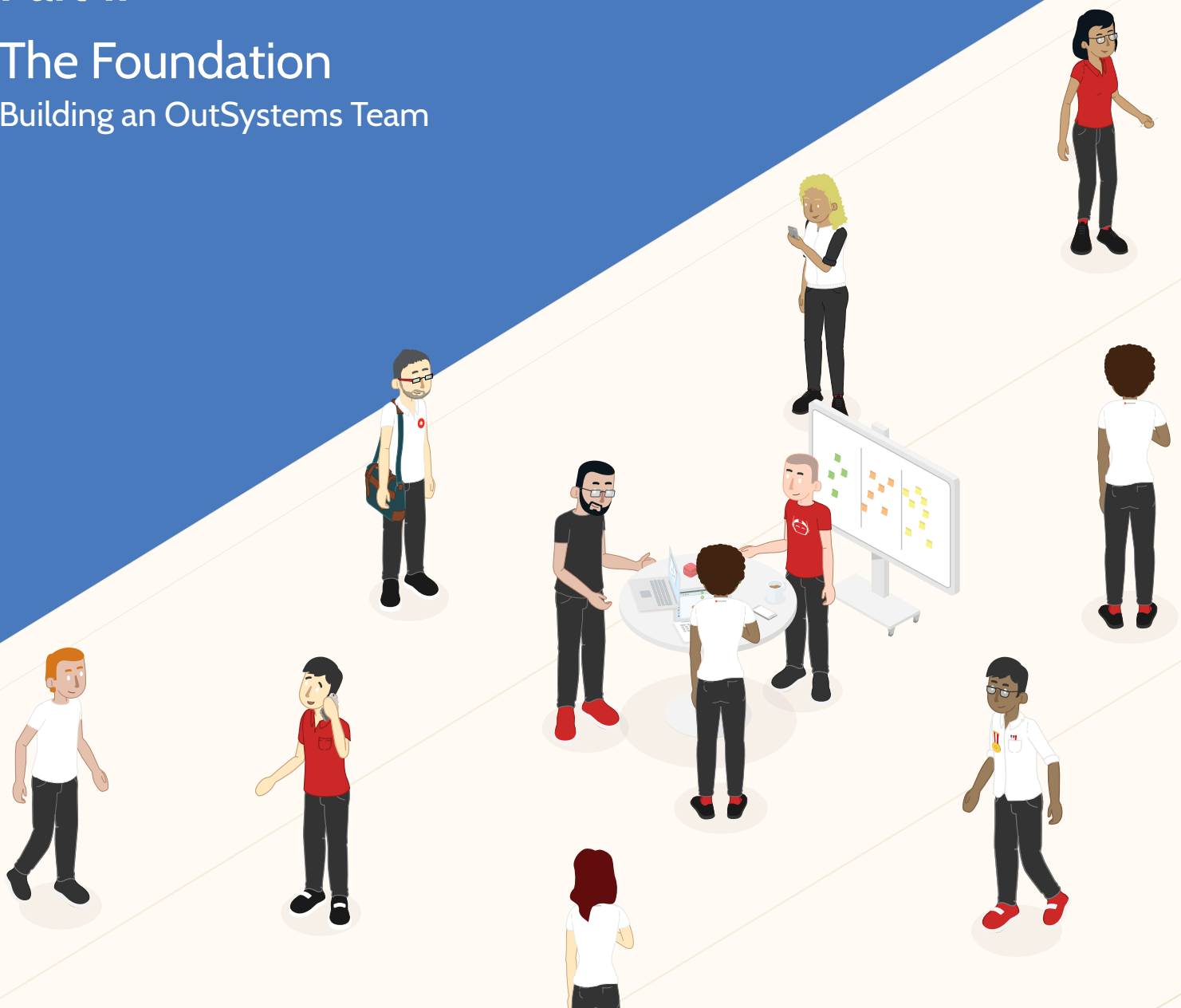


The Talent Playbook

Part 1:

The Foundation

Building an OutSystems Team



Contents

Introduction..... 4

What Is This Playbook For?

Who Should Read This Playbook?..... 4

How This Playbook Is Organized..... 5

Identifying Talent 6

Mindset	7
Skillset	7
Developers Role and Their Skills	8

Who Should You Target 10

Why Developers Should Adopt OutSystems . . .	11
Overcoming Objections	11

Some Tips for Looking for Talent . 13

Looking Within	13
Bringing in a Partner	14
Going to the Market	15

Mix and Match 18

The Team's All Here..... 20

Training and Onboarding Developers	20
Resources for Developers	21

Future-Proof 20

The OutSystems Developer Career Path

Conclusion..... 24

Appendix 26

A – Competence Skill Level Breakdown	26
B – Job Description Template.....	30
C – Tech Exercise Example	31
D - Developer Testimonials	33

Introduction

What Is This Playbook for?

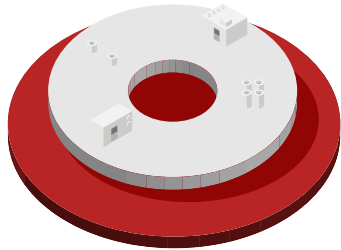
Introducing low-code technology in an organization is a significant undertaking best addressed with a framework called the low-code digital factory.

This framework has four dimensions: structure, talent, ecosystem, and process (STEP). It offers what you need to put the right organizational structure in place so you don't hit a wall and can respond to change at the speed the business demands.

These dimensions are explained in the [Digital Transformation Playbook](#), but this guide expands on the “T” in the STEP framework to help you assemble the best team for using OutSystems.

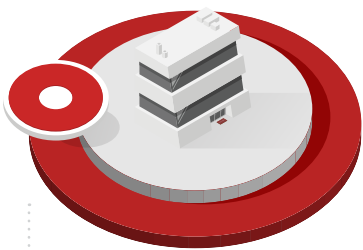
How This Playbook Is Organized

To build an OutSystems team, you have three options:



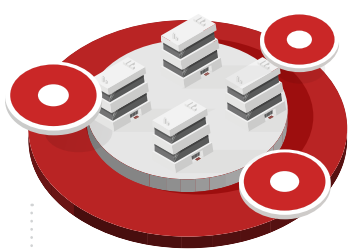
Look Within

People in your team have the right skills to take on the challenge.



Bring in a Partner

An OutSystems Partner brings in a team with the right skills.



Go to the Market

Hire a new team with the right skills.

Looking within: This approach capitalizes on people who already know your business and company culture. You identify who on your team has the skills to work on your OutSystems projects and provide them with the right learning path.

Bringing in a partner: Many organizations work with partners for faster delivery and added business and technology expertise. OutSystems Partners are a great option for delivering great projects. OutSystems partners have deep industry knowledge, technical expertise, and local presence so you can choose one that's right for you.

