The Developer Diaries

An Insider's Guide to Building Happy, Effective Extended Development Teams

zartis

www.zartis.com

Table of Contents

Building a development team is a formidable task in the best of circumstances. Building an extended development team, however, takes things to a whole new level. You don't need us telling you that remote work presents a number of unique challenges, such as:

- Developing an effective remote onboarding process
- Ensuring efficient and clear communication at all times
- Striking the right balance between micromanagement and zero oversight
- Making sure that everyone's input is taken into consideration when making key decisions ... and much more.

That said, remote engineers can be invaluable assets to your company — as long as you, the manager, take steps to promote a healthy and positive corporate culture.

The question is, how do you do that?

Don’t worry. We’ve got you covered. We picked the brains of some of Zartis’s top developers, who told us what works and what doesn’t — all based on their real-life experiences working in extended development teams. We then distilled their key insights into a handful of simple, actionable steps that you can start implementing today to radically transform your business for the better.

Here are the nine most important red flags to watch out for — and what to do instead to keep your extended development team happy, thriving, and performing at maximum capacity.

Just a side note before we dive in: most of the tips below apply to building happy and effective software development teams — both extended and in-house, so you get to kill two birds with one stone!

Red Flag #1: Business Always Comes First

Hey, we get it. The main goal of your company is to make a profit. All aspects of the business should be aligned with that goal.

But — and this is a "But" with a capital B — to make a profit, you need to build first-rate software products. Otherwise, you'll have nothing of value to sell.

And you just can't do that without excellent engineers, can you?

Unfortunately, time and again, we see managers who put business considerations first and their developers second. They prioritize the input of non-technical departments and cater to the demands of investors — who often have little engineering skills or insight themselves.

The result?

Subpar, rushed products. A performance that degrades over time while sucking in more and more resources for things to just stay afloat. Demotivated engineers handing in their resignations en masse.

Don't despair just yet, though. The good news is that this problem has a very simple solution.

But before we get into that, let's take a look at the telltale signs of a business-first culture.

Things to Watch Out For

- The product departments have limited technical knowledge.
- It is the sales and marketing departments that decide what new features should be built and set hard — and often unreasonable — deadlines for the engineering team.
- The requests for new features rarely take into account non-functional requirements such as performance, scalability, or testing coverage.
- Developers are expected to churn out additional features one right after the other, leaving no time for discussing different solutions and approaches.
- Engineers are afforded limited time to fix technical debt items due to monetary or business considerations.

What to Do Instead

It's simple, really. Something has got to give — on both sides. A little **inclusive decision-making** goes a long way.

While you can't afford to disregard business considerations, be sure to always **seek the input of your technical department when brainstorming ideas for new products and features or setting time-to-market deadlines.** Design discussions and whiteboarding must be key aspects of the decision-making process and must involve the engineering teams.

In addition, the **sales and marketing departments should work closely with the engineering teams**, sharing their insight into what the customer wants and where the market is headed. This way, everyone will be on the same page, and you will end up with a product that customers love.

The last point is crucial. Your marketing department would do a much better job advertising the product if they have a better grasp of its functionalities and technical characteristics. By the same token, your engineers would probably benefit from a more in-depth understanding of the overall state of the business and the ever-changing market demands. What’s more, they could offer valuable insight into how certain technological solutions can help further the company’s vision.

And don’t forget your **Product Managers**, either. Remember: engineers and marketers speak what can be very different languages. You therefore need people who are both business- and tech-savvy to help bridge the gap between your devs and sales departments — and that is precisely what a great Product Manager is supposed to do. So, make sure you have at least one highly skilled Product Manager to drive the product roadmap in a way that benefits sales and is also realistic from an engineering perspective.

Last but not least, you want to avoid setting hard and unrealistic deadlines. First-class products may take longer to develop, but the rewards are well worth the wait.

