SetalaGSE

===

Casper Lassenius: [00:00:00] A short while ago, I had the chance to sit down with Mr. Marco from Nokia Siemens Networks and talk about global software engineering. We started our discussion with talking about why companies would be interested in doing global software.

Marko Setälä: there are a couple of reasons why companies are looking for global software development.

One is that being close to the customers so that unfortunately in certain industries, Finland is not the central of the world. Second thing is that there might be need for certain competencies and skills which are not available in one particular country. That root cause might be that there's a high peak of the need so that people are not available.

Or the competency itself is not available.

Casper Lassenius: One thing that we often hear related to global software engineering is companies seeking big cost savings by using countries [00:01:00] with much lower wages that we have here.

Marko Setälä: In some extent, it matters when the competition is coming bigger and bigger. And if the difference is too Then the companies need to look at how to catch the competitive cost structure in place.

But in the end, the customers are still value most about the software products fit for use and the companies who are bringing the most value.

Casper Lassenius: If we decide to do global software engineering due to competencies or being close to the customer or maybe seeking cost savings, how do you do we know whether we should be doing it or not.

Marko Setälä: One reason is that if you are making smaller products, small teams, it doesn't make sense to start distributing when the time and the amount of the people is increasing or you are making something specific for some special markets. Then that might be the starting point for start, start looking about development in other countries.

Casper Lassenius: Are there any situations that you see companies making wrong decisions?

Marko Setälä: In many cases, they are [00:02:00] strong belief that you can transfer the competence in one place to another. Some people might even think that it can take place pretty fast, and based on my experience, it might take at least two releases when people start really understanding.

What the product is all about, especially when we are talking about the legacy product, meaning the products which have been on the markets for a while, most probably companies will not get fast cost saving.

Casper Lassenius: This leads us into a very important topic, which is issues you run into in global software engineering.

You mentioned one very important issue right now, which is transfer of competencies. What other kinds of issues do we run into when we distribute our development to various countries?

Marko Setälä: We need to understand always a little bit about the culture of each country. Then the more teams and the more countries are.

Involved the need for an additional effort for communication, ensuring that people understand and sharing the things in the same way. That is, in many cases, the most [00:03:00] difficult part.

Casper Lassenius: The fewer sites, the fewer cultures, the easier it is for you. Communication is one very important issue and knowledge transfer.

Are there other issues you run into when distributing development,

Marko Setälä: you need to be able to be ready to invest, at least in the tools, whether it's related with the development tools or the communication tools. Concrete example is that teleconference lines, the further you go. And when, for example, if there are people which try to start communicating the language which is not their native language.

They are coming pretty interested discussions every now and then, especially if the phone lines are bad and sometimes that what happens, that there might be some issues with the bandwidth of internet communications so that you might lose one day work if something's not working properly.

All these need to be taken into account.

Casper Lassenius: It's important to have a good communication infrastructure in place. Yes. Are there other [00:04:00] investments you need to make?

Marko Setälä: It also requires a little bit different skills. Software engineers are a little bit introvert. Courage to take the phone and call is much, much higher.

And then when you on a table any more seat around to one table or one corridor it's becoming actually pretty tough

Casper Lassenius: the threshold for initiating communication becomes higher, and that reduces communication.

Marko Setälä: It reduces, and especially if even most of the books and articles written say that the best way to build the team is that people see each other face to face.

Humans are social animals which need direct and face to face communication. And if people have never seen each other, the barrier to start making phone calls, even sending emails or using chat channels, it's actually pretty high.

Casper Lassenius: What about time zones and communication?

Marko Setälä: So far I have not seen that it's always some kind of compromise if you compare time zone so that [00:05:00] if looking from Finland point of view, if you are having the teams in far east like China, then it varies between five to six hours.

In India, it's a two hour and a half and up to three hour and a half. That means that the common working time, it's not the whole day, which means that if you want to need to agree something, you need to take that into account. And then the compromise, so that sometimes people in both locations need to be flexible and at work beyond their, the, let's say, traditional working times.

