

Risk Identification in Software Product Transfers

Darja Šmite

Blekinge Institute of Technology
Karlskrona, Sweden
+46 702-100858

Darja.Smite@bth.se

Claes Wohlin

Blekinge Institute of Technology
Karlskrona, Sweden
+46 455-385820

Claes.Wohlin@bth.se

ABSTRACT

Software companies today often face the necessity to decide where to develop their products. Inability to employ people or continue ongoing development with the same capacity in a given site of a company often leads to relocation of software work from one site to another. Software product transfers, however, are associated with numerous challenges that require investments, and may also have a secondary harder to capture effect on development productivity, quality and scope. In this paper, we share the results from previous empirical studies of software product transfers and offer a checklist for risk identification. The checklist shall be useful for software companies that consider, plan or execute software transfers. Although many risk factors included in the checklist may seem obvious, our empirical observations indicate that prior to obtaining the necessary experience these factors have been initially overlooked. Thus we believe that the checklist will be especially useful for managers with no or little experiences in relocating software work between the two sites of the same company.

General Terms

Management

Keywords

Global Software Development, Offshoring, Offshore Insourcing, Software Product Transfers

1. INTRODUCTION

To stay competitive in today's global marketplace organizations developing software-intensive products are often put in front of decisions related to evaluating whether to develop their products here, there or elsewhere. Employment in high cost countries has become challenging, and thus, it is not uncommon that many software organizations nowadays decide to transfer their software products from the "original" development site to an offshore site established in a low-cost country. While there are various assumed benefits of offshore insourcing, these decisions are often assessed in simple economic terms – "it is cheaper, and skilled labor is easier to find" [1]. However, there are indications that the assumed economic benefits are not always achieved. This is mainly because relocation of software work is associated with various challenges and extra costs, which are often overlooked or underestimated. Specifically, empirical observations suggest that a transfer of software work from one location to another requires direct investments into training, documentation and other transition costs, but may also lead to harder-to-capture secondary effects on productivity, quality and scope of future deliveries [2]. In other words, the value that you receive for the same money in different sites might not be the same. The magnitude of these consequences maybe affected by such factors as product complexity and maturity, product documentation, or previous

experience with the product on the offshore site [2]. Risk management in such uncertain situations is crucial. Because current research and practice related to software product transfers is quite immature, it is not surprising that companies often underestimate the complexity of transferring software work and do not always evaluate possible alternatives. In particular, while to transfer or not to transfer might not be a question, a company can wisely select products to be kept and those to be transferred. These decisions require input from product-aware managers, who are usually suspected to be biased and self-protective [4]. At the same time, for senior executives "service quality" and "technical competence" are esoteric aspects, and thus they perceive cost-savings as the primary criterion for success [4], which we believe may be in conflict with the potential risks for the product quality and maintainability.

In this industrial workshop paper we share a checklist for evaluating software product transfers and discuss the lessons learned from previous empirical work performed in a large telecommunications company in Sweden [2, 3]. The checklist prescribes identifying transfer process, product and people related risk factors and shall facilitate deliberate transfer decisions.

2. RESULTS

Our suggestions are based on empirical observations obtained through case studies of software transfers and interviews with experts in a case company. Research related activities are described in detail in our previous publications [2, 3].

Software transfers in the offshore insourcing context prescribes a relocation of work previously performed in one site to another site of the company. The offshore site often has no or little knowledge and experience with the product. A transfer is thus always a challenging task, since it requires the new staff to obtain product knowledge that might have evolved through years of development [5], and unavoidably leads to productivity slowdown [1-3]. Our empirical observations helped to formulate seven strategies for successful planning and execution of software transfers [3]:

- Evaluate the product-specific feasibility — not all products are equally easy to transfer. Product complexity, maturity, market pressure, documentation and other factors may affect the demands for specific skills and indicate the potential learning curve of the new site and thus their ability to handle the product.
- Establish the transfer process — successful transfers require a clear vision of the final state of the transfer and a well-planned process for handing over the work from one site to another.
- Evaluate transfer readiness — a company requires available and trained staff for handing over the work. Our observations suggest that employing and training the staff might take time.