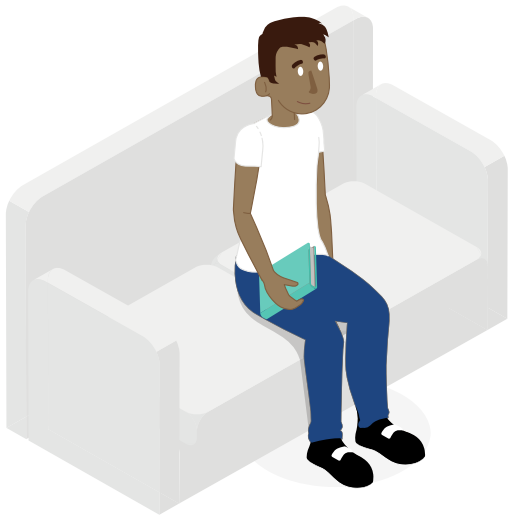
Going to the job market: If you don't have many in-house resources, you can hire your talent by preparing the perfect profiles, using the right interview techniques, and evaluating candidates for junior and senior roles.

This playbook offers guidance for all of these options: bringing in a partner, [looking within](#) and [going to the market](#). The latter two approaches require a similar process of [identifying talent](#): looking for the right **mindset**, looking for the **same skills**, looking for the **right developer** for your needs, and answering the same **initial questions discovering new technology**.

Therefore, the playbook is organized so that you can see what is required on an OutSystems project team and then get additional information on how to succeed if you're looking inside your organization or if you're hiring in the market. This is followed by a section on [ramping up your team](#) with guidance on **onboarding** and a list of some great **resources**. We end the playbook with a peek into the future, giving you ideas of what OutSystems [career paths](#) can look like.

Who Should Read This Playbook?

If you are a recruiter, hiring manager, or technical leader and your organization is planning to use the OutSystems low-code platform, this guide can help you build the right team. Or if you find you need to add or backfill a developer, you can use this guide as a reference.



Identifying Talent

This section covers the essential skills and attitudes needed to develop with OutSystems and how to pitch OutSystems technology and overcome initial questions.



This knowledge is the result of observing and working with hundreds of customer and partner development teams to understand and identify the essence of a successful OutSystems team.

In this section, we've broken down the attributes and target profiles for what we call the "developer persona." If the person you are evaluating for your OutSystems projects does not have this persona, then we strongly advise continuing your search until you find someone who does.

To be clear, it is not ideal to start a low-code journey with a citizen developer. Instead, we recommend starting with team members who have the right mindset, are comfortable with development, and demonstrate development skills. For them, OutSystems will be a great match. Once the foundations are there, then you can add citizen developers to the team.

Mindset

With the right mindset, a developer is likely to succeed with OutSystems. The easiest way to ascertain this is to ask: *What is the job of a developer?* If response is, *"To deliver solutions to my client or business unit"* or something along those lines, this is a good sign. Most likely, the developer is open to using various tools and focused on the outcome and not the way to get there.

Our experience tells us that developers who are more successful with OutSystems like to try new tools and are excited when a final product is in the hands of its rightful user. They want to succeed in the transformation of their organizations and to help the business to grow faster.

Skillset

What skills should developers who use OutSystems have? Let's start with the basic requirements and a "nice to have":

- **SQL (requirement):**
Relational databases are key. The more knowledge a developer has in this area - basic queries, joins, and so on - the better.
- **Web development (requirement):**
Any type of web development should be also a requirement. If people do not understand how the web works, it can lead to poor decisions.
- **Mobile development (nice to have):**
Because mobile development has its own set of considerations, having someone with these skills on your team is a plus. If your first projects include mobile apps, having someone with mobile development experience on board can help ensure success.



Developer Roles and Their Skills

Now that you have an idea of the mindset and skills that all successful OutSystems developers should have, we can get a little more granular. Depending on the type of project or projects you have planned with OutSystems, you are likely to need your team of developers to take on some specialized roles. Regardless of the developer type, the level of proficiency related to algorithms and SQL knowledge will largely influence the performance of any OutSystems developer.

Developer

Problem-solving and logic are the foundation for algorithmic prowess and are the essential tools for a good OutSystems developer. This graphic indicates the skills, experience, and expected responsibilities of a developer on an OutSystems project team or in a low-code factory.

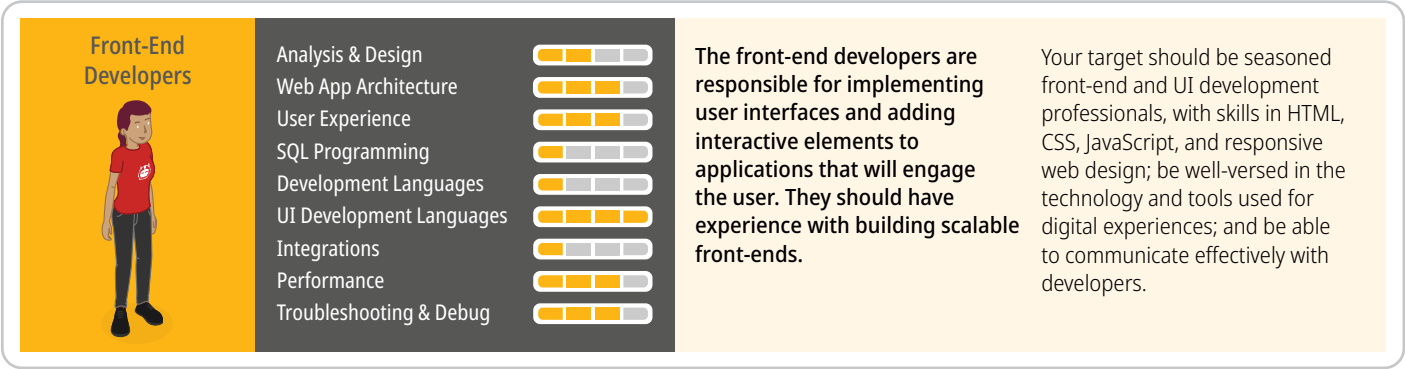
The yellow bars indicate the level of proficiency they should have to fulfill this role. As your developers increase their proficiency in certain skills, their responsibilities are likely to evolve along with the progression of their career. For more details, check out the [career path](#) section of this playbook.



