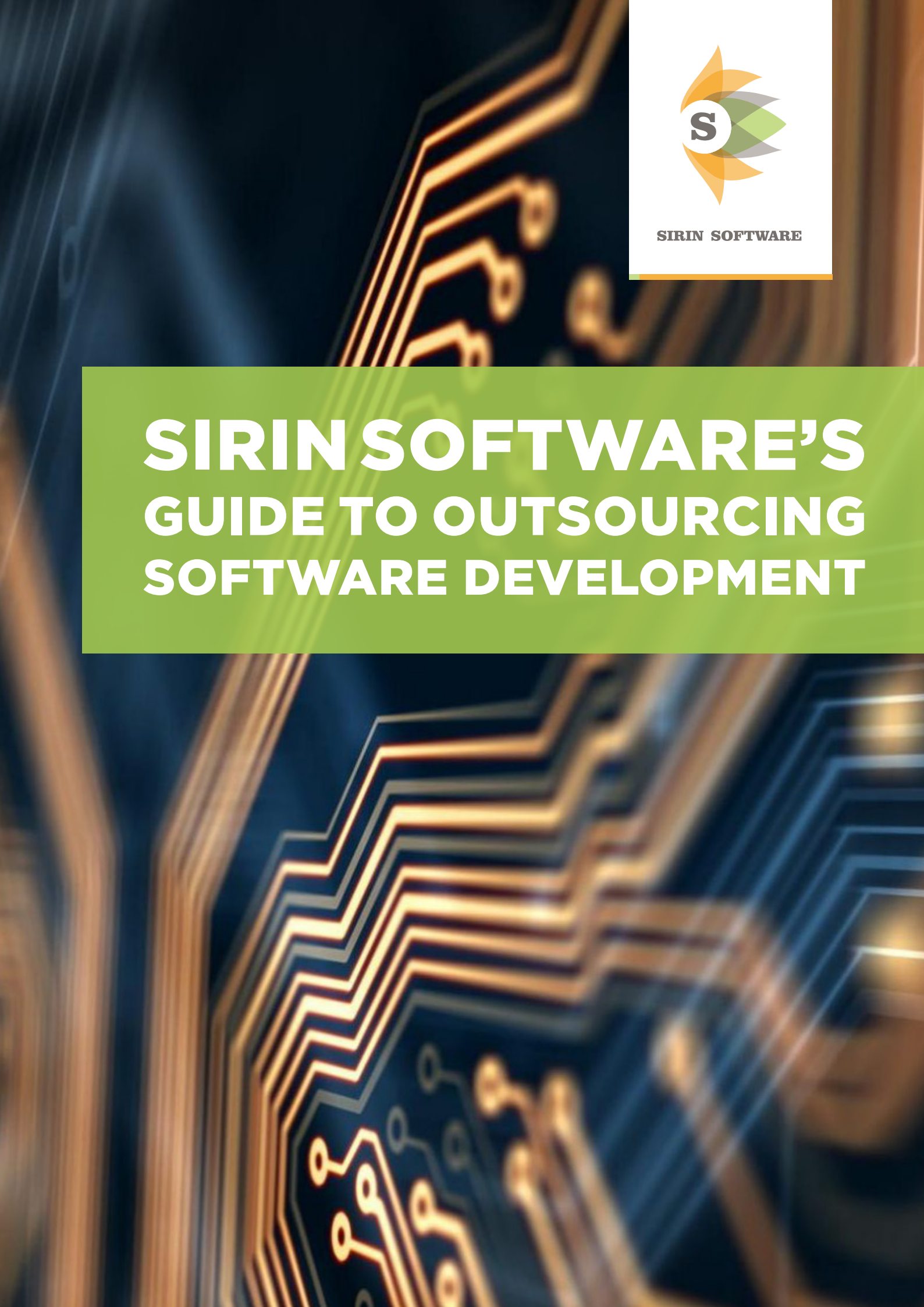
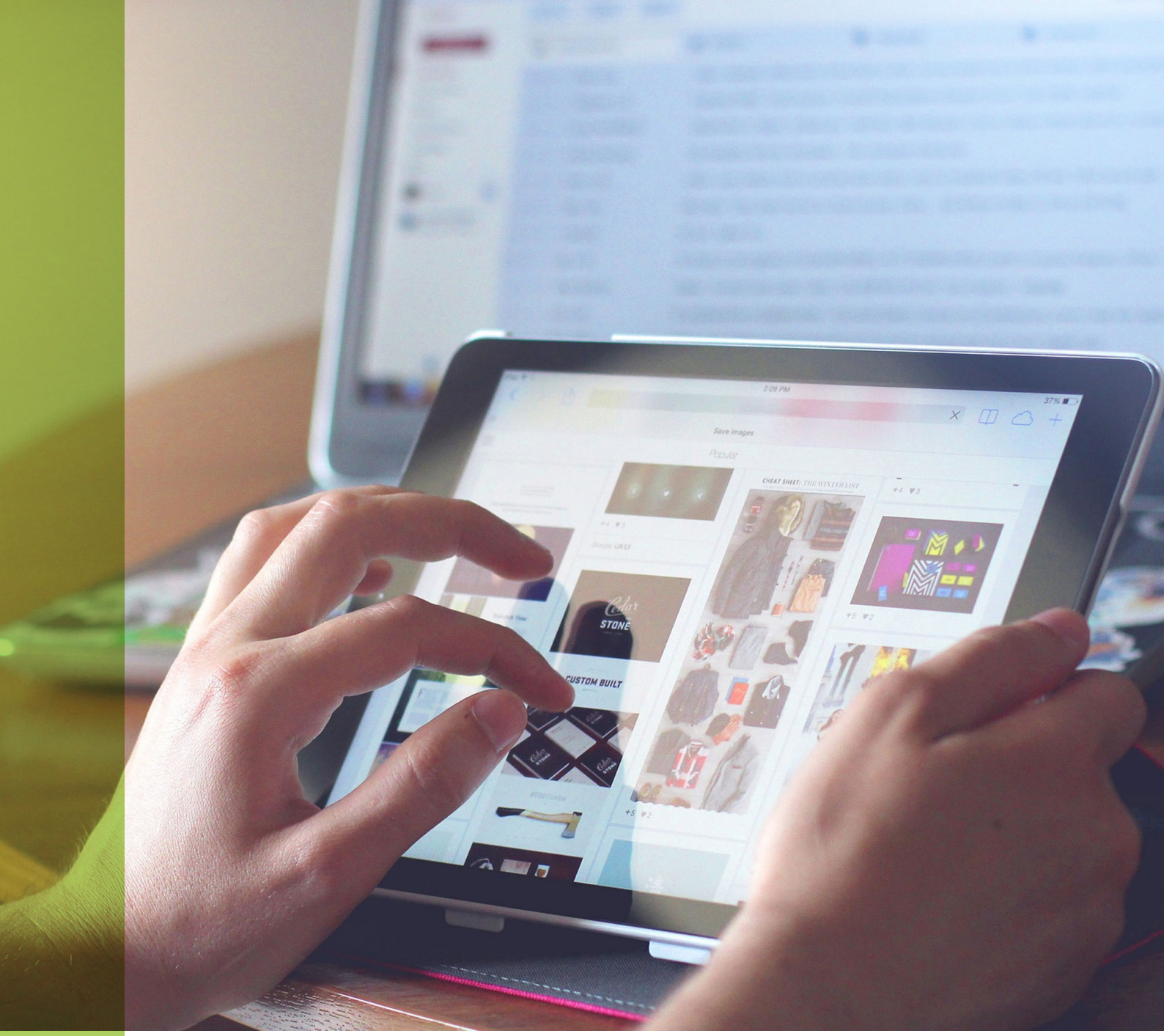


