinion in a subtle way.

Here's an example of a leading question during salary negotiations:

> *"With all the perks listed in our offer, don't you think this is a great package for you?"*

With leading questions, your interlocutor will be caught in a difficult position. They'll have to consider the options you're suggesting to them or defend their opinions against your suggestions.

Leading questions do not elicit further information from your interlocutor. They strain the conversation instead and may encourage the other party to provide false or presumptuous information.

b) Asking "why"

You should never ask "why" during a negotiation. I'm sure you're probably thinking "why" after reading the previous sentence. "Why" is a natural response in most conversations, but it's often used loosely.

On the surface level, it seems like a great way to request clarity and further information from your interlocutor. But at what cost?

Asking why puts people in a position where they have to defend their opinions. Many people view the question as confrontational. This can strain the conversation with the other party and make the negotiation process competitive rather than collaborative.

Therefore, if your goal is to sustain a conversation until you've successfully negotiated the terms, ditch the "why" questions.

Need further convincing? Here's a comparison of two questions during a negotiation with a vendor.

> **Question 1:** *Why can't you accept our 7-day payment terms?*
>
> **Question 2:** *What is it about our 7-day payment terms that is important to your company?*

Both questions seek to understand the vendor's opposition to the 7-day payment terms, but the first question comes across as accusatory. It forces the vendor to defend their position and doesn't guide the vendor to reveal helpful information.

The second question is more collaborative. It makes the interlocutor share more information that could be useful for the negotiation. It'll also help the parties to reach a mutual decision faster.

c) Yes-and-no questions

Questions requiring yes-or-no answers should be used sparingly in a negotiation. Such questions put your interlocutor in a difficult position. They are forced to pick from two restrictive options which may not fully capture how they feel about the subject matter.

Moreover, yes-or-no questions do not help in eliciting information from your interlocutor. They actually cut off further information. Your interlocutor focuses on making a decision at that moment that may not be best for them.

Yes-or-no questions make conversations one-sided and less spontaneous. This can rub off negatively on your interlocutor and generally make the conversation less productive.

Here's an example of a yes-or-no question while negotiating a pay raise with an employee:

"Will you accept our final offer of a 5% annual pay raise?"

Now, there are only two likely responses to this question: yes or no. If the counterpart says yes, all is well and fair. If they say no, you're left with two options. You may proceed to ask "why," which is another question that you'd best avoid or end the negotiation at a deadlock.

A better way to phrase this question would be to make it more exploratory and less dismissive. Here's a good alternative:

"What do you think about a 5% annual pay raise?"

Questions to ask during a negotiation

a) Open-ended questions

Open-ended questions are great conversation starters and boosters. They are useful during negotiation as they help you draw information from your interlocutor. Simply put, open-ended questions are questions that require detailed answers and not yes-or-no answers. They get the speaker to narrate their feelings and help you understand things from your interlocutor's perspective clearly.

Open-ended questions help you to build rapport quickly with your interlocutor who may be unwilling to share more details. They also position you as a great listener.

Here's a good example that can be used during employee appraisals:

"Tell me about your relationship with your team members."

b) Calibrated questions

Calibrated questions are another great way to ensure that your interlocutor flows with the conversation. Calibrated questions are also known as "how" or "what" questions. They help your interlocutor focus on the key terms and explain their position better.

Here are good examples of calibrated questions during salary negotiations:

"How did you arrive at this particular rate?"

"What other benefits can we offer to make you whole?"

Calibrated questions position you as an active listener and problem solver. It gives your interlocutor the impression that you are willing to collaborate and find solutions that meet their needs. It also helps you express empathy and concern for your interlocutor.

Final thoughts on business negotiation for technical managers

Negotiation skills give technical managers an advantage in their professional and personal lives. They allow you to take on difficult conversations and make decisions with great negotiation skills. You can also achieve personal goals and lead your team to achieve set goals and objectives.

Luckily, negotiation strategies can be learned. If you consistently practice the techniques shared in this chapter, you're bound to become a better negotiator.

Final Thoughts

Thank you for reading our Tech Leaders Hub ebook on management and growth. We hope it'll help you make better business decisions and strengthen your own position as a tech leader yourself.

For more Tech Leaders Hub content, be sure to subscribe to our newsletter or follow us on LinkedIn so that you get informed about upcoming sessions and never miss a chance to connect with industry experts on subjects that matter to you.

Please keep in mind that Tech Leaders Hub is just one content platform STX Next provides. We have an active blog we update regularly and plenty of useful resources for you to benefit from.

To single out one particularly special asset, The Global CTO Survey 2021 Report offers 70+ pages of phenomenal insights for tech managers and executives collected from 500+ CTOs worldwide.

