

9 questions you must ask yourself before starting development of a startup

This document is intended for non-technical founders who are about to start a technical implementation, but do not fully understand how to start it. Most likely you have questions that you have not yet found answers to- I hope you will find all answers in this article. Besides, in this article you would surely come across the questions that you have not even thought about, and it will be useful for you to know both the questions and the answers to them. If after reading this article you still have any questions, write to me, and I will willingly answer them.

How do we choose a technology stack?

In fact, for you as a non-technical person, the technology stack in itself does not play a crucial role. You need to proceed not from the technology stack, but from the person whom you involve in the development. There are not so many smart specialists on the market who are open to offers and are able to implement a project from scratch with a proper quality. If you manage to find the right person, and it turns out that he does not use the technologies that you were counting on, this is not a reason to refuse his services.

Nevertheless, I will give some recommendations regarding the technology stack. The technology stack should not be too exotic and hard to get, so that if you stop working with a certain developer, you can find other developers to continue the project. To determine if a technology stack is not too rare on the market, check the number of jobs that mention these technologies on the job search Websites.

If you plan on creating both a web service and a mobile app, which is a common practice, I recommend looking for a developer who uses React and ReactNative. The fact is that these technologies enable to first write the code for a web service, and then reuse most of the code by development of a mobile application. And with all that, the code for the mobile application will immediately work on both iOS and Android. If you choose other technologies, then you will have to hire a separate developer to implement the web service, then another one for the iOS application, and then one more to implement the Android application.

If you want to involve in development as few developers as possible, it would be reasonable to search for a full-stack developer (a developer who can independently make both the server and client parts of the future service). I will provide a detailed answer on whether to choose a full-stack developer or a team of diverse developers in the next paragraph, and in terms of implemented technologies I would recommend looking for a full-stack developer who uses a Node.js due to the fact that Node.js is a server-side programming language that turns JavaScript into the machine code. At the same time, JavaScript is also used by creation of the client part of the service (the external part of the web service or a mobile

application, if it is written on ReactNative). That is why there are more full-stack developers who use the Node.js as a server-side programming language than those who don't, because they only need to know JavaScript in order to write all parts of the service. And accordingly, in the future it will be easier for you to attract additional full-stack developers to the team, or in the event of a negative development of events- replace one full-stack developer against another.

Should you hire a full-stack developer or a front-end and a back-end engineers in separate?

I recommend you to involve a full-stack developer in the development (a developer who can independently provide both the server and the client part of the future service). In this case you will only need one person to start working on the project. Subsequently, if necessary, you can hire either more full-stack developers or a separate back-end and front-end in your team. At the same time, you will have a team member who fully comprehends all parts of the code and can coordinate the work of the rest team members from the very start.

In case if you hire a separate front-end and back-end engineers at the start of works you would encounter the following problems. You will not have a certain person, responsible for everything. You will personally have to coordinate these team members. The backend of the project is written on average 2 times faster than its frontend. Hence, at some point of time the back-end engineer will simply have nothing to do, while you can't put him in charge with the front-end engineers as that is not his area of responsibility. You will either have to hire another front-end engineer to balance the team, or temporarily suspend the back-end engineer from work, taking into account the risk that he may not return to the project later.

Should you prefer Agile or Waterfall?

The Waterfall-based approach assumes that you submit for development a fully compiled technical specification and design of a future web service or mobile application from the outset, and the developer, in his turn, estimates the full cost of development and the required time frame. This approach does not imply the demonstration of intermediate results of the work, and you simply wait for the deadline to see the fully finished product. Besides, the developer puts the risks into the time frame and the cost of development in order to avoid running at a deficit in case of any negative scenarios. Should such scenarios bypass you, then you overpay for development. Moreover, if you choose an unqualified expert for implementation of your project, you will find that out by the quality of work only at the time of the project's completion, when it will be too late to introduce corrections. I would say that this approach does not match for startups.

The Agile-based approach implies an iterative development. You pay the developers a fixed amount every month. The developers demonstrate intermediate results of work each sprint (usually once in one or two weeks). You can show these intermediate results to your future users or potential investors and receive the feedback already in course of development. Alternatively, this approach allows you to quickly bring the product to the market and continue to increase the functionality after the launch of the project. Therefore I recommend using the Agile-based approach by creation of startups.

How much should one pay the developer?