Casper Lassenius: So you mentioned that in both ends. So it's about sharing the pain. Of timezone differences,

Marko Setälä: I guess that's come from pretty natural vision so that when the deadlines are coming so that people are normally finding whatever ways to get things done.

Casper Lassenius: What about holidays?

Marko Setälä: I guess that each country are having a set of days which are pretty important for their culture.

New York Festival in China, Diwali in India, Christian holidays in the Central Europe, Thanksgiving in the [00:06:00] US . People are not willing to work during that holiday period. You need to understand, you need to know when people are having those additional point, start learning how people are actually communicating.

What is the semantics of some answers, how you should actually address some possible issues. For example, in eastern culture, people don't want to lose their face, which means that if you are making yes, no questions, you will most probably look at over. Try to use open questions, meaning that, can you describe how you have solved that one?

Have you tried this one? Have you find it difficult to operate with that one? Can you show us the test cases? Can you show us the demo? Can you show us a working progress if we are talking about some document, for example? Then that's the area where you can dig out whether there might be some

issues,

Casper Lassenius: what we'll be talking about.

Communication has mostly been about synchronous communication. What is the relationship between a need for synchronous versus asynchronous [00:07:00] communication in

global software development

Marko Setälä: Personally I prefer always the direct communication because let's face the fact if you are using, for example, scrum, many people do not hang around their emails.

And emails is the most dangerous. It's some kind of outsourcing the problem somewhere else, meaning that yes, I have sent an email that means nothing in global software engineering, if there's

a thing, need to be clarified, need to be done accurate. Some need for help. The email. people are getting too much, many emails.

You don't know whether people have read it, whether some actions have been taken, whether the message have been as understood in many cases, so that whatever you try to communicate, the main assumption is that people will interpret that differently. And you originally.

Casper Lassenius: Should we

focus all our efforts on synchronous communication and try to avoid asynchronous communication then altogether,

Marko Setälä: I [00:08:00] believe that we cannot forget a synchronous communication, but it is only some kind of backup or something that general information sharing, which is not relevant in the time point of view.

And otherwise, try to use direct communication, whether it's a chat channel, whether it's a video conference. Whether it's a direct phone calls, whatever.

Casper Lassenius: This leads

us to the question of the different media. As you mentioned, very typically we have a set of possible media that we can use for synchronous communication like phone calls, chats, and video conferences.

When should we use which media?

Marko Setälä: If you start looking about distributed work, the starting point is naturally try to have some kind of kickoff face to face. If the original reason for distributed development is to looking cost savings, then that might be even difficult to start investing immediately face to face sessions, when developers are doing the work, most probably they prefer the chat.

If you can use the collaborative tools, for [00:09:00] example, editing the same files, that would be even. Peer working that has been tried several times and you can do it and it's actually pretty good way of creating the common understanding what we are supposed to do. Video conferences, if people are using Scrum sprint planning might be extremely good for video, conferences, phones, whatever.

There's issue which need, you need to get the answer immediately.

Casper Lassenius: Different

situations require different media and you should have all available as far as. Knowledge sharing, you said it's difficult and it can take up to a few releases until a new site comes up to speed in a project. How do you facilitate that learning?

Marko Setälä: I guess the best way is that if you can use peer work, you can send somebody in other site, and actually it could be done in the both way. So that people get first knowing each other and then normally people are learning by doing things, helping each other. It also requires a little bit of mental change so that in many [00:10:00] cases when companies, especially if the companies start establishing new sites, new locations, the existing ones see that as a threat, which means that once again, we are.

And it leads to the situation that it is not seen as a part of we, it's they. Then we need to be really careful and understand that what are the needs for the humans. If the people feel that there's a high risk of becoming redundant. I have not seen any good tricks how to actually be successful in that one.

If there are cases that, that in long term situation is that yes, this product will be moved in one place to another, then one opportunity and possible is that it is actually set pretty openly. After two years, this product will be done only in this other location and then try to make the story and what is the continuation from the.

Team, but it naturally depends quite much of the situation, how the company's doing. [00:11:00] It's easy to say in this kind of discussion, so that use A, B, or C, but it depends quite much that what is the business situation of that particular company.