Here are a few other most notable examples:

- The Ultimate On-Site and Remote Guide to Hiring Software Developers
- The True Cost of Hiring In-House Developers
- A Practical Guide to Outsourcing Software Development
- The C-Level Guide to Software Development Nearshoring

We also provide a variety of services to meet any software development needs you may have—from Python and React Native to DevOps and Product Design. Whether you're looking for team extension, end-to-end development, or technical consulting, our experienced specialists are ready to support your project right away.

Simply reach out to us and we'll gladly see what we can do for you.

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

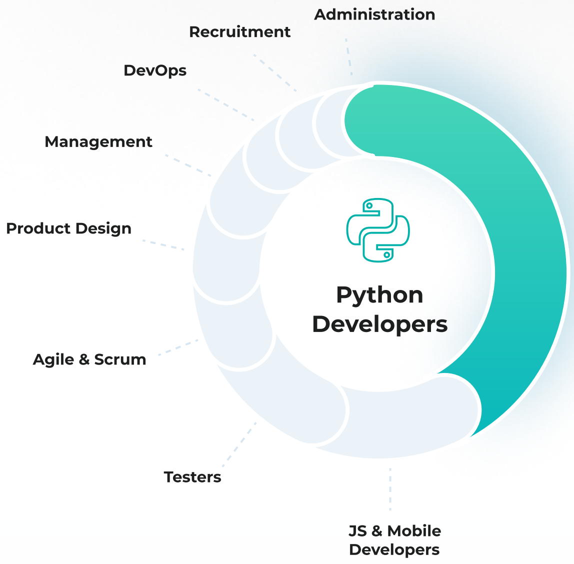
Additional developers or experts supporting your development efforts within 14 days

End-to-End Development

Full development team taking your project all the way from discovery to deployment

Consulting & Expertise

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

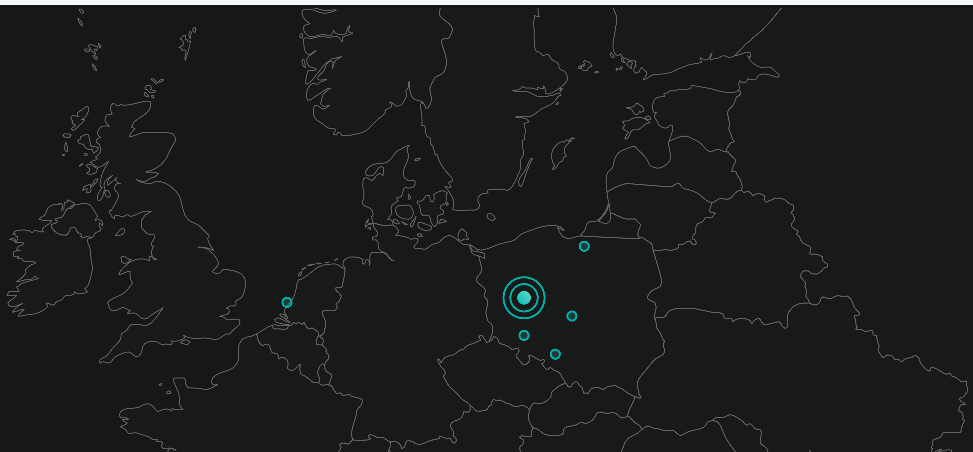
6.5+ years
average experience of our developers

