

# Strategies Facilitating Software Product Transfers

Darja Šmite, Claes Wohlin  
*Blekinge Institute of Technology*  
 [{Name.Surname}@bth.se](mailto:{Name.Surname}@bth.se)

**Abstract:** Globalization of software work became a common way of how the market operates today. As a consequence of cost reduction strategies many product-focused software companies started to ship their products to their insourcing and outsourcing offshore locations. Unfortunately, moving software products from one site to another is not always a good business neither for the organization nor for the product. In this article we discuss our findings from studying software insourcing transfers in Ericsson, a large software product development company headquartered in Sweden. Our findings suggest that certain product, people and process related characteristics can facilitate the execution of an offshore insourcing transfer. Based on research conducted together with the company, this article shares a list of key factors alleviating transfers and seven strategies facilitating transition of software work across sites.

## Keywords

D.2.9.e. Organizational management and coordination  
D.2.9.f Planning

## Introduction

The process of globalization significantly transforms the way the market operates today. Consequently, a growing number of software companies turn to offshore insourcing – allocation of whole or parts of development to an offshore site within the organization. While some organizations choose to open a new office starting new development, others relocate existing software work from an “original” site to the new site. We call this process a software transfer and the objective here is to explore the strategies that facilitate such endeavour.

The starting point of our investigation are the myths that embrace global software work, which becomes almost anecdotally popular due to the promised benefits of cheaper and faster software development utilizing skilful low cost resources. Interestingly, it has been noted that despite the claims that companies start offshoring because of less offensive reasons than simply reducing costs, the main driving force for offshoring has always been related to costs [1]. Nonetheless, in reality the assumed benefits are neither clear-cut nor can their realization be taken for granted [2].

Aiming to achieve significant return on investments in a short time, companies often fail to realize that global software work is enabled through various settings: different locations, organizational relationships and types of work, to name a few. Each setting or scenario is associated with its challenges and benefits, thus informed decisions are vital. Unfortunately, by now discussions in research and practice on offshoring as a business approach can be characterized by the lack of common definitions and theoretically grounded explanations [4].

## What do we call global software work?

While a lot of research has been published on global software development, the readers can easily conclude that this label is attributed to quite diverse scenarios. Attempting to classify different scenarios we offer characteristics that can be considered by the companies initiating global collaborations (see Table 1). Classification of organizational relationship is inspired by [3], [6] and [8], locations by [4] and type of work by [8].

**Table 1.** Global scenarios

Characteristics	Choices	Examples
<b>Approach</b>	Transfer	Existing development activities are relocated from one site to another. This may result in distributed development or a different single-site execution.
	Distributed development	More than one site is involved in shared execution of distributed development activities either on a product, functionality or development stage level.
	Single-site execution	Development is initiated and undertaken in different locations across the globe.
<b>Location</b>	Onsite	On the premises of the company
	Domestic offsite	Within national borders
	Nearshore	With nearby countries
	Offshore	With a far location
<b>Organizational relationship</b>	Intra-organizational	Collaboration within an organization. Referred to as offshore insourcing, when sites are located in different countries.
	Inter-organizational	Subcontracting to third party vendors. If the main contractor is located in one country and the third party vendor in another country, referred to as offshore outsourcing.
<b>Type of work</b>	Entire project or product	New development, customization or maintenance
	Selected functionality	Sub-system, module or component
	Selected development stage	Coding, testing, etc.

Because the forms of global software work are quite diverse it is fair to assume that what works well in one context, will not necessarily work in another context [3]. Thus, it is important to distinguish between general and contextual experiences. The objective here is to provide general experiences and strategies that can be applied in other contexts than the actual case. However, these must be evaluated for any new context.

This article explores one specific scenario related to global software work, namely intra-organizational software product transfers – the process of moving work from one site to another site within the company. Our investigation is motivated by the challenges outlined by a large international software organization that aimed at better understanding the factors that influence software product transfers.