Casper Lassenius: It's about dealing with the possible fear.

Marko Setälä: Yes, people are looking, what is it?

Whenever you are inviting people to get involved, you need to have the story. What is it for me?

Casper Lassenius: What are symptoms? What kinds of things do you see when it becomes us versus them? Situation?

Marko Setälä: When I start hearing they, that's the easiest way to start recognizing. A second thing is that you start hearing, claiming and finger pointing.

That's the indication. Easy questions to see a situation. Is that how many times you have discussed with your colleagues from the other? . These are the quite self evident, but these are the early big signals when you start recognizing things are not proceeding or in, in the sprint reviews, less and less are actually getting done.

That's so naturally, but many times that the big [00:12:00] signals are coming much earlier. And if you are not reacting, then immediately then you might face the situation that you have already lost one sprint. And that's, that the companies don't have a fault to do with that.

Casper Lassenius: How do you solve or avoid the us versus them mentality.

Marko Setälä: The

main issues inside the projects are non-technical. Most of the people are educated as engineers. They can solve technical issues, but whenever the people came on board, things are getting complicated. We are complex adaptive systems in the big organizations. How to try to prevent that. I believe that's when you start knowing the people, meaning that the more you are having the common work, the more you share the same goals you are working together, the more gelled the team will became.

Casper Lassenius: What does working

together in a global context

mean?

Marko Setälä: I guess it varies a lot. I don't believe if using the methodology of scrum. That I would not start at [00:13:00] least in the first place so that half of the teams are in location A and half of the teams are location. B. How team can work together is that for example, code reviews, that is also part of the sharing and competence development, working and identifying the use cases that can be done together.

Preparing the demos there, you can already start working together. It might not be task as such, you can start with the small ones, asking comments from document you have written, because when you are going in the one location to another, there are a couple of things that will actually change. The need for documenting things is actually increasing a quite much that is, once again, good way of start, co-operating.

For example, using webcams when you are having the chat, because quite much of our communication is actually body language. It's not what we are talking. It's good to remember that whenever there's a conflict between the words and the body language means.

Casper Lassenius: [00:14:00] What is the role

of face to face meetings and

travel?

Marko Setälä: The main, I guess the main challenge is that when people are traveling, it's immediately visible in the company's cash. It's money away from the cash. When we are misunderstand couple of user stories, theres not yet price tag immediately, which means that the making the business case really so that people who are not maybe so familiar with software engineering really understand what.

Impact as such. It might be that this cost became visible only after releasing our product, meaning that increased failure ready demand, we are start spending more and more time on, on maintenance type of work, clarifying things, sending hero teams to solve the customer issues. And so even the rule cause is that people have just misunderstood couple of user stories.

What need to be done actually done. How the customers are using the products as such, I believe that having multisite work, it cannot be done [00:15:00] without any trouble. What is the right amount of the travel then? We are once again having that, that whatever consultants . Normally saying that, it depends on the majority of the teams in different locations, the competence level.

And in competence, there are two kind of competence. It's a domain competence. You understand your business environment and for example, software engineering, competence. Those are a little bit different things. And the main issue is that how to make it visible, how to make it transparent, how to make it understandable for the non-engineering guys that actually why we need this kind of face to face and why we need to actually travel every now and then.

Casper Lassenius: How do you make it.

Marko Setälä: Unfortunately, in many cases it's only quite reactive mode after something has gone wrong and figuring out that what was actually the root cause and many times it actually happens that the root cause is that missing communication, misunderstood requirements or user stories even.

It's a written, but [00:16:00] there's one story In my experience and in my, in working life in very complex product, there was instructions how the installation framework should. And it was written a document. It was reviewed and then it was spread to the teams, and six teams implemented that a little bit different way it didn't fit together and how to actually should have done it is that the guy who actually written the document should have actually visited.

And ensuring that the team has really understood and working perhaps even some time with the teams and ensuring that they have really understood the framework.

Casper Lassenius: You mentioned communication, infrastructure. . And what about the technical infrastructure, let's say configuration management and development tools.

