Distributed Development: Fade In (Part 2)

5 February 2015 | **Business Innovation, Software Engineering** | **Miloš Andrejić**
Reading time: 4 minutes

In the previous part we covered environment setup, while this part will focus on transporting domain context and other relevant information in a distributed setup.

Domain knowledge share – transporting the context

It is much easier to join a project that's already in full swing when you can find most of the information about it in a project wiki. And I'm not talking just about the software architecture part, but also about many of the small infrastructural things, such as URLs, credentials, daily routine, common processes, and most importantly – business logic. Informal documents created by other team members like flow diagrams, class diagrams or any other type of documents/sketches are also very welcome for the newcomers.

It can be really hard to join the project if these things are not documented properly, especially in a distributed setup. This is because you'll need a lot of asynchronous communication with your distributed peers, and this will add a lot of overhead.

It is necessary to have documentation written in a language that can be understood by all members of the team. If the documentation needs to be translated for new team members, focus should be on the initial setup and configuration of the environment. If professional translation services are not available, you can assign some of the translation tasks to the development team itself (but only if this is doable within available capacity).

Speaking a common language is a must! Translating the entire documentation, team processes and how-to pages to a common language is a time consuming process that ends up saving a lot of time in the end, so please don't take this part lightly. If the project is at a start, it really is the best thing to agree on a mutual language in which all documentation and correspondence should be written and to maintain this agreement. Usually, the best candidate is English.

> As an example, we had a project that was written in a non-common language because it was written by a customer and the common team language was English. This caused a lot of overhead for team members that spoke the language because they had to translate and explain everything in English for people who didn't speak the language the documentation was written in.

If the business logic of the project is very complex, it poses one more obstacle for the distributed setup. This is especially true if the team is not familiar with the business domain. In these cases it is of utmost importance to have a dedicated person at the customer's location, to analyze processes, document and transfer them to the distributed team members regularly. It is also advisable for all distributed team members who need to know this logic, to spend as much time with the business as possible when they visit customer's location.

Regularity in transferring knowledge is key as always. Everyone should be aware of team members who hold specific domain knowledge on the other side from the very inception. Everyone has to use the same communication tools so they can be reached easily and regularly. Common keywords for domain concepts should be established in every part of the team, but also with the customer, and these have to be in a common language exclusively.

> In one of our projects, especially in the beginning, we organized domain knowledge share sessions whenever we met in person: talks were about a specific part of the domain and each talk would end with a Q&A session.

Next blog topic is a very important one: how to align with the team during distributed fade-in, so we recommend that you stay tuned to our blog series.