Front-End Developer

This next graphic shows the skills, experience, and expected responsibilities of a front-end developer on an OutSystems project team. For this type of developer, you see a number of the same requirements as a developer, but they are

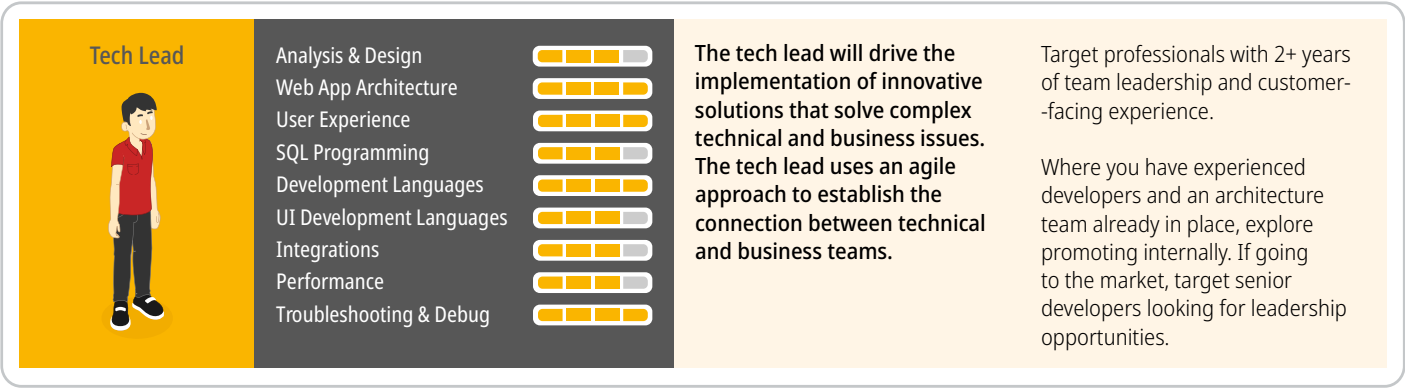
expected to have more experience with analysis and design, web app architecture, UI development languages, along with added HTML, JavaScript (required for mobile apps), CSS, and responsive web design.



Tech Lead

The tech lead role is crucial for the success of a low-code project or a low-code factory. Whether you look within or go to the market, your tech lead needs to have a strong

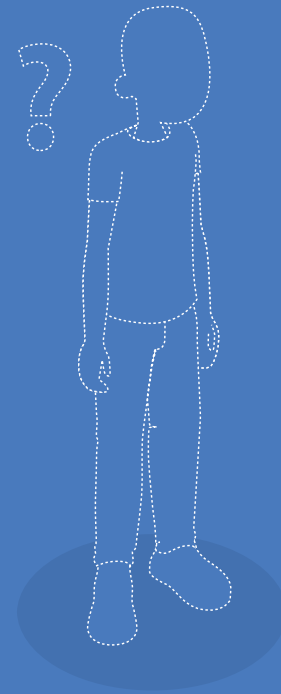
technical background and be able to design app-level architecture and operational tasks. Experience in leading a team is also a must.



| **Note:** You can find details about all these skills in [Appendix A](#).

Who Should You Target?

Now that we've identified the technical skills required to succeed, we would like to share some of the components that are important for you to assess once you've found someone to interview.



For example, who do you target to recruit? What kind of developer is ideal to pursue for long-term success? The short answer is that any developer with a solid understanding of the technical skills we've shared and the right mindset would make a great addition (assuming they meet your own cultural and team requirements as well).

Traditionally, any C# or Java developer could be your target audience. That said, we've seen experienced programmers who have worked with technology like Lotus Notes, Visual

Basic, VB.Net, and Coldfusion make excellent OutSystems developers. These are also the types of technology that developers view as "dreaded" coding languages, according to the Stackoverflow developer annual report, and therefore, candidates with these skills might be easier to lure to a newer technology.

Whether they're currently using new or older technology, it's worth noting again that the right mindset - one that is solutions-oriented - is key to adoption.

Why Developers Should Adopt OutSystems

OutSystems enables developers to create complex, robust applications in a fraction of the time. They can focus on solving problems, creating solutions, and understanding business logic, instead of spending time on mundane, repetitive tasks.

Developers have the flexibility and extensibility to be creative without limitations, while using templates and widgets to make the repetitive functions simple. The apps they create using OutSystems are secure and easy to maintain without the extra leg work.

OutSystems developers are proud of their work and are only limited, in both development and career, by their imaginations. However, it is likely that you will run into developers who are nervous about adopting OutSystems and/or have heard things about low-code that make them hesitant to try it.

So, the following section covers the common objections developers have and some examples of how you can overcome them.

Overcoming Objections

You can help developers be more open to adopting OutSystems by enabling them to understand how their current experiences stack up to a new platform or tool. To help you help them, here are specific questions that developers ask when evaluating OutSystems. For additional responses to why developers adopt OutSystems, see [Appendix D](#). If you'd like to go beyond these questions, check out our [evaluation guide](#).

1. Will I hit a wall when developing?

When people ask us this question, they have often just experienced our visual application development model or seen a demo and are concerned about whether they will be able to include their own code. They worry that they will encounter platform limitations later in the development process that will prevent them from building enterprise-grade applications, designing rich and complex mobile and web user interfaces, or integrating with the myriad of systems and databases that exist in their environment.

OutSystems is open by design to allow [all layers of applications to be extended with your own code](#): front-end, back-end, database, and integration. In addition, OutSystems offers an expressive visual language for developing your applications and [avoiding rapid application development pitfalls](#).

2. Can OutSystems be used to build complex and big applications?

Yes. OutSystems offers a fully integrated approach to development, quality assurance, deployment and management - the entire application lifecycle - which enables it to support high-scale app development.

OutSystems developers can [create robust architectures](#) that support their core businesses and applications that scale to millions of users. It works on top of a customer's preferred middleware stack and scales from small applications to large enterprise installations. [Governance](#) is part of this approach.

A number of our [customers](#) have built complex portals and systems on OutSystems, and [Forrester](#) acknowledges that we are a top choice for scalable architectural development.

3. If something goes wrong, how can the platform help me pinpoint and fix the problem?

Comprehensive auditing and monitoring tools make it easy to manage application performance proactively and to detect problems by allowing the identification of performance issues in real-time. A performance monitoring dashboard delivers application-level analytics. A business activity monitoring dashboard handles business process and workflow monitoring. There are also logging and analytics APIs. All information and metrics are provided by default without any extra work and without introducing any overhead to the runtime environment. Plus, any console that uses web technology can be integrated with OutSystems by means of a plugin that allows for the addition of new customized tabs.

4. Can I build pixel-perfect applications with OutSystems? Do I have any limitations with the UI editor?

OutSystems enables developers and teams to create amazing user interfaces that work across all devices. In the visual designer, they're able to design and tweak the user experience to what's needed without having to be HTML, CSS or JavaScript experts.

As new devices and new application designs evolve, new interaction patterns appear. Therefore, OutSystems includes widgets that allow developers to create beautiful interactive applications that include chat, bots, and voice and that offer an omnichannel experience so that a user can switch devices in the middle of a process with repeating steps.

There's no need to write custom code; however, to make sure developers aren't limited in any way, OutSystems offers the option of using standard [HTML](#), [Javascript](#), and [CSS](#) to [extend their applications as needed](#).

5. How long does it take to learn OutSystems?

Experienced developers can learn OutSystems in a couple of weeks because it's built on top of universal software development practices. So, basic programming acumen and an understanding of relational database concepts are all that is required to master OutSystems out of the box.

Also, with OutSystems, developers are not alone. Along with the [OutSystems training](#) and certifications, they have a large, active community to help [answer questions](#), build [reusable components](#), and continuously learn and grow with.