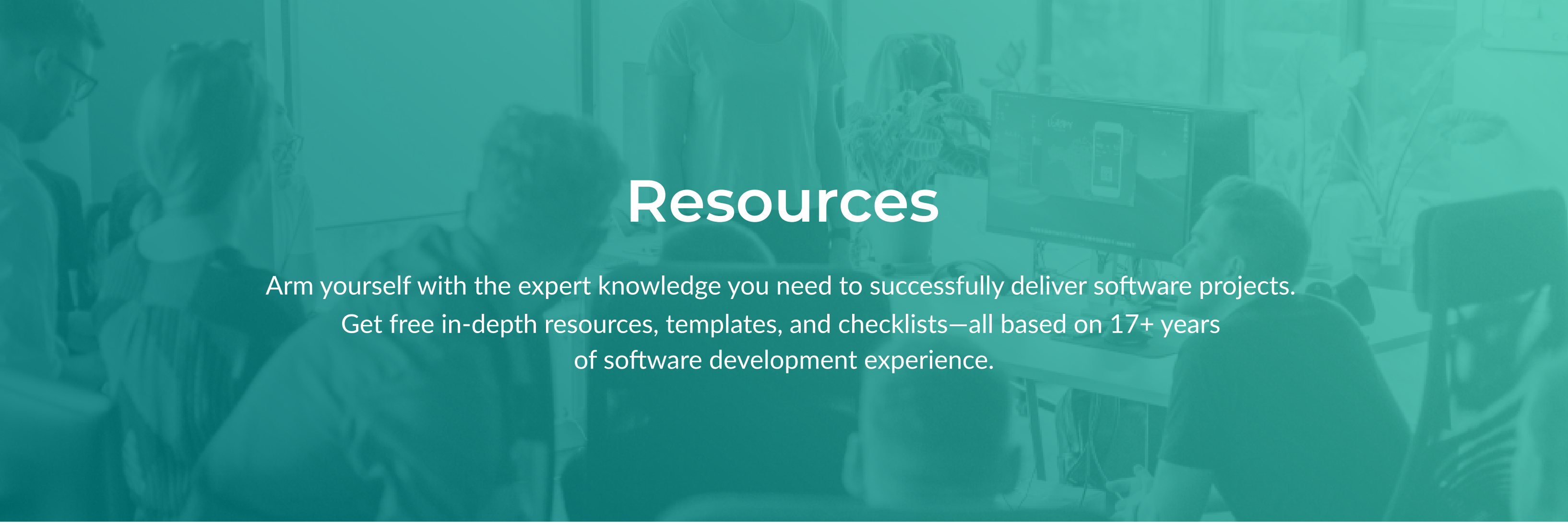
Locations

Poznań (HQ)

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

Wrocław
Olsztyn
Katowice
Łódź
Hague (Netherlands)





Resources

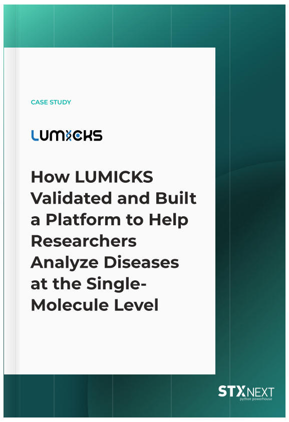
Arm yourself with the expert knowledge you need to successfully deliver software projects. Get free in-depth resources, templates, and checklists—all based on 17+ years of software development experience.

Reports and case studies

TheGlobalCTOSurvey × STX NEXT

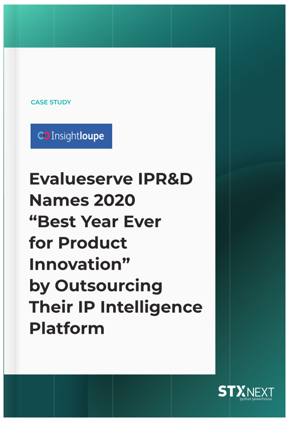
SURVEY RESULTS & INSIGHTS

The Global CTO Survey 2021 Report


CASE STUDY

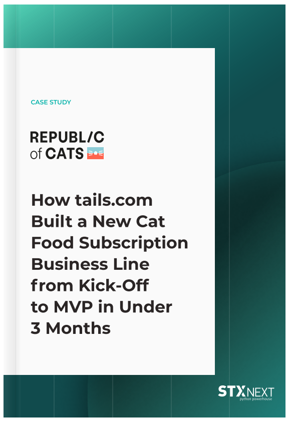
LUMICKS

How LUMICKS Validated and Built a Platform to Help Researchers Analyze Diseases at the Single-Molecule Level


CASE STUDY

Insightloupe

Evalueserve IPR&D Names 2020 “Best Year Ever for Product Innovation” by Outsourcing Their IP Intelligence Platform


CASE STUDY

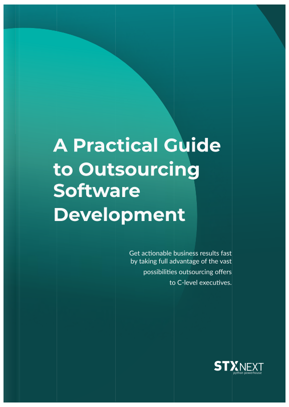
REPUBL/C of CATS

How tails.com Built a New Cat Food Subscription Business Line from Kick-Off to MVP in Under 3 Months

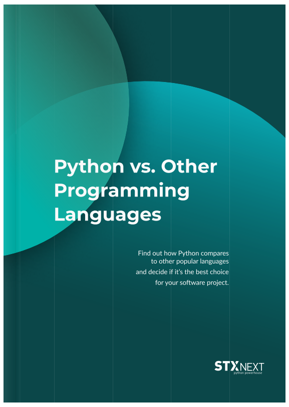
Guides

What Is Python Used for?

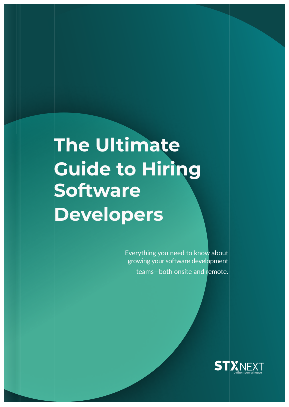
Discover why Python is such a great fit for so many industries and what areas of technology or business it benefits the most.