That said, though, you probably cannot afford to take all the time in the world. The secret to success here is to have your Product Manager define **a clear and detailed roadmap**, taking into account the input of the engineers but also that of the sales team, your customers, and other stakeholders where appropriate. The roadmap should outline the key priorities at every stage of the process, making it crystal clear which functionalities or steps are essential and which can wait if needed.

Red Flag #2: One Person Calls All the Shots

Charismatic leaders are a double-edged sword. Their strong personalities can inspire and motivate people. However, with software companies, they can also get developers used to relying on a single person to architect the entire solution and make all important decisions on tooling and the technology stack. At Zartis, we jokingly refer to this phenomenon as "the God-decision culture."

Sometimes, it's not a single person but rather a small, tight-knit team of architects or tech managers who get to call the shots — and who may be a few years removed from hands-on experience with coding or delivering the solution.

In either case, however, the input of the other engineers in the company is fairly limited — which stifles innovation, lowers morale, and leads to a lack of trust between the architecture/technical management and development teams.

How do you know you've fallen into the God-decision trap?

Things to Watch Out For

- Key technical decisions are made by one person or a small band of engineers or tech managers — typically, the architects.
- Decision-makers often do not work closely with the product and may not have the expertise required to make fully informed choices on their own.
- The decision-making process is not transparent. Choices and solutions are rarely discussed with the engineering teams, if at all.
- Communication of the reasons behind technical decisions is usually poor.
- The responsibility for fixing any problems that arise because of bad technical decisions lies with the engineering teams and not the original decision-makers.

What to Do Instead

The solution is in the three Ds: discuss, diversify, and delegate. You want to open up the technical decision-making process to as many different (and expert) viewpoints as possible. Not only will your engineers have valuable insights about new products and features, but they can also predict potential pitfalls, saving you a lot of time and resources in the long term.

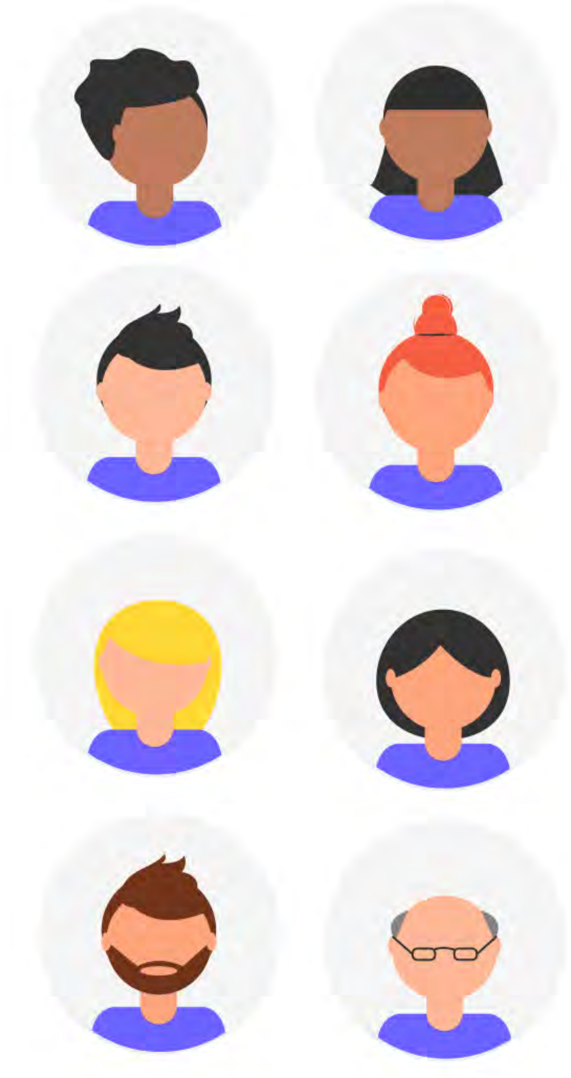
You also want to have clear lines of communication, issue regular company-wide updates, and encourage different teams to work together and get to know each other's perspectives.