"OutSystems offers a lot of abstraction but it doesn't mean you don't have to understand web/mobile architecture. One of the advantages is that it saves you a lot of plumbing work, allowing you to focus on the challenging things."

Developer Community Advocate

"I am an old-school-former-oracle-consultant making the transfer to OutSystems. Working with 44 Certified Professionals for Transfer Solutions, an OutSystems Global Training partner. With my Oracle and OutSystems knowledge, I'll be able to solve Legacy challenges and leverage those applications to a higher level."

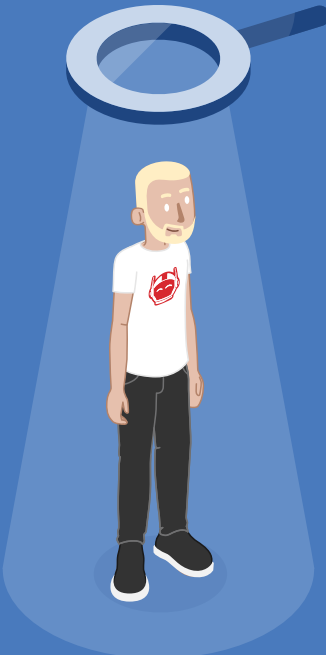
Consultant @ Transfer Solutions

"I no longer had to worry about semantics and infrastructure. I could turn my PC off at the end of the day and do not need to worry about it again until tomorrow unless I wanted to chase my own personal projects. OutSystems has been great for my physical, mental, and emotional well-being. It has had the added benefit of returning my pride in my work and it has greatly boosted my professional self-esteem. There is something about this tool that makes me feel empowered and it has become basically an extension of myself."

Developer in OutSystems, Minneapolis MN, USA

Some Tips for Looking for Talent Inside and Out

As mentioned at the beginning of this playbook, when finding talent for your OutSystems project or low-code factory, you can look within your IT organization or team, bring in a partner, or go outside your organization and search the market.



Let's take a look at some tips for all three.

Looking Within

Using an internal team, one that already knows your business, is the optimal solution for getting your low-code project or factory up and running in no time. But, how do you find the right people with the right skills in your company who are ready to take on the challenge? Often it is just a matter of setting up the team's expectations about the work to be done and the tools to be used.

Here are some tips to get started on the right foot:

- **Present a vision for the new project** and promote the positive impact it will have on your organization (for example, create more time for innovative projects).
- **Promote the opportunities for acquiring new skills** and illustrate the career path available for your team.
- **Advocate for OutSystems throughout the process** and make sure your team fully understands what this technology means for them as a team, for them as professionals, and for the business itself.

Bringing in a Partner

Sourcing talent from OutSystems certified partners is a great option. They combine the experience of delivery with rigorous process of training and technology validation. In other words, such a partner has a high level of commitment to OutSystems technology, an understanding of its value, and a lot of field experience. There are plenty of partners to pick from, with different experience and business vertical knowledge: it is of great importance to make the right choice based on your needs.

Selecting the right partner

Picking a local Partner is a great option because they are probably already familiar with your needs and culture and it is easier to deploy resources onsite when needed, reducing travel expenses.

Global partners cover wider territories with a worldwide network, multiple teams, and multiple delivery centers. This can be the right choice for global, enterprise-grade, complex projects.

is a good indicator of their consistency in delivering top-quality projects. Yours will be no exception. Partners are also rated by a customer satisfaction score (CSAT), based on independent reviews from their customers, which should also be considered when making a decision.

A full partner list that includes the number of OutSystems certified resources, industries and project references, can be found on our [website](#).

Team composition and certifications

When selecting a partner, check the number and level of their certificated professionals. A capable partner should have a balanced ratio of junior vs senior profiles (i.e. 1 Professional/Expert/Tech Lead - senior - for each 2 to 3 Associate developers - juniors).

Industries and project references

Partners usually focus on specific industries; therefore, it is crucial to use this as criteria when choosing a partner. Having a partner with in-depth knowledge of a business vertical will bring a lot of experience and value to your future interactions.

Past performance isn't always an indicator of future results, of course. However, selecting a partner who has multiple project references from existing OutSystems customers that demonstrate proficiency and success



Going to the Market

After evaluating all your options, you might realize that recruiting from outside your company is the best approach. You now know what to look for in your candidates, but where can you find them and how can you attract them? Here are some insights and tips.

Job description

As you are probably already aware, developers receive job offers every day, so you'll need a creative, compelling and accurate job description. This list of tips on what type of information to share with potential candidates when publishing a prospective job can help:

- **Promote the product or project** that you're starting as much as you can. Developers like innovation and challenges. Knowing your company is embracing digital transformation or has plans to conquer market share is attractive for any developer. Most developers don't want to be stuck in never-ending maintenance projects.
- Although this is often easier said than done, **include an estimate of the compensation range** and package that your company is willing to offer for the job.
- Be sure to **be clear about the hiring process and expected timeline**. Nobody likes the ordeal of a long vetting and interview process.
- Share as much as is feasible about your company, including mission, market differentiation, and career progression options.

| **Note:** A Job description template is available in [Appendix B](#).

Sourcing strategies

Once you have the job description ready, you should publish it on the usual job websites, your company career page, LinkedIn, and others. But there are some other creative places to share your job opening. Let's take a look.

OutSystems Job Board

Make sure you reach out to us and list your job opening on the OutSystems Community [Job Board](#). Numerous developers belong to our community, and many are interested in taking on new roles. It's also one of the best places to find developers with significant low-code experience.

Local job boards and Slack organizations

In many communities, developers have created their own meetups and Slack organizations to share information, network, and learn about new job openings. Even if some of these Slack organizations are invitation-only or need approval to join, many of the requests are approved right away once they gauge level of interest. Search the web for these groups, or ask a fellow developer if they are a member of one, and join the conversation! More organized communities will even have city-specific tech resources ([see this example from Atlanta, GA](#)). If one doesn't exist, why not start one?

Local technology-specific (non-OutSystems) meetups and user group meetings

More than 70% of people land jobs by networking. Seek out your local meetups (e.g., NET, SQL, Coldfusion) and user group meetings that align with your work and the profile of the person you are looking for. Below are some specific ways to use meetups to source the next members of your team.

1

Announce your job opening. Often organizers will allow attendees to stand up and briefly announce that their company has a job opening. Mention some fun and exciting aspects of the work they'll be doing to entice them to find you after the meeting.

2

Give a talk. Offer to speak at the local meetup on a topic relevant to the group's interest. Use this opportunity to demo a project you're working on that is related to the topic, and explain how it's accelerated or optimized with OutSystems.

3

Sponsor a meeting. The commitment varies by group, but it is usually a set amount for a certain timeframe (e.g., \$500 for 3 months of recognition). These costs typically go toward covering food and drink for the meetings, use of meeting space, and rental of any equipment, etc. Usually, your company receives recognition in the form of social media, branding on their meetup page, and a "call out" at the meeting itself. This can generate interest in your company and working for you.