- Avoid rushed and ad hoc execution — although longer transfers require bigger investments, incrementally planned transfers have a higher chance of smoothening the subsequent effects on the product by ensuring the necessary training time and availability of expert developers.
- Ensure resource availability, capability, and motivation — a successful transfer requires availability of motivated and skilled staff from the “original” site for teaching, and motivated and capable staff from the new side actively involved in learning.
- Ensure product maintainability — documentation and architectural improvement needs shall be evaluated prior to transferring the products to ensure possibly easier handover.

- Ensure efficiency — disconnection from the “original” site should occur gradually and carefully, because training will never result in 100% knowledge transfer. While the first year of independence is seen as a critical one, a company may decide to relocate several expert developers to the new site.

These strategies have formed the ground for a risk identification checklist (see Table 1). The checklist aims at supporting risk identification for events that may require additional attention and investments and may occur prior, during and after the actual transfer. These risks are divided into process, product and people related factors. Managers shall evaluate each factor in the chosen transfer context and select their choices in the result column. The overall score indicates whether the context is favorable for transferring the selected product or not.

Table 1. Risk identification checklist

| Risk identification at the moment of decision-making | | | | |
|---|---|--|--|----------------|
| | Factors | Favorable condition | Unfavorable condition | Results |
| Process | Decision | <input type="radio"/> Deliberate and discussed | <input type="radio"/> Forced or announced | ☺ ☹ ☹ |
| | Vision of the end state | <input type="radio"/> Clear | <input type="radio"/> Unclear or not communicated | ☺ ☹ ☹ |
| | Vision for the sending resources | <input type="radio"/> Clear | <input type="radio"/> Unclear or not communicated | ☺ ☹ ☹ |
| | Vision for the product | <input type="radio"/> Clear | <input type="radio"/> Unclear or not communicated | ☺ ☹ ☹ |
| | Process for the transfer | <input type="radio"/> Well-established | <input type="radio"/> Not established | ☺ ☹ ☹ |
| Product | Maturity | <input type="radio"/> Mature | <input type="radio"/> Immature | ☺ ☹ ☹ |
| | Complexity | <input type="radio"/> Simple or small | <input type="radio"/> Complex or large | ☺ ☹ ☹ |
| | Dependability | <input type="radio"/> Independent or decoupled | <input type="radio"/> Part of a compound system | ☺ ☹ ☹ |
| | Documentation | <input type="radio"/> Well-documented | <input type="radio"/> Poorly documented | ☺ ☹ ☹ |
| People | Receiving resources (competence) | <input type="radio"/> Competence in place | <input type="radio"/> Competence is incomplete or missing | ☺ ☹ ☹ |
| | Receiving resources (availability) | <input type="radio"/> Resources in place | <input type="radio"/> Resources are partially present or missing | ☺ ☹ ☹ |
| Risk identification during the transfer | | | | |
| | Factors | Favorable condition | Unfavorable condition | Results |
| Process | Overall vision | <input type="radio"/> Clear | <input type="radio"/> Unclear or not communicated | ☺ ☹ ☹ |
| | Schedule | <input type="radio"/> Sufficient | <input type="radio"/> Stressed | ☺ ☹ ☹ |
| | Approach | <input type="radio"/> Step-wise | <input type="radio"/> Full transfer from scratch | ☺ ☹ ☹ |
| | Process for the transfer | <input type="radio"/> Well-established | <input type="radio"/> Ad-hoc | ☺ ☹ ☹ |
| People | Receiving resources (competence) | <input type="radio"/> Competence in place | <input type="radio"/> Competence ramp-up | ☺ ☹ ☹ |
| | Receiving resources (availability) | <input type="radio"/> Resources in place | <input type="radio"/> Ramp-up of resources | ☺ ☹ ☹ |
| | Sending resources (competence) | <input type="radio"/> Competence in place | <input type="radio"/> Competence is incomplete or missing | ☺ ☹ ☹ |
| | Sending resources (motivation) | <input type="radio"/> Motivated | <input type="radio"/> Unsecure or confused | ☺ ☹ ☹ |
| | Driver for the transfer | <input type="radio"/> Receiving site (pull) | <input type="radio"/> Sending site (push) | ☺ ☹ ☹ |
| Risk identification after the transfer | | | | |
| | Factors | Favorable condition | Unfavorable condition | Results |
| Product | Market pressure | <input type="radio"/> Low | <input type="radio"/> High | ☺ ☹ ☹ |
| | Number of customers | <input type="radio"/> Small | <input type="radio"/> Large | ☺ ☹ ☹ |
| | Maintainability of the product architecture | <input type="radio"/> Easy | <input type="radio"/> Complex | ☺ ☹ ☹ |
| | Documentation | <input type="radio"/> Well-documented | <input type="radio"/> Poorly documented | ☺ ☹ ☹ |
| People | Receiving resources (competence) | <input type="radio"/> Competence in place | <input type="radio"/> More training is necessary | ☺ ☹ ☹ |
| | Receiving resources (availability) | <input type="radio"/> Resources in place | <input type="radio"/> More resources are necessary | ☺ ☹ ☹ |
| | Receiving organization | <input type="radio"/> Mature | <input type="radio"/> Immature | ☺ ☹ ☹ |
| | After-support resources | <input type="radio"/> Available | <input type="radio"/> Limited or unavailable | ☺ ☹ ☹ |