SIRIN SOFTWARE

SIRIN SOFTWARE'S GUIDE TO OUTSOURCING SOFTWARE DEVELOPMENT



BEFORE YOU ADDRESS A SOFTWARE DEVELOPMENT COMPANY, YOU SHOULD:

1 **Define your goals** – that is, the business outcomes you want to achieve by outsourcing software development. These may be cost reduction, workflow optimization, team expansion or filling the expertise gap;

2 **Set priorities.** When it comes to choosing a software vendor, what matters most to you? It might be experience, reasonable developer hourly rates, expertise, scalability, etc.;

3 **Choose a cooperation model that best meets your needs.** There are two ways you can cooperate with your remote development team. The first one is **Dedicated Team (DDT)** model which enables customers to hire developers full time, appoint their own Project Manager (PM) to manage the remote team directly, implement certain project management tools that they use in-house, meet or e-meet their remote employees on a regular basis, etc. With DDT, you basically get an extension of your in-house team. There's one more working model called Managed Resources & Development (R&D) which is gaining popularity with both outsourcing companies and their clients. According to the model, an outsourcing company chooses a skilled PM from their in-house team. The manager converts a customer's needs into technology requirements and is responsible for providing deliveries and facilitating communication between the client and his remote developers. With Managed R&D, you can focus on business processes from day one and trust the tech part and communication to those who know it best;