How We Do Things

At Zartis, we believe that the best solutions are found by combining different points of view. In our experience, two heads really are better than one. This is why we make a point of inviting all our team members to share their perspectives.

> "Finally working in a company where the management truly hears you out. I love working here for the fact that any idea is recognized, evaluated, and you actually get to follow it through to reality. The management team is also very aware of each individual's success and is keen to help you progress your career within the team."
>
> — **Current Zartis employee in Berlin** Source: Glassdoor

Red Flag #3: One Too Many Yes-Men



It's lovely when everyone on your team gets along and is on the same page. However, the problem is that it's almost... too lovely. As a manager, you may fall into the temptation to surround yourself with yes-people.

Not only does that stifle healthy debate and constructive criticism, but it also cuts the line of communication between you and your developers. Middle managers may fail to flag important technical issues just to avoid having lengthy discussions, keep you happy, and save face. That may seem like a quick fix but is extremely unproductive in the long term, resulting in frustrated engineering teams and misuse of the chain of command.

Things to Watch Out For

- Middle management seems to always agree with you and do not raise concerns flagged by the technical teams.
- Middle managers may take credit for positive developments while blaming the engineering team for mistakes and failures.

What to Do Instead

To avoid all that, maintain an open door policy. **Make sure that everyone on the development team knows you are always available to answer questions or discuss any concerns they may have**, in person or online. Don't forget to also actively request their input when making key technical decisions.

How We Do Things

We want everyone to feel included in the decision-making process and get a chance to grow as a professional — regardless of their job title.

> "People actually matter in this company. They are inclusive, smart, nice people to work with. Even the CEO is a very approachable person! No matter how long you've been in the company or what your position is, if you want the chance to make the company grow, evolve, improve... if you put yourself to it, you can get it. There's room for you to shine." — **Current Zartis employee in Madrid** Source: Glassdoor

Flag #4: The Cool Kids Club

We are all human. And you know what's all too human? Relying on your personal relationships with your superiors to fast-track your career.

Look, we are not saying that's happening in your company. But it's not an uncommon occurrence, either. Similarly to the yes-men scenario above, managers often unwittingly find themselves surrounded by all-too friendly employees. These employees avoid challenging the decisions of their superiors out of fear that it might undermine their personal relationship and endanger a potential promotion further down the line.

Eventually, they may well get that promotion. Unfortunately, that hinders the career growth of your development team, and great people and ideas may get passed by.

Things to Watch Out For

- Decision-makers have strong interpersonal relationships with middle managers or specific engineers but remain on formal terms with the development team as a whole.
- There seems to be a pattern of hiring or promoting people based on friendships rather than merit.
- Middle managers avoid challenging their superiors.

What to Do Instead

Keep reminding yourself that if something sounds too good to be true, it probably isn't. The chances are that you aren't right 100% of the time. **Always seek the input of your development teams** and invite everyone to voice their criticism and concerns. They may raise valid points you haven't considered, which will improve the decision-making and development process, and, ultimately, the finished product.

Of course, sometimes, your engineers will not be right. If that's the case, take the time to listen to what they have to say and then explain why you — respectfully — disagree. That will improve their understanding of the overall picture and increase engagement (or, at the very least, minimize any feelings of unfairness or discontent).

Don't forget to also **implement the best industry practices for ensuring fair and objective recruitment, performance review, management, and promotion policies.**