The checklist supports the risk identification by distinguishing favorable and unfavorable conditions. While these conditions are given in generic terms, companies may adopt the values to their own contexts. For example, product maturity, complexity, and market pressure may be judged relative to other products within the company. Note also that the risk factors in the checklist are not prioritized and the importance of each factor shall be evaluated in the context of each company or even transfer project.

We have applied the checklist in practice in one particular transfer project in the studied company retrospectively and reported our experiences in [3]. In summary, we have evaluated the context risk factors and calculated the average scores for the three phases (☺ received +1 point, ☹ received 0 points, and ☹ received -1 point). The identified risks in the studied project were addressed by mitigation activities and though the overall scores in the beginning of the transfer were low, our observations suggest that with the necessary investments a company can compensate the unfavorable conditions and successfully transfer a project [3].

3. CONCLUSIONS

3.1 Summary

Transferring software work across geographically, temporally and culturally distant sites even within one company is not a straightforward task. Related research indicates that transfers are often associated with decreased productivity and lengthy recovery. In this industrial workshop paper we have offered a list of risk factors that might help companies to avoid transferring products that are likely to lead to significant challenges and timely mitigate these challenges through well-planned transfers.

3.2 Relevance

We believe that our risk identification checklist shall be useful for companies that take offshore insourcing decisions, in other words consider transferring products within the boundaries of the company. Outsourcing, or transfers to external companies, may encompass additional challenges, because it gives companies less control of recruitment, transfer processes, and knowledge management.

3.3 Applicability

The risk checklist can be used for different purposes:

1. Risk identification — managers can use the checklist to identify the risks that may influence transfer decisions. This shall be especially beneficial for managers who have no or limited experience with transferring software work.

2. Risk management — results from the risk identification shall also support planning of product transfer projects, in which unfavorable conditions are mitigated throughout the transfer execution.
3. Selecting products for transferring — managers can perform risk identification for different candidate products and make a more deliberate decision of whether to keep or transfer a particular product based on the results.

3.4 Importance

Potential challenges discussed in this paper shall provide a deeper understanding of the complexities associated with transferring software work. Up to date there is little empirical research that addresses software product transfers, and specifically in the context of offshore insourcing relationships. And although some of our suggestions may seem obvious, our empirical observations suggest that without experience these are often initially overlooked [2].

ACKNOWLEDGEMENT

This research was funded by research grants from Ericsson Software Research and the Knowledge Foundation (2009/0249 and 2010/0311) in Sweden.

REFERENCES

- [1] Carmel, E., and Tija, P. 2005. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press, NY
- [2] Šmite, D., and Wohlin, C. 2010. Software Product Transfers: Lessons Learned from a Case Study, In *Proc. of ICGSE conf.*, (Princeton, USA, August 2010), IEEE Computer Society, 97-105
- [3] Šmite, D., and Wohlin, C. 2011. Strategies Facilitating Software Product Transfers. In *IEEE Software*, 28(5), 60-66
- [4] Hirschheim, R. and Lacity, M. 1998. Reducing information systems costs through insourcing: experiences from the field. In *Proceedings of HICSS*, 644-653
- [5] Mockus, A., and Weiss, D.M. 2001. Globalization by Chunking: A Quantitative Approach. In: *IEEE Software* 18(2), 30-37