4

Encourage members of your team to attend and present. Developers relate to other developers or techies. If people on your current team like meetups and want to go, let them. They can share their work and get other users excited. You can even encourage a group to go.

Share your knowledge (and job openings) on social media and in blogs

Your team works hard to build amazing applications and services for your users. By sharing the results or future plans on social media and successes in blogs on Medium or LinkedIn Pulse, you could catch the eyes of potential candidates. Writing blog posts and tweets or posting a video on YouTube builds credibility with candidates to see the impact and fun your team is having *while highlighting the tools you use*. When the time comes to add a new teammate, candidates will find evidence (or a lack thereof) in their search results that will either affirm or deny their decision to continue on in the hiring process.

Interviewing

Interviewing for an OutSystems developer requires the same skills as interviewing any other developer profile. However, following the guidance in this section can go a long way toward helping you hire an excellent OutSystems developer.

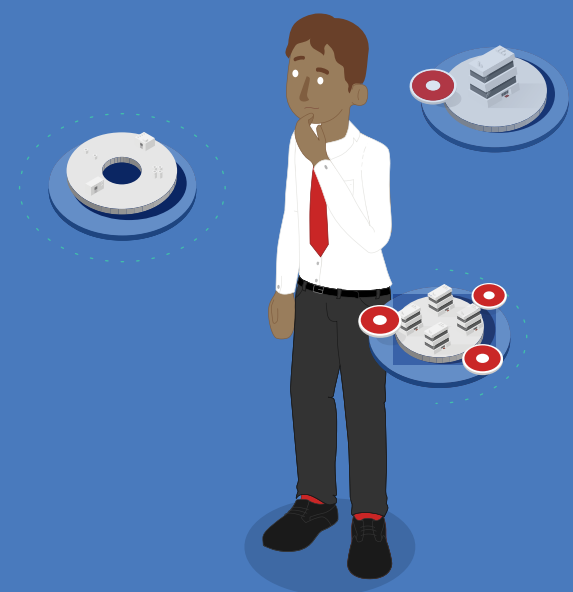
During the tech interview, test all the skills mentioned in the job description by designing an exercise that will be difficult enough to show the needed proficiency and at the same time simple enough for candidate to be able to do it with pen and paper in a reasonable time frame (90 minutes max). The intention should be to evaluate the approach to the problems, not necessarily the accuracy of the syntax. The clarity of approach and the problem-solving logic should be the main premise. The exercise should assess that the candidate has:

- **Mastery** of good architectural principles and the ability to design applications using 4-layer architecture
- Enough **experience** to guarantee good performance, scalability, and maintainability
- The **knowledge** needed to avoid, detect, and correct circular references
- **Familiarity** with the tools used to monitor project architecture

For a senior developer, the exercise should also test competencies related to web architecture and solution design. For the tech lead, you should design an exercise that assesses team and client management experience. An example of a tech exercise for a more junior position is in [Appendix C](#).



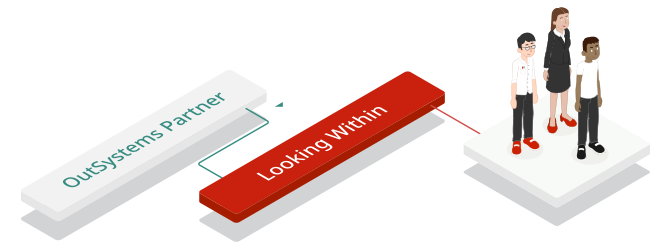
Mix and Match



Now you're probably wondering whether you should look within, go to the market, or bring in a partner. So, here's some good news: you don't have to choose just one approach.

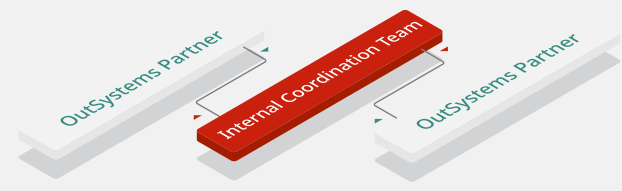
You can look both inside and outside of your company, or do one and have a partner help where needed. What we've observed from our work with many clients is that they mix and match, using some internal resources and filling gaps by bringing in external resources.

Here are some models of how mixing and matching can work.



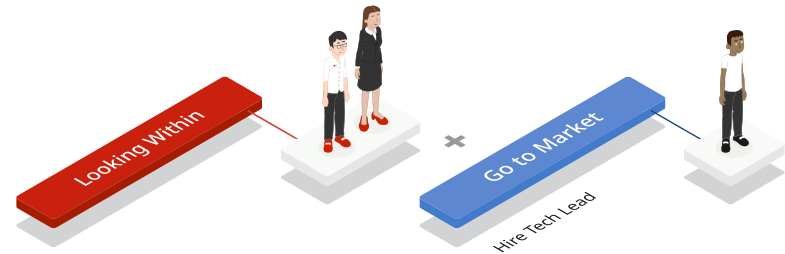
Model 1

The most common model is bringing in a partner for the first project, shadowed by your internal teams so they can learn and lead in the next applications.



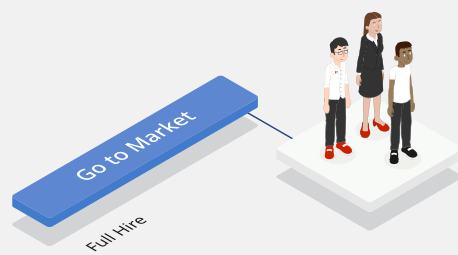
Model 2

Using an internal team to coordinate project delivery from different partners.



Model 3

Building an internal team, and hiring a senior developer as a tech lead.



Model 4

Going to the market for hiring a whole OutSystems team while having a partner for the first project who also enables the new team.

The Team's All Here:

Ramping Up

Assembling your team is just the beginning of a beautiful OutSystems project or digital low-code factory framework.

This section offers guidance on training developers, onboarding, supporting your team and ecosystem, and career paths.

(If you are looking for information about how to onboard or ramp up your entire organization, check out the [Digital Transformation Foundation Playbook](#) for all the details.)

Training and Onboarding Developers

OutSystems offers training for developers in the form of courses and docs that teach them how to build high-quality mobile and web apps with OutSystems. Guided paths are provided for various OutSystems profiles, from the very beginners to those with extensive expertise who want to take their talents to the next level. Check which learning path is right for your team on our [Training page](#).

[Certified](#) OutSystems community members can evolve through the different certification levels to master the OutSystems technology.

In addition to the guided learning paths, we provide snippets of advanced training you can use to enable your internal team as they work on their project. Depending on the scope of your first project, whether you opt for web or mobile app development, and taking into consideration your team's set of skills, we can provide **laser-focused portions** of training through our new Training planner tool. For detailed information on this new tool please check Talent Playbook 2 - Ramping up!

Resources for Developers

From day one, your developers will not be alone. A community, resources, and an ecosystem are there to support them.