Is it important that all sites share common repositories and common tools, or can they work separately with the own

tool sets

Marko Setälä: when companies have start using iterative development that the cycles have become shorter. The continuous integration is absolute. and that requires immediately that whenever [00:17:00] the developers are making checking, it's immediately visible in all the other locations.

You need to have one place where an software is actually moved from one location to another, how to actually do it. Then once again, you can make it and implement that in different way, but the idea is that whenever you are touching the code, it should be visible immediately to other guys after the rebuild

if I would put my own money on the company, I would implement it in that way. If there are some other reasons that, that in software you can do nearly everything and in even stupid things, that's up

to you. But if I would put my own company, I would immediately ensure that, that we have the common system and integrated continuous integration system in place.

Casper Lassenius: Are there other big things or issues in global software engineering that we need to think?

Marko Setälä: Even it's related once came with this communication, say, if looking about. How many steps they are from the customer to the developer. And whenever we are increasing [00:18:00] the number of locations and number of teams in different places, it once again, additional handover.

And whenever there's a handover, you are always losing piece of information, some piece of information that we need to at least understand

Casper Lassenius: these days.

We can see many companies moving towards Agile approaches to software development, whereas traditionally they have been using waterfall sequential processes.

How do you see the fit between global software engineering and these two models?

Marko Setälä: We need to understand that in, in iterative development, the cycles are much, much faster. And then the creating the common understanding between the teams is even more important. If you having the traditional water polish, that they are, they have never been any company just doing pure waterfall as such.

Then there's a belief that you can, make the intermediate product outcomes or documents and people understand that at the same. , when you start scaling, if you're [00:19:00] talking about iterative, whether it's a KanBan, whether it's a scrum, it requires once again some special attention how you actually are being able to lead the team coming from beyond 50.

Beyond 100, beyond 500, beyond 1000. It's a scaling issue. Scrum originally was developed by co-located teams, up to few teams with one product owner. Scaling is issue and once again, it is not easy. Then we are coming back that whether we should target in component based team or feature teams.

I have seen that after even painful learnings you can do, distribute development. Using the scrum, but it is not easy, and this is something that you cannot learn by reading the books. So you can

always get some hints. But personally I don't believe that some best practices can be used because for me, the practices are always context dependent.

It [00:20:00] requires that you understand the culture and the teams and so on. And the most dangerous way is that you copy paste. one practice without really understanding the whole context around that one, why the team have end up certain solution. Some practices might work extremely good in one environment and be, can be totally disaster in other environment.

It's good to share experiences, but we should understand what is actually the difference between the really the practice and the experience.

Casper Lassenius: Doing large scale distributed agile is a learning experience.

Marko Setälä: It will be a

journey.

Casper Lassenius: There has been during the, let's say last decade, this push and a lot of talk about global software engineering.

That's the future. And we see a lot of companies doing global software engineering these days. It's almost a standard motor operation. How do you see the future? Will we see companies going towards doing more local development again or has this

come to stay.

Marko Setälä: believe that what might [00:21:00] happen is that, that where we are start getting more and more data, meaning that what has been the real success?

What has been the overall cost, that many companies have not really calculated what are the. Total cost of distributed development. I believe that in some extent the distributed development will stay, but the companies, they will learn and they will be, for example, minimizing that putting more emphasis on the architecture so that it is more modular, meaning that some features can be developed in parallel in different places.

Reducing the dependencies and reducing the sites involved. I was once visiting in one big company from us and they were telling us that the product was actually done in five different countries. I believe that most probable they will start reducing the number of [00:22:00] sites because adding additional sites, it's not only doubling the need of communication, it's going to exponentially.

It, but it will most probably it will stay. Companies will take those learnings, most probable through motor products, through less dependencies learnings, most probable through motor one product. That's the way to go. Or making the smaller products, which can be done independently, but most probable that will stay because where we started the discussion, the cost factor is not the only.

And in the global companies you need to be where your customers are and sometimes that's, they might be the market situation that actually you need to be close with your lead customers.

Casper Lassenius: Thank you very much, Marco, for your valuable time and I'm sure our students will appreciate the learnings they can get outta this.

Thanks.

Marko Setälä: Thanks.[00:23:00]