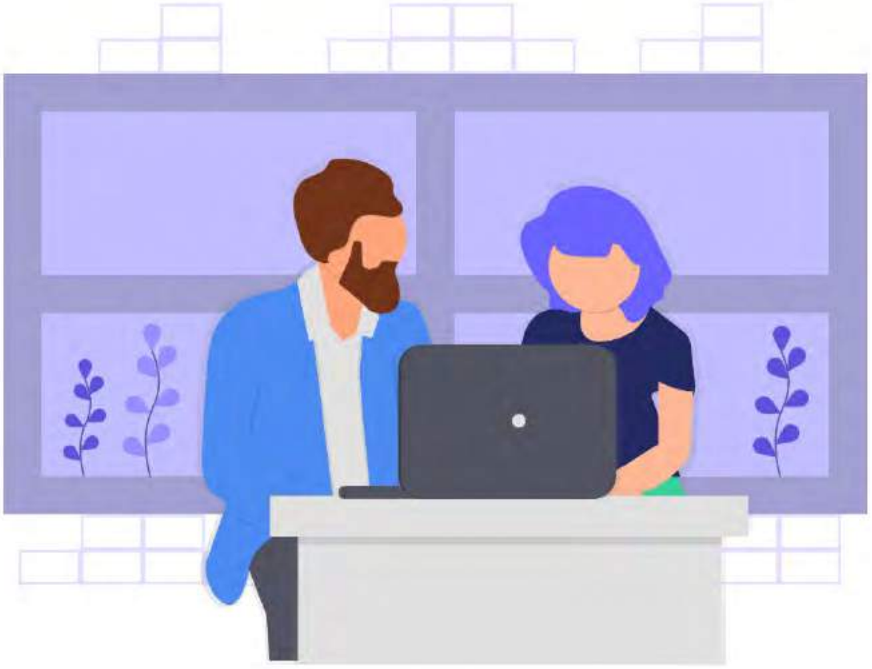
Red Flag #5: Blame Culture

Blame culture is very common — which is surprising, given that it is not at all helpful. Technical mistakes are inevitable. The trick is to handle them in a way that is supportive, constructive, and encourages learning and growth. But if everyone keeps blaming the developers instead, you have a problem.

A blame culture creates a toxic environment where your engineers avoid taking creative risks and suggesting changes out of fear that the slightest mistake might get them in trouble. Teams will be content with having less responsibility to keep the margin of error as small as possible. As a result, innovation will stall, and your company will lose its competitive edge.

Things to Watch Out For

- When mistakes are encountered, the focus is on finding the responsible party and having them fix the problem.
- Teams tend to cover up their mistakes, fail to address problems upfront, and do not share troubleshooting know-how with their colleagues.



What to Do Instead

Stop pointing your finger at people — and get everyone else in the company to do so, too. Aim to transform every mistake or so-called failure into a learning opportunity and a chance to gain a fresh perspective. **Organize training** on how to fix common issues and encourage your engineers and other team members to share information on errors they made and how they fixed them.

Last but not least, **have the company's KPI measurements focus on outcomes rather than outputs**. That would make the whole vertical company responsible for undesirable outcomes. One way to do that is by implementing an OKR system — a tool used by numerous tech leaders, including Google, Facebook, Netflix, Eventbrite, and many others.

Red Flag #6: Too Much Micromanagement

This is something we often see in consultancy companies. Managers track and obsess over the time and resources their engineering teams spend even on minute tasks and projects. All that does is lead to a lack of trust, low morale, and frustration among developers.

No one likes to be watched 24/7, and focusing on metrics too much leaves barely any room for creative solutions. Plus, if you hired someone, the assumption should be that they can do their job efficiently, to a high standard, and without wasting resources.

Things to Watch Out For

- Managers work with detailed spreadsheets that track time and the use of other resources way too closely.
- Engineering teams are treated as a mere resource rather than creative thinkers.
- Managers do not trust their engineering teams and demand to know what everyone is doing at all times and how much time is spent on every task.

How we do things

At Zartis, we don't believe in micromanagement. What we do believe in is building a work environment that is supportive and inclusive but also encourages our engineers to continuously develop new skills and grow as professionals.

