Risk Management in Global Software Development

Weihua Hao
School of Computing, DePaul University
243 South Wabash, Chicago, IL, 60604
WHAO@cdm.depaul.edu

ABSTRACT

There are a lot of papers concentrating on the benefits getting from global software development, and that is really true, but meanwhile we also need to realize that the risks in the global software development can lead to the failure of whole project. And there are some future risks arising after the development phrase. So it is really important to find risks and control them during the period of global software development.

KEYWORDS

Global software development, Risk, localized software development, Quality, Upgrading phrase, Maintaining phrase.

INTRODUCTION

Global software development is getting more and more popular in the software industry; a lot of companies adopt this way to reduce the development cost and cycle time arising from 'Follow-the-sun' software development and time-zone effectiveness [1]. However at the meantime, it also introduces some risks for us, especially the future and potential risks.

On this paper firstly we will discuss the risks associated the global software development from the time dimension; there are three main phrases – developing phrase, maintaining phrase and upgrading phrase.

After discussing the potential risks associated the global software development, the rest of the paper will provide some suggestions to control these risks.

RISKS OF DEVELOPING PHRASE

There are more risks in global software development than in localized software development. The main risk is you may get lower quality product from the global software development than the localized software development. The reasons are global software development is more complex, and offshore team members are less innovative. We will expand the reasons which lead to the low quality product.

- **More complex organization**

  In contrast to the localized software development, besides the regular teams, you also need to organize team form the angle of geographical location; Figure 1 is a typical organization for three different teams. That means you have some logical teams with people from different location, and some real team with people from the same location.

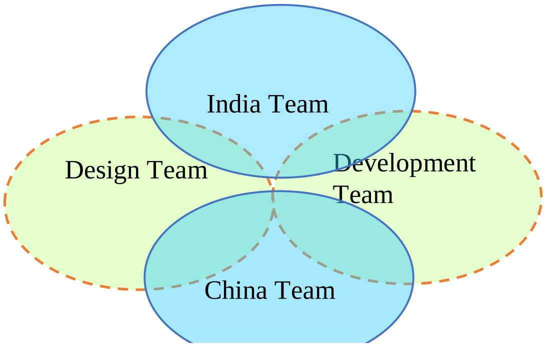

Figure 1: For example there are three different location team (US, China and India), then the real organization looks like this.

The question associated with the complex organization is you have more sub-team managers and less workers.

- **More expensive communication**

  Why is the communication in global software expensive? Different team is located in different country, and they maybe speak different language and have different culture, so it's much hard to convey something to a team with speaking another language. They will probably misunderstand the meaning you want to express. Even if they can exactly understand what you said, the other question is the time zone. You are maybe in the different zone, so the time you can use to communicate is much less than in the same time zone. There are two ways to resolve the time zone problem, one is some team need to work overtime, and the other is postponing the due date of the project. Obviously both of them are not a good solution.
  So it will take much more time to communicate with each other in global software development than in localized software development. Time is money! so the communication in global software development is expensive.

- **More detailed design**

  Why do you need more detailed design document in global software development? In a global team, people cannot talk face to face all the day, so if the development team members find something in the design documents they couldn't understand well and accurately, they will spend so much time to work out this misunderstanding. So in a global team you need more detailed design documents, I think the more detailed your documents are, the more cost you should pay, the cost consists of time and money. So one thing you should clear is the design phase of the global software development will be more expensive and longer than of localized software development. And if the manager don’t agree this, then the result I can imagine is the delay of the whole project and the mess of the implementing.

So if you really want to do a global software development, then you must find a very professional and experienced architect. Sometimes finding the right person really makes the project successful. But in the real life, the more professional and experienced person you get, the more you will pay. And they are rare and hard to invite.

- **Less innovation**

  Especially in the software industry, the innovation of workers can influence the quality of the product. As I know, the project can be globally developed is usually short-term. That means the testing team mostly belongs to another companies. They can be hardly motivated to be innovative; they can find some normal bugs, but they cannot find some hidden bugs if they are not innovative.

RISKS OF MAINTAINING PHRASE

After the product entering the maintaining phrase the offshore development team may be disbanded, so if the maintaining team get some big problems they cannot resolve, it is hard to get support from the development team, or you find the right developer but he/she is located on a geographically remote site and cannot speak English fluently.

Besides getting support, you also need to fix the urgent bugs. Assuming you kept a small team of developers to deal with this situation, but these developer may not the developer associated with the urgent bug, so definitely it will take more time to fix the bug and may cause another bugs due to their unfamiliarity with those codes.

On the other hand, if you do not control the risk at developing phrase well, then you get a lower quality product more possibly. It’s very hard for an unfamiliar team to maintain a low quality product.

All in all, if you cannot respond the customer and fix their problems quickly and accurately at the maintaining phrase, this will cause the reduction of product sale and the falling down of the credit of your company.

RISKS OF UPGRADING PHRASE

After the maintaining phrase, you collected a lot of responses and suggestions about the product, so you decide to upgrade the product, at that time you will find that the team in charge of the prior generation of the product is disbanded, and the team member may be sent to another team or another company. That means you only can find a new team whose members are not familiar with the product, you must pay more money and more time to train them to be familiar with the business and the prior version of product, at the meantime you have to face the reduction of the quality. This is really a disaster for the product manager.

So the risk of the upgrading phrase is a potential and future risk.