4 **Prepare a brief or detailed specification of a software solution.** There are several options here. In case you have an in-house IT department and employ skilled Business Analysts (BAs), you can write a proper SRS – that is, a document which outlines functional and non-functional requirements of a software solution and features wireframes/mock-ups and user stories. If you do not possess in-house resources to complete the task or have only a vague idea of what an application is supposed to do, you can describe it from the perspective of a businessperson and bring examples of similar solutions implemented by your competition;

5 **Decide on the deadline and available budget** (no explanation is necessary here, right?)

4 TIPS TO CHOOSE THE RIGHT SOFTWARE DEVELOPMENT COMPANY

The best way to choose a software vendor is to study independent software development companies' ratings on platforms like Clutch.co, AppFutura, GoodFirms, etc. On these websites, you can filter companies by country, size and expertise, compare their ratings and read customer reviews.

- **Define a vendor's level of competence.** A company's profile should correspond to at least one of these requirements: 1) the experience of working with businesses from your industry; 2) similar projects in their portfolio; 3) relevant expertise (provided you know which technology stack your software solution is going to use);

- **Hire a tech consultant.** Unless you have a CTO, you should find a software architect/tech lead to help you navigate the complex requirements elicitation process. Yes, consultants do not come cheap – but it's the only way to avoid investing in a product that doesn't meet your requirements;

- **Check the vendors' websites, testimonials and social media presence.** Although outsourcing companies are pretty selective about reviews and testimonials they publish on Upwork, GoodFirms and Clutch, those reviews are written by real customers (the platforms' editors do see to it). Also, make sure the companies you'll contact have professional-looking websites with no user experience (UX) issues and are active on social media – after all, you don't want to have your software built by a team of no-name freelancers under the guise of a valid contract. Also, you can directly contact businesses that previously worked with the vendors and ask for references.

- **Contact at least 3-4 outsourcing companies and compare their estimates/business proposals.** If it's your first time outsourcing, you should send your brief over to up to 10 vendors; in reply to your email, the vendors should provide you with a ballpark estimate of the software solution featuring the duration of your project, technology stack choice and a tentative budget. At this stage, you should pay attention to the vendors' response time (a Business Development Manager should respond to your initial inquiry within 24 business hours), the accuracy of English (in case you consider outsourcing to a non-English speaking country) and the estimate itself (was it really tailor-made for your product?). Mind that a ballpark estimate is a preliminary one; in order to evaluate developer efforts and prepare a detailed technical vision of the project, you'll have to conduct several phone or video calls with the companies' sales managers and tech leads. Their follow-up questions will be another indicator of the vendors' competence (or lack thereof). And yes, you should sign a non-disclosure agreement (NDA) before laying out an innovative software idea for a third-party vendor. We recommend that you draw a chart and evaluate the vendors according to the criteria mentioned above;

	Company 1	Company 1	Company 1
Experience	10	8	12
Similar cases	+	+	-
Expertise	+	+	+
Testimonials	+	+/-	+
English level	Advanced	Upper-Intermediate	Upper-Intermediate
NDA	+	+	+

GETTING STARTED SO, YOU HAVE CHOSEN A VENDOR AND CLARIFIED SOFTWARE REQUIREMENTS. WHAT'S NEXT?

THERE ARE SEVERAL THINGS TO CONSIDER BEFORE SIGNING A CONTRACT. THESE INCLUDE:

- **Intellectual Property (IP) rights.** Make sure to specify that you will be the sole owner of the end product – otherwise a contractor might simply steal your idea;

- **Conflict of interests.** If you are a contractor yourself, make sure to legally forbid the vendor to contact your customer directly;

- **The right to hire.** In case the outsourcing company expands their expertise up to the level exceeding the expertise of the developers you've employed, you are free to hire new specialists to level up your team;

- **Delivery acceptance.** When the project is completed, your vendor is bound to send you all source files including Bundle Soft, tech documents and source code.

WHY IS IT IMPORTANT?

One of our clients, for instance, simply wanted to upgrade a business app which had been developed by another vendor. Obviously, our software engineers requested source files – and there were hardly any! We tried to reach out to the original developer only to find out the company had gone out of business. The app in question was a complex one, and there was no way Sirin Software could customize it without the source files. As a result, we had to write the software from scratch – and that's hardly a minor upgrade our customer had counted on.

