E-NGINEERS

WORK EXPERIENCE OUTSOURCING AN EMPLOYEE IN SEVERAL TEAMS

www.e-ngineers.com

OUTSIDE THE BOX: HOW ONE MAN’S OUTSOURCING EXPERIENCE CAN MAKE A DIFFERENCE

WORKING AT AN OUTSOURCING COMPANY IS LIKE RIDING A HORSE. IT CAN SEEM UNCOMFORTABLE AT FIRST, BUT IT GIVES YOU A DIFFERENT PERSPECTIVE ON THINGS. IN A WORD, IT’S SOMETHING EVERYONE SHOULD TRY AT LEAST ONCE. THE PROS AND CONS ARE WELL DOCUMENTED BUT ARE LARGELY SUBJECTIVE DEPENDING ON YOUR TYPE OF PERSONALITY AND LIFE STAGE.

The main advantage I’ve found is the opportunity to gain experience with various companies and teams and compare different working cultures. Broadening your knowledge in this way can be really enriching, and working with different clients also forces you to get a feel for different approaches. Sometimes, you will find that simply by preparing yourself for a new interesting client, you will benefit from a deeper understanding of new technology and innovative practices. Working for global companies, you become well versed in both the industry and technology of the customer and you will get exposed to different ways of working.

Since outsourcing companies provide services outside the country, it’s possible some of your working hours will coincide with your client’s schedule; but this is not always the case. This model may influence how global clients schedule their processes, among other factors, and apply different ways of organising their working structure to make distributed teams productive, harmonious units.

Working for some clients allows you to work from home on certain days of the week. Many of them also operate on a flexible time schedule – as long as you complete eight hours of work in a day and are available during core hours, you can start later or end earlier depending on your needs and interests. This also tells us that, without a strong process, you may face some difficulties in delivering high-quality software on time.

Because outsourcing attracts such a range of people and personalities, diversity is something that is encouraged rather than frowned upon. People who choose to work in outsourcing generally find it easier to embrace different styles of working, and the international experience can be a real asset for future projects.

Still, despite the benefits, it can be intimidating to approach new ways of working. My goal here as someone with years of experience in large distributed teams is to pass on some hints and tips that should make the transition easier and help you sit more easily in the saddle. And also pay attention to some details for building international teams that may seem obvious and simple, but in practice turn to be extremely important.

TEAMING – NO NAME BADGES REQUIRED

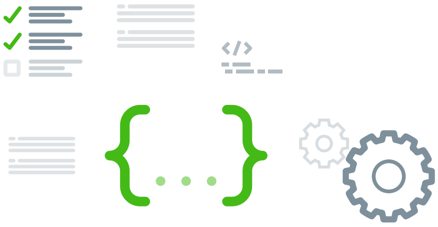
I'm not a fan of forcing everyone in a team to get to know each other personally. Firstly, in-person meetings or even all-team calls can be difficult to arrange and achieve without leaving people out, especially if team members work in different time zones or do flexible hours. Secondly, in the same it's often quite awkward to have to distract your colleagues from work. In addition, in large teams, introducing all participants, many of whom might never contact one another again, does not represent a good use of time.

The best approach, in my opinion, is simply to have a document that contains a list of employees and their areas of responsibility. It's even better if the international team also lists the hours of work of each employee in this document. At the start, and even after several weeks of work, it can be difficult to remember which particular business analyst or designer is responsible for what. The document solves all these problems. Unfortunately, in the past I have often come across a situation where, instead of a direct question to a business analyst, the company's processes required me to ask my immediate bosses a question – and they asked the business analyst! When that happens, resolving the issue can take several days rather than a few hours.

TEAM TALKS

As with many other dealings, it's useful to agree on a preferred way of communicating before starting work. Some teams prefer Slack, some like email, and some find it easiest to pick up the phone. Discussing preferences and setting up a communication model that suits your team make it possible to collaborate more enjoyably and productively. It's also well worth clarifying who can be contacted in case a misunderstanding crops up between employees. It's not a good experience when you're not sure if a colleague has read your question and is preparing a response or if they've started working on it.

That's why I consider it so important to agree on a "mandatory response" model within my teams. The model means that if you are asked about something but you cannot respond in full immediately, or you need to clarify some information with the customer, at least you are still responding. This allows your colleagues to switch to another task, and prevents irritation arising from email silence.