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.

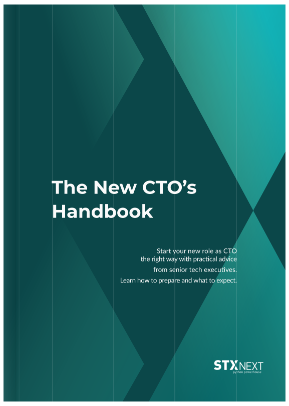

Python vs. Other Programming Languages

Find out how Python compares to other popular languages and decide if it’s the best choice for your software project.

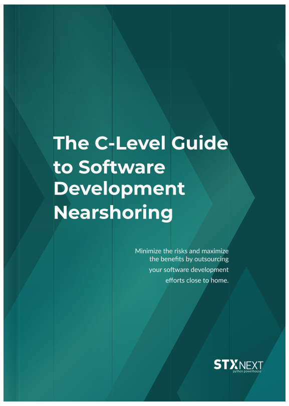

The Ultimate Guide to Hiring Software Developers

Everything you need to know about growing your software development teams—both onsite and remote.

Ebooks


The New CTO’s Handbook

Start your new role as CTO the right way with practical advice from senior tech executives. Learn how to prepare and what to expect.


The C-Level Guide to Software Development Nearshoring

Minimize the risks and maximize the benefits by outsourcing your software development efforts close to home.

TECH LEADERS HUB by STX Next

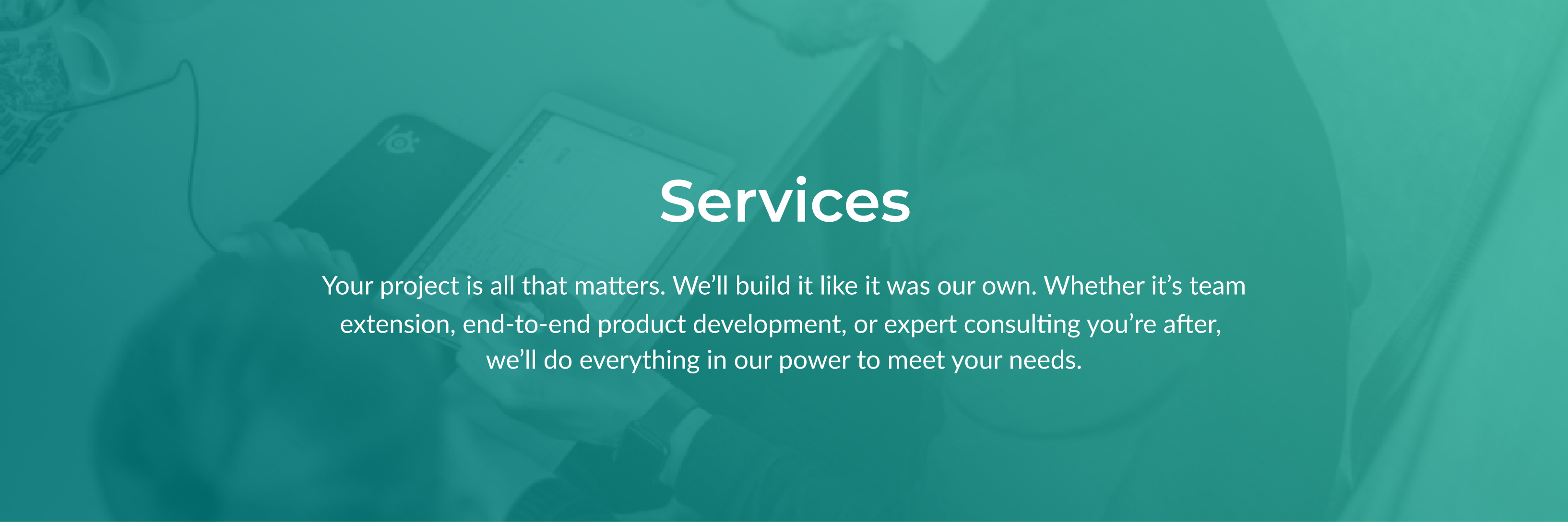
Tech Leaders Hub: Management & Growth

Become a great leader by leveraging the huge experience of tech leadership experts who have spent years managing and growing teams.

The True Cost of Hiring In-House Developers

From training through benefits to paid leave, the cost of adding new members to your team is never just the salary.

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

REVIEWED ON Clutch ★★★★★ 82 REVIEWS

python SOFTWARE FOUNDATION

TOP PYTHON & DJANGO Clutch DEVELOPERS 2021

TOP DEVELOPERS Clutch POLAND 2021

TOP CUSTOM SOFTWARE DEVELOPMENT Clutch COMPANIES 2021

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