ALSO, THERE ARE DECISIONS YOU HAVE TO MAKE TO ORGANIZE THE OUTSOURCING PROCESS:

- **Choose the right pricing model.** As a rule, outsourcing companies bill their efforts according to three pricing models: **Fixed Price** (FP), **Time & Material** (T&M) and **Dedicated Team** (DDT). These models use different project management methodologies: **Waterfall** (or traditional approach to software development), which is applied to FP projects and **Agile** (a customer pays for the actual dev hours spent on his project and has a lot more freedom in prioritizing tasks/features on the scope and managing remote teams), upon which the DDT and T&M models are based. You have an SRS and know exactly what you app is supposed to do, both business- and technology-wise? Go Waterfall then! In case you're outsourcing innovative software development (IoT, AI) or aren't sure about the software's final feature set, it's better to choose Agile and be more hands-on with the project;

- **Determine the appropriate communication channels.** If you're reading this, you probably consider outsourcing software development to a nearshore (located in a neighboring country) or offshore (located on another continent) company. This basically means that you won't be able to meet your team in person frequently. With Skype, phone calls, messengers and emails, that shouldn't be problem. However, you need to specify the preferable means of communication and reporting periods in advance;

Even if you partner with an experienced vendor, complications might still arise – and it's a positive sign if your contractor tells you about such complications beforehand. After all, software development – unless we're talking about trivial projects, of course – does not always go by the book.

FOREWARNED IS FOREARMED, RIGHT?

And yes, it's totally fine to start with a milestone-based contract which is designed to produce a single deliverable like a certain feature of a product, detailed technical specification or high-level understanding of software architecture. In IT, a typical milestone lasts for three to six weeks. During the period you'll be able to evaluate the vendor's tech expertise, decide on the optimum cooperation model and optimize your communication (after all, the non-obvious things like time zone differences, cultural gap and communication issues might undermine the success of your outsourcing project).

> By the way, the willingness to sign a milestone-based contract speaks of a vendor's reliability and dedication to customers. On the contrary, companies that insist on signing a complete lifecycle contract or making prepayments most often simply seek ways to buffer against possible risks and do not really care about outcomes.

5 PITFALLS TO AVOID WHEN OUTSOURCING SOFTWARE DEVELOPMENT

- **Failure to define the role of a vendor in software development.** Before you sign a contract, you should clearly understand the role and responsibilities of your outsourcing provider. A nearshore/offshore company may provide an extension of your in-house team, simply performing the tasks you assign to them. A remote team may also be a fully-fledged participant of the software development process and, having assumed the role of a product owner, contribute to your company's growth. Your failure or reluctance to define the parties' roles might cause misunderstandings (including arguments over IP rights) in the future;

- **The wrong choice of a project management methodology.** If your business largely implements one of the Agile project management methodologies (Scrum, Kanban, etc.) and your vendor is a Waterfall-only company, you'll have little control over product development and project outcomes. Similarly, if you have a well-written SRS and defined budget, you don't need that "agility";

- **Communication issues.** These include limited English proficiency, time zone differences and cultural gap. While you can easily evaluate your vendor's English language skills and manage time zone differences, it's cross-cultural communication issues that undermine the success of 28% of all IT outsourcing projects. How could you possibly choose the right outsourcing destination then? Geert Hofstede, a notable Dutch psychologist and former IBM employee, developed a comprehensive

cultural dimensions theory according to which employees from countries with high Power Distance Index (it's mostly Asian and African countries) make poor partners to US and Western European businessmen, that's why you probably should pay a closer attention to outsourcing companies located in Eastern and Central Europe;

- **HR attrition.** A skilled development team is what makes or breaks IT projects. Outsourcing companies with high turnover rates aren't likely to maintain the same level of service throughout the project (particularly a long one). In order to choose a company with loyal staff, you should visit local websites like Glassdoor, XING and DOU and check reviews written by your vendor's former employees;

- **Lack of motivation on the part of your remote employees.** Vendors who don't invest in employee education and recreational activities often struggle to keep developers motivated. What does it have to do with your project? First and foremost, the lack of motivation might result in employee turnover and affect product delivery dates, as well as the overall code quality. Second, motivated teams are generally more productive, make less coding mistakes and gain a better understanding of your product. That's why you should clarify motivation issues in advance and suggest improvements if necessary.