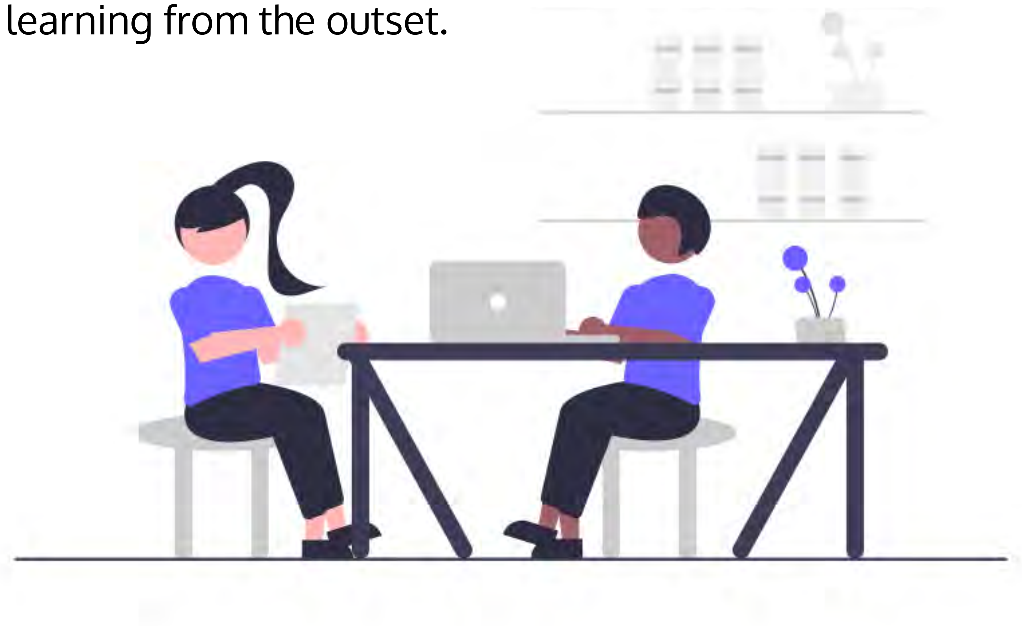
> "They say 'If you want to go fast, go alone. If you want to go far, go together.' Our work as managers is to build trust, provide a proper environment, and help our team achieve their goals. The reality is we can't be everywhere, but the company culture can." — **Angel Benito, CTO of Zartis**

What to Do Instead

Take a deep breath and let your developers do their job. You hired them for a reason. Your role as a manager is not to control but to **enable, coach, and support your technical team**. So, don't micromanage them excessively and try to delegate responsibility and empower them as much as possible. The advice from point #5 to focus on outcomes rather than outputs would be hugely relevant in this situation as well.

Red Flag #7: Too Many Junior Developers

There is nothing wrong with hiring fresh talent. However, junior developers require continuous mentoring and coaching. So, if you have less experienced engineers on your team, be sure to allocate enough resources to guarantee a thorough onboarding process and ongoing follow-up training. The goal here is to enable a growth culture and encourage young developers to develop a mindset for lifelong learning from the outset.



Things to Watch Out For

- Too many junior developers on the team that seem to neither evolve nor learn.
- Engineers are content with relatively straightforward tasks that leave no room for innovation.
- Tasks are assigned with detailed how-to instructions that don't encourage developers to do independent research and think critically.
- Junior engineers have limited or no opportunities to seek guidance and mentoring from more experienced coworkers.

What to Do Instead

The challenge here is to find the delicate balance between guiding new hires and micromanaging them to death (or not investing in their professional development at all).

Once the initial onboarding and training process is over, **give out tasks that tell engineers what to do rather than how to do it** — but be sure to let them know that they can ask any questions if something is not clear. Encourage independent research and continuous learning and consider implementing a mentorship program that pairs young developers with experienced engineers.

How We Do Things

We continuously invest in our junior developers and all our employees. We provide them with ongoing training opportunities, encourage independent learning, and are always open to suggestions and exploring innovative solutions that can help everyone on the team grow into their full potential.

> “Great people, great environment to grow and try out new ideas, great opportunities to those who want to take lead on anything that could be beneficial at the end of the day. Love the mindset of the management and how the company is open to constant change.” — Anonymous reviewer in Berlin Source: Glassdoor

Red Flag #8: Limited Training Opportunities

If your developer training budget is tight or nonexistent, you might want to reconsider that. While for the most part engineers learn by feeling their way around new technologies on their own, formal training makes sure everyone is on the same page and shows them that you care about their growth.