PROJECT WORK

Every posse will have their own approach to working on a project. Some places require code reviews, while in others you can update the master branch. There will probably be established ways of doing things to help simplify your work on code review and conflict resolution. But equally, in my time I have met many teams where this does not apply.

It's best to define the project code as early as possible. Depending on the language, platform, there can be various ways of doing this. But a single style will greatly simplify the code review, and you won't run into conflicts due to the fact that someone uses two spaces, while someone else used four.

Protecting repository branches from changes and setting permissions are good fail-safes. Some people think that this is not necessary, but several times I have had to deal with a situation when the automatic branch pull deletes the dev or master branch.

Configuring CI / DI, using git hooks with mandatory testing, before changing the main branch, is a very useful practice. This allows you to ensure that you always have valid code. The likelihood of a situation when someone breaks something is greatly reduced. Yes, this approach slows down the addition of changes to the code, but it can save a ton of time in the long run. If you often have situations that hotfix should be made as soon as possible, flexible pipelines can help (for example, adding comment specifications to ignore tests). Also, if these situations are constant, you may need to change something during the development process.

If you have code reviews, agree on how often they should happen. Ideally, it should occur as early as possible so as not to delay development. But since all employees are usually busy, it's good practice to arrange specific times, for example, first thing in the morning, after lunch and an hour before leaving. Then you will avoid the situation where employees are left idle for half a day because everything has been done and they're waiting for the code to be reviewed.

Document the project, especially if more than two people are working on it. This will help both developers and testers as well as the customer. Maintaining up-to-date documentation is time consuming, but on large projects it pays off. It's very convenient if there is proper versioning of documentation and notes on changes that are tied to the version (or date) of the project. From time to time you will face a situation where the product is tested according to different documentation than under its development (for example, where documentation has already been updated, design changes were only in the comments for the task, etc.). It generally takes a lot of time to clarify and update requirements.

Finally, there is always a chance that you will find yourself working on code produced by another outsourcer. Any low-quality code will send your cortisol levels sky-high. Upping the frequency of code reviews and quality assurance can help ensure the product meets requirements and can be easily maintained. Nobody is perfect, and most customers understand that where mistakes were made by a previous outsourcing company, the reasons (if not the ways of solving it) are generally quite straightforward. The key, for joint success, is to remain calm and keep communicating.

REGULAR CALLS AND MEETINGS

If you have regular project calls and meetings in the diary, try and make sure they don't take up a lot of time. In some teams working on a Scrum basis, a daily call could see more than 20 people participate and they could last more than half an hour. After a while, inevitably, almost everyone begins to do something else in parallel rather than participate in the discussion. There are plenty of ways to ensure things do not drag on too much, including replacing meetings with stand-ups or huddles.

SOME FINAL MYTHBUSTING

One final piece of advice in closing is it's helpful to look at things from the other side of the fence. Again, this is much more easily done on horseback. Critics of outsourcing believe that the arrangement needs constant management and a finger on the pulse. Lack of proper and constant management is indeed the most common reason why outsourced projects fail. But with a skilled manager in constant contact with the company, who understands the requirements and goals and ensures at all times that the company understands the requirements too, you can play a vital role in preventing a project galloping out of control. Just don't forget that a properly set up process can always optimize the number of man hours required in big teams. With all you've learned in all those teams, you can always bring a lot of value to any new project.

THE AUTHOR

SERGEY IVANOV

Lead Developer, Team Lead

Sergey is a Senior software developer and Technical lead with 14+ years experience. There were projects, in his career, implemented by him alone and also many involvements into projects with large distributed teams, Agile teams. In this regard, Sergey shares some practical tips.

E-NGINEERS

DO YOU HAVE ANY QUESTIONS ABOUT THIS ARTICLE OR SIMPLY WOULD LIKE TO DISCUSS SOME ASPECTS OF WORKING IN DISTRIBUTED TEAMS, PLEASE CONTACT SERGEY:

sei@e-ngineers.com

WHEN E-NGINEERS CAN BE OF ANY HELP, YOU ARE WELCOME TO CONTACT HANS PEETERS

+31 (0)6 51245088
hanspeeters@e-ngineers.com