Your team must include at least one developer who assumes the role of a senior developer, and in case if there are more than one developer, he must also act as a team lead. Accordingly, such developer must be paid in accordance with the current market standards. If you offer him a lesser remuneration and he accepts it- most likely he does not meet the declared level of proficiency, because otherwise he would simply find a job with a higher salary. The rest developers should at least possess the middle level of proficiency as the junior level developers are not suitable for a startup, because at this point it is important for you to launch a product with a reliable team rather than enhancing their knowledge. The middle level developers must also be paid in accordance with the current market standards. You can find out the current salaries at the job search websites. You can only pay less than an average market payment if you incharge the person with a task beyond his capacity hoping that he would manage it in the course of work. As you understand, such result may never be guaranteed.

Is it worth hiring a developer for a share in the project?

It is worth hiring a developer for a share in the project only in two cases: either you do not have enough money for development, or your project is technocentric (for example, you are creating a new operating system).

The person whom you give a share in the project must be tested by you in real life, you must trust him by 100% and be ready to follow him to the ends of the world with a desire to complete any kind of projects with this particular man. If you just need to develop a web service or a mobile application, I recommend that you simply hire a person to do the job. If you are not confident in the developer and hire him for a share in the project, you run the risk of tying yourself up forever to someone who turns out to be useless.

You can offer a hired employee an option in your company provided to fulfilment of certain KPI-conditions, but this will rather become your advantage when hunting for this employee and will not evade the necessity to pay him a salary.

How does one verify the proficiency of a developer?

As a non-technical founder, you cannot check the quality of the code. But you can interact with the interface of projects, previously developed by this developer. Check out whether everything works fine and nothing is screws up. Do not pay attention to the beauty of the interface and the presence of illustrations, as these things do not depend on the developer. Make sure that the project which performance you are testing was actually created by this developer, as he could only be a part of the team and simply accomplished the tiny elements of it. Contact some his former customers and ask for the feedback about this developer. You can take the contacts of the previous customer in the section with contacts of the project which operability you verify.

What do you need to provide to developers before work starts?

In an ideal scenario you should have a clearly defined technical assignment and a fully drawn design for all screens of a future service or mobile application. At the same time, you, as a non-technical founder, will not be able to draw up a sufficiently detailed and thoughtful technical assignment on your own, most likely you will miss something, and make logical errors somewhere and the developers will be able to refer to them in case of a low-quality implementation. It would be preferable to form technical specifications and create a design with the participation of a developer, who will then deal with its technical implementation. If you manage to find a developer who agrees to participate in the process of writing a technical assignment and creating a design, in principle, it will be enough for him to superficially describe the idea of your startup and you will think over the rest together with him.

There also exists an approach which does not require a technical assignment, where design is enough, and it is believed that the developer should understand it all from the design itself, and what is not clear, he will simply interrogate during the development.

In principle, the development can be launched having only a graphical prototype (an outline) of the future service, but, of course, the developer must agree to this. In parallel, you can give the prototype to the work of the designer, and when the design is ready, transfer it to the developer so that he can integrate it into the development already under processing.

How can one control the developers?

I recommend establishing such agreements with the developers, in which they demonstrate intermediate results of work once in a week or two weeks. Immediately after gaining access to the intermediate result of the work, you should check the product for operability and report an incorrect behaviour in case if the developers did not understand you and did something wrong or the code simply does not work. Do not leave the review until the last minute, because it is much easier to fix errors at the moment they appear. You can also involve the QA specialists to perform the verification, however I believe that would be redundant in case of a startup.

I also recommend that you single out a team lead among the developers, who would be responsible for entire code, in order to communicate with him only. This developer would subsequently coordinate the work of entire team.

In general, you'd better choose the developers whom you can trust from the very start, because you cannot check the quality of the code as such, and you just need to choose those people on whom you can rely.

How will the process of technical implementation of the project be realised?

It is important to ask this question to the future contractor before work starts. All developers have their own approach to development. I describe my approach to development of

startups at the example of cooperation with the startup “SureTask” in this article:

<https://10.com.ru/case-suretask.pdf>

To get answers to your questions, write to me by the contacts below or choose a convenient time for you to call in the Calendly service: <https://calendly.com/iserdmi/30min> During the conversation, we will discuss your project, I will answer all your questions, and at the end of the call I will offer a specific plan for the technical implementation of your project.

Contacts:

<https://10.com.ru> — on my website you can find out more about me and my team, learn about the benefits of working with us, look through the examples of work, the terms of cooperation, the time frames and the prices for development

iserdmi@gmail.com

Telegram: <https://t.me/iserdmi>

Facebook: <https://www.facebook.com/iserdmi>

LinkedIn: <https://linkedin.com/in/iserdmi>