Otherwise, you risk having the skills of your developers becoming outdated. Their motivation will also suffer in the long run, as they won't be able to take your product to the next level and keep it up to date with modern practices and technologies.

Things to Watch Out For

- There are limited or no training opportunities for developers.
- The training budget is low and not enough to cover everyone on the technical team.
- There is no time allowance for organized training during working hours. Self-training is promoted as a free-time activity only.

What to Do Instead

Forget the question, “What if we train them, and then they leave?” Instead, ask yourself, “What if we do not train them, and they stay?” (Yes, we do realize this cliche is as cheesy as it gets and you’ve probably heard it a thousand times, but it bears repeating — it’s painfully true!).

With this in mind, make sure to **enable your developers to continuously gain new skills and discover new technologies during office hours**. In turn, they will share their insight on how to best utilize them and how to modernize your existing stack and software.

And while it’s not your job to train your extended teams, **think of ways your in-house and extended teams can benefit from sharing knowledge.** Are there skills your extended team can pass on to your internal team through training, mentoring, and webinars? Can the in-house team share their deeper product knowledge through more detailed onboarding?

And lastly — if your stack is evolving and you need your developers’ skills to match, let your extended team provider know in advance so they can organize appropriate training.

How We Do Things

At Zartis, not only do we organize training for our developers but also actively encourage them to do independent research, exchange ideas, and constantly seek ways to sharpen their skill sets.

> "My work at Zartis provides me with the opportunity to explore new technologies and at the same time keep learning, improving, and challenging myself on a daily basis. That was just what I was looking for."
>
> — **Leticia Garcia, Senior Software Developer**

Red Flag #9: Fake Agile Culture

Agile has taken the world by storm. It has revolutionized the way software companies are structured, the way they develop software, and — from a broader perspective — the way they are able to adapt to changing markets and customer requirements through a fast-paced process of flexible evolutionary development and continuous improvement.

However, if implemented incorrectly, agile can do more harm than good and create discontent within your teams. See, becoming agile comes with a specific mindset that changes the way you think about software development. If you fail to adopt that mindset, your new agile methodology may become more of a burden than an asset.

What are some telltale signs of a fake agile culture?

Things to Watch Out For

- Useless standups.
- There is no time for retrospectives, and not everyone jumps in.
- Retrospective results are ignored from one session to the next, and no follow-up action is taken.
- Useless software grooming and refinement with no real whiteboarding or design solutions.
- The agile process is not vertical, which leads to prioritizing sales and marketing rather than delivering great software and customer collaboration.

What to Do Instead

For starters, take the time to implement agile processes properly and in a way that is best suited to the needs of your team. **Plan and lead meetings in a productive way.** Put more focus on retrospective results and always outline exactly what actions need to be taken next — and by whom. Furthermore, be open to revising old practices if certain meetings or standups become redundant. A vertical agile process and a software-centric approach are also crucial.

How We Do Things

Curious to learn how we at Zartis implement agile in our internal processes? Here are a few insider tips:

> "There are four pillars to a good agile process:
>
> 1. **Team agreements.** Make sure the whole team is aligned.
> 2. **Collaboration.** Offer your help to others.
> 3. **Trust.** Without micromanagement, be honest and raise your issues as soon as possible.
> 4. **Any failure is an opportunity to improve.** Retrospective is probably the key piece of agile."
>
> — Ignacio Lopez, Team Lead at Zartis

Need Help To Build a Happy Extended Development Team from Scratch?

Do you want to:

- Retain technical talent?
- Develop best-selling software products?
- Keep your engineers and, by extension, your customers and investors happy?
- Build a positive corporate culture that encourages growth, innovation, and excellence while maintaining a healthy work-life balance?

We can help.

Zartis specializes in assembling high-performing remote development teams for clients in Ireland, the UK, Germany, and the United States.