Community

A worldwide [OutSystems Community](#) provides training, tech components, forums and, of course, an extended well-functioning family where developers can not only learn but also share their new skills. Developers can ask questions, share their experiences, teach, propose new ideas, and progress in their career. They should be encouraged to join and create a **community profile**.

With this profile, developers are recognized (with badges and kudos) and ranked according to their contributions to the community. Their profiles also enable them to tell other OutSystems developers about their interests, what opportunities excite them, and what they can offer. Make sure your developers start climbing the rankings from the very beginning of their OutSystems journey.

Forum

The Forum is a safe space where developers can ask any kind of question in the OutSystems Community when they feel stuck. Even the most complex questions are resolved in a single day, often in a matter of hours (depending on complexity, some answers can have a viable solution in 1 hour or less). To get a picture of how well the [Forum](#) works, some of our clients decided that one of their OutSystems developers should spend a couple of hours per week solely dedicated to it. They read, reply, and provide solutions or they look for answers. This gives them the opportunity to learn from other people's successes (and mistakes) and get quick solutions to complex problems.

MVPs

Each year, the OutSystems community awards its most helpful and experienced members with **Most Valuable Professional (MVP)** status. MVPs are the technology experts, and they are recognized as such by OutSystems partners and customers. Others view them as elite developers, and their reputation is global. MVPs are active in the community, and they can help new developers with guidance, advice, and information on specific topics. Read more about the **MVP program** [here](#).

Ideas

There is also a collaborative space where community members share and discuss ideas. When they have an idea about how to improve the platform they just reach out to the **Ideas page** and write it down. While not all suggestions become reality, our engineering team considers each proposal, and many of them get implemented. It's a great opportunity for developers to have fun and be bold.

Newsletter and Daily Digest

Subscribing to the [Developer Newsletter](#) is a great way to stay in touch with everything that's happening in the world of OutSystems developers and in the OutSystems Community. Any developer who wants to be on top of the game should subscribe. Additionally, [subscribing to the Daily Digest](#) will give developers updates on daily happenings in the community as well.

Ecosystem

In the market, where changing demographics and skill requirements make top talent increasingly scarce, companies keep looking for innovative ways to create and nurture talent ecosystem. This is important from the standpoint of managing and developing your own talent as well as extending talent pool. Here are two types of events that helped OutSystems community and ecosystem grow.

OutSystems User Groups

Spread all over the world, OutSystems User Groups are events for our customers, partners, and developers that focus on technical content, such as best practices, tips and tricks, live demos, upcoming features or inspirational experiences and testimonials. The events are co-organized by OutSystems and the community, and they are solely focused on sharing knowledge and learning. The ultimate goal is to build, grow, and nurture our local communities. You can learn more [here](#).

Hosting a hackathon

Organizing a coding challenge or hosting a hackathon keeps young professional OutSystems developers vibrant and informed. These activities have a positive impact on business because they consistently generate new ideas and quick proof of concepts, plus they help build a knowledgeable and available talent pool. OutSystems hosts a number of hackathons on our own, but also with our partners, clients and universities. From this experience, we have created a very easy-to-use Hackathon [Starter Kit](#).



Future-Proof: The OutSystems Developer Career Path

A critical advantage of using the OutSystems platform to develop and deliver applications is that its users have the opportunity to grow in their careers, something that a specialization in outdated technology cannot offer. Here are some of the career paths available that start with developing and recognize the low-code pioneers responsible for delivering early wins.



Dev

Dedicated **developers** can build and enhance their careers by evolving and mastering their skills (associate, senior, lead developer).



Architect

Those interested in taking their technical skills to the next level and design layered OutSystems architectures can opt for the **architect** role on the OutSystems team.



Tech Lead

Furthermore, for those who would like to use their OutSystems expertise to help their team find success, but still stay hands-on with the tech part, they can opt for the tech lead path. A **tech lead** is the interface between the development team and the business, and the one who comes up with the big picture of how to solve business problems sustainably.



DevOps

Technical resources with more of an interest in improving CI/CD processes could move into the **DevOps** role, where they will be responsible for monitoring infrastructure as well as applications deployed in production.



Product Owner

Those who would like to become closer to the business can do that by taking on **product owner** responsibilities. A Product Owner has the vision of the final product, and is responsible for sharing that vision with the whole team, building the backlog of features and influencing the project priorities.



UX / UI Designer

The **UX/UI designer** provides a clickable prototype that visually demonstrates the look and feel of the user experience, thereby increasing the understanding of the application's value. Most importantly, this role maps the customer journey.

Conclusion

We hope this playbook has provided you with valuable insights on what to look for when you're assembling your team of developers, how to conduct a search internally or externally, how to ramp up your team once it's built, and how to ensure the success of your team.

These insights and guidance are a result of many years of working with clients and helping them write their OutSystems success stories. So, what will your success story be like? Let's get started and find out.

For more information on the roles for Center of Excellence and further tips on training and development of your team, check out the Talent Playbook, Part 2, which explores these topics further.



Appendix A

Competence Skill Level Breakdown

For each of the low-code roles described in the talent section, there is an assessment of the required competence on a 5-point scale according to the following chart:

Skill Level: 0 <i>Not Applicable</i>
The competency is not applicable to the role.
Skill Level: 1 <i>Basic Knowledge</i>
There's a common knowledge or an understanding of basic techniques and concepts. Focus is on learning.
Skill Level: 2 <i>Limited Experience</i>
Able to understand and discuss terminology, concpets, principles, and issues related to this competency. Has a level of experience gained in a classroom, experimental scenarios, or both—or as a trainee on the job. Is expected to need help with this skill. Focus is on developing through on-the-job experience.
Skill Level: 3 <i>Intermediate</i>
Able to successfully complete tasks in this competency. Has applied this competency in situations successfully and with minimal guidance. Is expected to use the skill independently with occasional helpf from more a senior resource. Focus is on applying and enhancing knowledge or skill.
Skill Level: 4 <i>Master</i>
Able to provide guidance, troubleshoot, and answer questions related to this area of expertise and the field where the skills is used. Has demonstrated consistent excellence in applying this competency across multiple projects and organizations and also created new applications for this competency, lead the development of reference and resource materials for this competency, or both. Is expected to coach others in the application of this competency by translating complex nuances into easy-to-understand terms. Focus is on broad organizational or professional issues.

Competence Breakdown

The following table provides more detail about each of the nine competency areas identified as part of the talent dimension.