PROPOSED SUGGESTIONS

As I know, the final purpose of global software development is producing a great quality product with the reduction of the cost and of the time, so it is worth to figure out some methods and enforce these methods to control the risks, especially the future and potential risks.

I think the risks is a bad cycle, one risks arose, and then the subsequent risk would arise. It looks like the cycle in Figure 2.

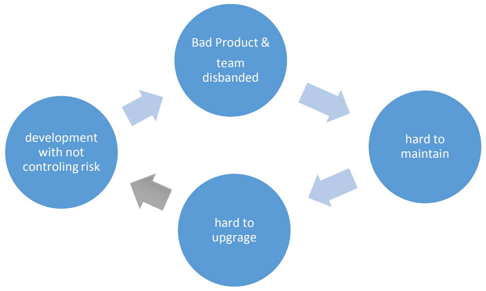
Bad Product & team disbanded
hard to maintain
hard to upgrage
development with not controling risk

Figure 2: the cycle of the risks

From the figure 2, we know that the source of risk is we did not control the risk well at the developing phrase. If this occurred, then we might get a bad quality product; and with the disbanded team it’s hard to maintain and upgrade this bad quality product. So the developing phrase is a very significant phrase of the global software development, we must consider all the risk and control them; and we must make sure that we can get the high quality product when the developing phrase is over. As you know a software product consists of documents and codes, so the high quality software product means very detailed and qualified documents and very robust code.

How to control risks and improve the quality of the product? I think there some principles we can adopt.

- Let the risks you cannot avoid arise earlier.

Then what are the risks you cannot avoid at these phrases? I think the most obvious two risks we cannot avoid in global software development are:

- More complex organization
  This is caused by geographical reason. If we care out more managers and less workers. We can assign a person multi roles at the same time.
- More expensive communication
  There are too many kinds of communication in the global software development, but in my opinion the most important one is the communication between designers and developers. We can let this risk arise earlier by splitting the design team to different

location, Making sure each team with different location at least has one designer. Then the local development team can communicate with the designer in the local native language.

- Development team will be disbanded after finishing the development of the product.
  This is caused by the pattern of global software development, and this is a big benefit of global software development, so we cannot avoid these. Ant we cannot let it arise earlier, but we can do some adjustment. Maybe we can keep some of the core developer or make a strategic relationship with the offshore team.

For the other risks we can avoid or control, we can using some methods some companies have applied successfully, like smart sourcing and standardization.

So here are my suggestions about how to control the risks arising in global software development.

- Splitting the design team to different location.
  This can let us face the communication risk earlier. Because we cannot assume that all the developer can speak English fluently. If they get a problem about the design, they can ask the local design team member directly using the local native language.
- Standardizing all activities.
  This method is not only useful for the global software development, but also for the all kind of software development. Because we must consider that some members of the team may leave this team someday, and we should reduce the influence of the leaving of members. Especially for the team of global software development, it is a short-term team, so members of the team have not a high loyalty. Considering about this, we should standardize all activities of the software development and make sure that each activity can be traced. Maybe Capability Maturity Model Integration–(CMMI) [2] must be enforced during whole development process.
- Very detailed design documents and related prototypes.
  A great, detailed and right design is a beginning of a high quality of product. How can we be sure the design is right and good? The only thing which can prove it is prototype. How can we measure the design is detailed enough? Actually there is no perfect way to measure it, but we can follow some standard design document templates to finish our design documents, like ISO9001 [3] and CMMI [2].
- Building counselor team and reviewing all the documents and codes.
  This is a very effective and usual way to improve the quality of the documents and code, and we must make sure all documents and codes have been reviewed by the counselor team. Of course the counselor team consists of most experienced engineers and architects.
- Making a strategic relationship with offshore team.
  In contrast to the short-term relationship with the offshore team, we can get more benefits and overcome the risks of short-term relationship from a strategic relationship with the offshore team. We get more innovative workers and more stable team from a strategic relationship. About the detailed content of making a strategic relationship, we can refer to the Smart Sourcing [2].
- After developing phrase, keeping a small and core team to maintain and upgrade the product in the future. Maybe this is difficult for some companies, because keeping a team all time means these companies must pay some extra money for this team. But I think it is worth, this small team is the real wealth for these companies.

CONCLUSION

If we want to be successful in a global market, we must control and manage the risks of global software development. This paper discussed some risks of global software development from my personal perspective and proposed some methods to manage these risks. Our methods of managing risks are summarized below.

We encourage splitting the design team to different location.

We encourage standardizing all activities.

We encourage building counselor team and reviewing all the documents and codes.

We encourage making a strategic relationship with offshore team.

We encourage keeping a small and core team to maintain and upgrade the product in the future.

ACKNOWLEDGMENTS

We thank SE volunteers, and all publications support and staff, who wrote and provided helpful comments on previous versions of this document. Some of the references cited in this paper are included for illustrative purposes only.

REFERENCES

1. Eoin Ó Conchúir, Helena Holmström, Pär J Ågerfalk, Brian Fitzgerald Lero, University of Limerick, Limerick, Ireland. Global Software Development: Never Mind the Problems – Are There Really Any Benefits?
2. CMMI:
   http://www.sei.cmu.edu/cmmi/
3. ISO9001:
   https://es.wikipedia.org/wiki/ISO_9001
4. Smart sourcing:
   Driving Innovation and Growth Through Outsourcing