LET'S TALK

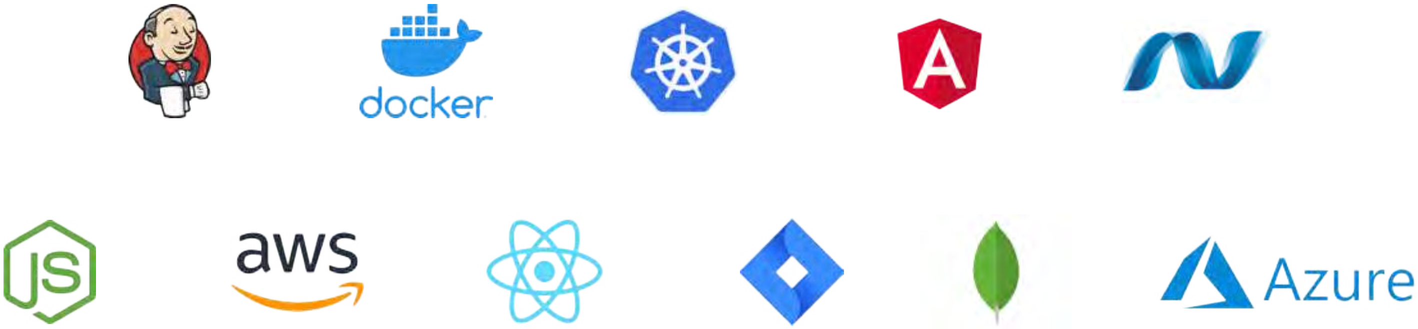
www.zartis.com

Office Locations

We will handpick the best engineers for your project from our software development centers in Spain, Poland, and Portugal. They can work alongside your core team or help you transition to working exclusively with distributed teams.

Technological Expertise

All our extended developers are certified to work with the following (to name just a few out of 100+ technologies):



Key Stats & Figures

At Zartis, we like metrics. Here are some stats that testify to the quality of our work:

- We have a stellar 4.8/5 rating on Glassdoor.
- The retention rate of our extended developers has been 90% for the past four years.
- The average length of tenure for our engineers is 2.5+ years.
- Over 50% of our engineers are at Senior, Principal, and Lead level of experience, with a strong background in architecture and management.

What People Say About Us

We don’t mean to blow our own horn, but our engineers, clients, and partners love working with us. They appreciate our international and multicultural team, as well as our dedication to Zartis’s core values: quality, innovation, integrity, diversity, and a people-friendly culture.

But you don’t have to take our word for it. Here’s what people say about us:

> “One of the best companies to work for.”
>
> — **Former Zartis employee in Berlin** Source: Glassdoor

> “Great place to learn new technologies, have fun.”
>
> — **Senior Sharepoint Developer in Madrid** Source: Glassdoor

> “The company offers a variety of projects, and before participating in the recruitment, I had the possibility to choose what I find more interesting. The first day at Zartis is warm and welcoming, everybody is there to help you out and explain all issues — from invoicing to machine setup. Within the customer’s domain, I also found I can always reach out to people from Zartis if I have questions or face difficulties, and they were always there to suggest a solution.”
>
> — **Alicja Kempa, Software Engineer**

> “Colleagues from other projects are in touch with each other so you don’t feel like you’re working on your project and there’s no one else there. Also, their family-friendly policies are great for people who need to take care of their relatives in this COVID-19 time.” — **Data Engineer in Madrid** Source: Glassdoor

> “For many of us, our strongest desire is to be more creative developers and work in a team that’s highly engaged with the quality of the product. That means being in a team of great listeners as well as kind critics. Zartis’s success is due not only to its passion for technology and engagement with the products but also to its values of self-development, respect, tolerance, and kindness. When a company is that dedicated to its employees, going to work every day is a real pleasure.”
> — **Leticia Garcia, Senior Software Developer**

For more positive reviews, check us out on Glassdoor!

Want to Know More?

For further details on our extended software team services,

Drop us a line

We will be in touch shortly.