Skills	Description
Analysis and Design	
Architecture / Solution Design	Master good architecture principles and be able to design applications using the 4-layer architecture Guarantee good performance, scalability, and maintainability Know how to avoid, detect, and correct circular references Know how to use tools to monitor project architecture Is expected to be considered the “go to” person in this area within and outside the organization. Focus is strategic.
Requirement Gahtering	Master requirements gathering techniques: <ul style="list-style-type: none">Identify vision and driversClearly understand business contextIdentify personasDefine user storiesDevelop business process modelsBuild mockups
Analysis and Design	
Web Application Architecture	Understand web application architecture (client/server architecture): <ul style="list-style-type: none">How and when to POST or GETPage renderingDatabase accessUnderstand session behaviorUnderstand browser cachingView StateAJAX requests
Web Application Architecture	
Understand and Apply Usability	Understand information architecture: <ul style="list-style-type: none">What it isWhy build itOrganizationLabelingCostNavigationUser locationAuxiliary navigation Understand SCRAP (symmetry, contrast, repetition, alignment, proximity). design principles and be able to apply them when building an application Know the 11 UX Rules for IT Developers Understand the importance of usability tests and be able to execute them Use real sample data or generate realistic test data

Skills	Description
Analysis and Design	
SQL Programming	<div><div>Able to implement:<ul style="list-style-type: none">Common table joins (left join, inner join)ExistsCountGroup byTopRow numberAggregate functions (SUM, AVG, MAX, MIN)</div><div>Able to implement:<ul style="list-style-type: none">InBetweenOuter joinSQL CaseSubqueries</div><div>Understand indexes and their impact on performance</div></div>
Troubleshooting and Debug	
App Troubleshooting	<div><div>Able to analyze error logs</div><div>Able to analyze monitoring logs</div><div>Able to analyze performance logs</div><div>Know how to troubleshoot, identify, and fix performance issues</div><div>Understand indexes and execution plans</div></div>
Development Languages	
Algorithmn and Pseudocode	<div><div>Understand the problem or requirement</div><div>Think through the solution and develop a design - be able to drill-down and explain it</div></div>
C#, Java, Python, PHP	<div><div>Understand language best-practices and most commonly used code libraries</div><div>Able to implement a basic algorithm or functionality</div><div>Experienced with MS Visual Studio, Eclipse, or other IDE</div></div>
IOS & Android	<div><div>Swift and Java</div></div>
UI Development Languages	
CSS	<div><div>Know CSS syntax</div><div>Able to explain the box model</div><div>Know what a pseudo element is</div><div>Able to troubleshoot an issue in the browser</div><div>Know the order of import of CSS in OutSystems</div><div>Know how to read and apply selectors</div><div>Know the differences between CSS and CSS3</div><div>Know the difference between positioning values</div><div>Able to understand the organization of a theme</div><div>Able to understand vertical alignment</div><div>Able to explain RESS (responsive web design + server side) and RWD (responsive web design)</div></div>

Skills	Description
UI Development Languages	
HTML	<div><div>Know what elements are used and how by</div><div>Know what APIs were mande available in HTML5</div><div>Know how to add HTML standard elements to an HTML file</div><div>Understand accessibility</div></div>
JavaScript	<div><div>Understand JS/jQuery</div><div>Select an element by id</div><div>Select an element by class</div><div>Able to show and hide an element</div><div>Able to import (and use) a jQuery plugin</div><div>Able to troubleshoot an issue in the browser</div><div>Know how to create a JS object</div><div>Know how to create a jQuery plugin</div><div>Know how to do a manual jQuery Ajax request</div><div>Has created and shared a jQuery plugin in the community</div></div>
Integrations	
REST & SOAP	<div><div>Able to consume a REST and SOAP API</div><div>Able to expose a REST and SOAP API</div><div>Know how to test REST and SOAP integrations</div><div>Know how to handle null values for SOAP API</div></div>
Third-party Integration	<div><div>Know how to create extensions using the SAP Wizard for Integration Studio</div><div>Know how to handle transactions between OutSystems and SAP</div></div>
Performance	
Assess Application Performance	<div><div>Able to drill down into reports and identify bottlenecks .</div><div>Use performance reports (understand difference between client-side vs server-side time)</div><div>Use the browser performance tools</div></div>
Optimize Application Code	<div><div>Reduce the page size</div><div>Reduce the size of page resources to load and cache them</div><div>Reduce the data fetched from the database</div><div>Cache data from external data sources</div></div>

Appendix B

Job Description Templates

These templates are based on the skills and experience OutSystems has been using to select the best candidates for two core positions: developer and tech lead.

Developer: Desired Skills and Experience

- Strong experience in developing web and mobile applications
- Proficiency in web-based languages: JavaScript, CSS.
- Knowledge of C# and SQL languages are a plus.
- Minimum of 2 years of professional experience in software development.
- Is flexible and responsive to changing priorities and multiple, concurrent responsibilities.
- Drive to collaborate with others and a growth mindset.
- Key individual contributor to successful delivery on a large product or project with applied knowledge and participation in full lifecycle development.
- Experience with building proofs of concept is a plus.
- Degree in Computer Science or Engineering and academic achievements valued.
- OutSystems certification is a plus.

Tech Lead: Desired Skills and Experience

- Knowledge of software engineering principles, algorithms, data structures, and object-oriented design.
- Able to handle an even split between architecture and design responsibilities (50%) and programming responsibilities (50%).
- Proven ability to design and optimize application and data integration processes across systems.
- Collaborate on team growth through peer coding, code review, and knowledge transfer.
- As a technical leader or contributor (or both), solve problems of considerable scope and complexity and mentor less senior developers with their work.
- Experience with leading other members of the development team in design, development, implementation, peer code reviews, and technical meetings.
- Drive the continuous evolution of best practices in the OutSystems development team.

Note: These are the basic skills, and the proficiency you are looking for will vary based on the seniority of the profile needed for your team. For example, a junior developer should be self-motivated, have a learning mindset, and know the basics in JavaScript, CSS, HTML, and SQL. A senior developer, on the other hand, should have experience in web application architecture and understand the logic for a 4-layer architecture.

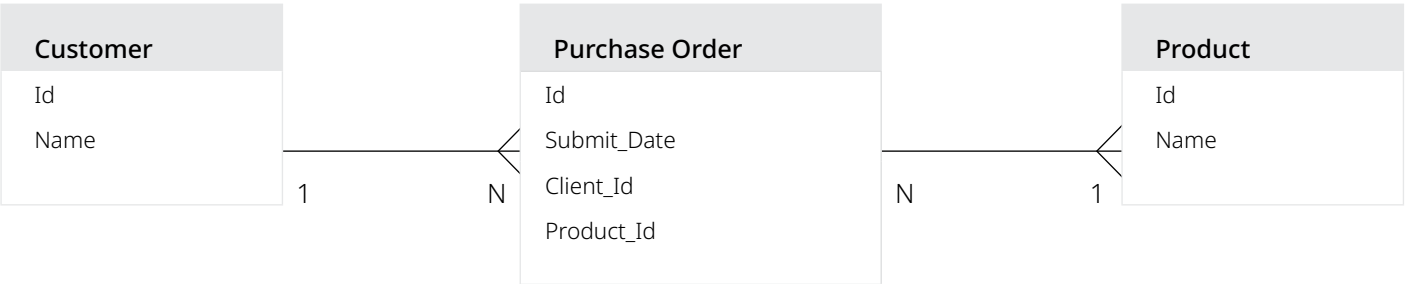
Appendix C

Tech Exercise Example

Programming Test

The following exercises are the ones we use to evaluate a candidate for a junior position, which includes evaluation of the capacity for database modeling, SQL querying, and algorithms. If the position you want to fill is more senior, naturally you should adapt the level of complexity, the exercises, and the tech questions to that position.

1. SQL



Assume the following model with the following data:

Customer		Purchase Order				Product	
Id	Name	Id	Submit_Date	Customer_Id	Product_Id	Id	Name
1	Mary	1	2019-02-04	1	1	1	Laptop
2	Noah	2	2019-02-05	1	2	2	Web Cam
3	Anna	3	2019-02-06	2	2	3	Keyboard
4	Elizabeth	4	2019-02-07	3	1		
		5	2019-02-13	1	2		

1.1 How do you obtain the list of client IDs that purchased product 2 (Web Cam) in SQL?

Expected result: ID (1,2)

1.2 How do obtain the list of client IDs that did not purchase product 2 (Web Cam) in SQL?

Expected result: ID (3,4)

1.3 How do you obtain a report with the number of purchase orders for each client ordered with the best clients first in SQL?

Expected result: Mary, 3; Noah, 1; Anna, 1;

1.4 Adapt the database model to support multiple products for each order.

2. Algorithmics - Part 1

Write a function to merge two strings in the reversed order.

```
InvertedMerge( "abcdefg", "12345" ) == "g5f4e3d2c1ba"

InvertedMerge( "x", "12345" ) == "x54321"

InvertedMerge( "ABCDEF", "z" ) == "FzEDCBA"
```

3. Algorithmics - Part 2

[University of California] The Soundex Index System was developed so that similar sounding names or names with similar spelling could be encoded for easy retrieval. It has been used by the U.S. Bureau of the Census, and some states use it to help encode driver’s license numbers. Your task is to convert a sequence of names into the corresponding Soundex codes.

A Soundex code always consists of a letter followed by three digits, whereby:

- 1. The first letter of a name appears (unencoded) as the first character of the Soundex code and is capitalized. It is also the only letter.
- 2. The letters A, E I, O, U, Y, W, and H are never encoded when they are not the first character in a word. They do serve, however, to break sequences of like-coded letters (see next rule).
- 3. All other letters are encoded according to the following table, except when they immediately follow a letter (including the first

Code	Key Letters
1	B, P, F, V
2	C, S, K, G, J, Q, X, Z
3	D, T
4	L
5	M, N
6	R

letter) that would be encoded with the same code digit:

- 4. Codes are truncated after the third digit.
- 5. Trailing zeros are appended as needed so all names are encoded with a letter followed by three digits.

The input contains a sequence of words in free format and ends at end of the file. The output written to the file should consist of a column of names and a column of their corresponding Soundex codes in the format shown in the example.

Appendix D

Responses from 2019 Developer Survey

In January 2019, we surveyed developers who use OutSystems. Respondents ranged from beginners to experts in OutSystems and from across the globe. The following responses are pulled from an open response question “What do you like most about the OutSystems platform?” The top areas mentioned were speed and ease of use.

“The ability to deliver fast and efficient deliveries, and exceed customer expectations; The ability to make one of the agile fundamentals, “Responding to change”, with no pain; Time to market very fast.”

“Web and especially mobile development have been made so simple and easy. So much of the process that a developer would normally have to be concerned about has been made an afterthought.”

“Once you understand the basics, it’s just easy to use. There are also many examples of solutions that the community posts and allows me to see new areas of development that I could pursue.”

“While it’s an easy platform to start with and fast to develop an application, it’s not a toy for only unexpert developers, but a professional platform that can be used by all parties involved in a development project, from business staff to technical software engineers”

“The community support is what impressed me the most, whenever I’d have any doubts I’d reach out to my teammates and whenever they’d fail to help me, I would air out my queries to the community and get a response almost immediately. The support and service provided to the customers shows how much you guys care about us. Mighty impressed with the support provided by the community, easily the thing to vouch for.”

“That is very developer-oriented. I mean the community. The company implements the stuff that we suggest, and it is great! Being part of a community you know that you have some value, and that you are welcomed. And that is the best thing that the company can offer.”

“It allows me to focus on solving client’s issues instead of solving the technology puzzle that comes with any web or mobile application.”

“The ability to concentrate on Business Logic and functionality rather than solving technology issues. The platform takes away a lot of technical implementation issues.”

“Especially the scaffolding functionality is very helpful and saves a lot of time in coding. It also helps to understand the platform by checking the generated code.”

Developer Testimonials

When we directly asked our MVPs what they enjoy most about OutSystems, the following responses were received.

Craig St Jean - MVP @ Netlink Software Group

“OutSystems has allowed me to deliver more in less time, while increasing application maintainability at the same time. In addition to the huge productivity we have gained, OutSystems has advanced our security by at least 5 years.

My experience with OutSystems led to me being hired into my current position. I love developing applications, and now I can spend my time focusing on the complex components of development, instead of the more tedious and time consuming tasks involved in web and mobile development.”

Jordan Welch - MVP @ Highland Solutions

“With OutSystems I get to spend my time solving interesting problems and doing the fun parts of developing. I get to skip over a lot of the boring parts that the system handles for me.”

I learned OutSystems in whatever time I could spare at my IT Support job, building an internal tool for our company. A year later I was recruited to a full time development job working with the platform. I never thought that it would take me where I am now and I LOVE my job.

I often get to explain to my managers that ‘yes, I AM sure that it will deploy to production error free.’ or ‘don’t split that story up, half of it will be accelerated by the platform, it is almost no work for me.’ They are blown away every time.”

Eduardo Jauch - MVP @ IT Up

“I program since computers had only 640KB of memory. I love to create solutions. I learned to create hexadecimal code, assembler, basic, pascal, C, C++, Clipper, VB, etc. I always look to spend less time typing, looking for bugs, etc. Every new language I learned improved this one way or another. Every time more high level. One day I’ll become obsolete. And it’s OK.”

Pramod Jain - MVP @ Netlink

“I spend so much time making my developers understand what OutSystems can do and it still required logics and technology experience .When I started this practice it was a big challenge to ask developers to start Outsystems and some of them left the company just because of this. But now developers are coming to us asking getting them trained in Outsystems and giving them the opportunity to work with it, and the great thing it’s the customers who want to use this platform to reach out to market earlier than anyone else can.”

Joao Melo - MVP - Independent Consultant

“In my case, I developed in different languages, but I’ve never been able to get the results I got since I chose OS. But again, it works with people who are seeking business results, which is not that easy to find.”



www.outsystems.com

© Copyright OutSystems 2019. All rights reserved. OutSystems and the OutSystems logo are registered trademarks of OutSystems. All other trademarks are the property of their respective companies. Unauthorized copying or distributing is a violation of copyright law.