### Software Product Transfers

Aiming to avoid delays and inefficiency that is observed in distributed multisite development [7], many software companies started to ship their operation offshore. We call this process *software product transfers* and define it as a relocation of software development activities from an “original” development site (the sending site) to an offshore site (the receiving site).

This endeavour is associated with challenges related to transferring competence, knowledge and ways of working from people who may have handled product development from the beginning of its existence to people who have no or limited previous association with the product. This becomes especially challenging given the global context – geographic, temporal and cultural distance that may separate the sites.

Investigation of transferring different types of work within Lucent Technologies [8] suggests that successful transfers are associated with low coupling between distributed work items, as this decreases the amount of cross-site coordination and communication, and associated delays. The same investigation indicates that it is critical to ensure the capacity of the receiving site and substantial training for those unfamiliar with the product [8]. On the contrary, a study of cross-site modularization of work in HP, Intel and Fidelity [2] shows that decoupling has its pros and cons. The findings suggest that sending large ‘chunks’ of work reduces coordination and communication issues and provides the receiving site with some level of ownership, contributing to the sense of goodwill. However, there is no

efficiency gained in modularizing smaller tasks and too much independence might also be an obstacle to cross-site teaminess [2].

The findings reported in this article extend the mentioned research by proposing a broad list of process, people and product related factors that alleviate software transfers and seven facilitative strategies. These findings are based on an industrial investigation in Ericsson, a large multisite company.

## Overview of the Study

The studied company, Ericsson, is one of the leading suppliers in telecommunications worldwide and is rapidly extending its operation in Asia. The company develops a wide range of solutions, including generic software products offered to an open market and complex compound systems with customised versions.

Empirical data was collected through interviews, email inquiries, informal meetings, a workshop and a case study. Our investigation targeted two recently transferred products and two products in transfer. All transfers prescribed full responsibility for handling the products.

30 semi-structured one hour-long interviews were conducted with developers and different managers, including managers for products, development and actual transfer projects. The interviewees represented those involved in transfer activities from the sending site in Sweden (20 interviewees) and those from the receiving sites in India (7 interviewees) and China (3 interviewees). The interviews were held in person in Sweden. In addition, we organized email inquiry with three Chinese developers due to that it was impossible to meet in person and to avoid misunderstandings over the phone.

Qualitative analysis was used to derive factors that alleviate software product transfers from the data collected. Initial findings were then presented to the participants in a workshop. Discussions around the initial results helped to refine the factors. Interviews with four strategic product managers and a case study of an ongoing transfer were performed to validate the applicability and completeness of the results. Practitioners involved in the validation approved the existing factors and suggested new factors that were added. From the interviews and the case study we conclude that the alleviating factors are found to be useful for risk mitigation, relative to organizational experience, and also interdependent and thus shall be weighted in a given context.

## Factors that Alleviate Software Transfers

Our investigation suggests that certain products are easier to transfer. Here we refer to product as the object of a transfer, which in reality can be represented by the entire product, a module, a component, or a piece of functionality. We have also observed that independent of the complexity of the product, certain activities make a transfer easier. Table 2 and sequential sub-sections describe the favourable conditions and strategies that alleviate software transfers.

It is important to observe that factors in Table 2 are described in generic terms, since these are context dependent. For example, product maturity, complexity and market pressure are to be judged relatively to other products within the company.

**Table 2.** Factors alleviating software transfers

	<b>Decision</b>	<b>Transfer</b>	<b>Post-transfer</b>
<b>Product</b>	<ul style="list-style-type: none"> <li>◆ Mature product</li> <li>◆ Simple or small product</li> <li>◆ Independent, decoupled product</li> <li>◆ Sufficient documentation</li> <li>◆ Long product life cycle</li> </ul>		<ul style="list-style-type: none"> <li>◆ Low market pressure</li> <li>◆ Small amount of customers</li> <li>◆ Sufficient documentation</li> <li>◆ Easy maintainable product architecture</li> </ul>
<b>Process</b>	<ul style="list-style-type: none"> <li>◆ Deliberate and discussed decision</li> <li>◆ Clear vision of the end state               <ul style="list-style-type: none"> <li>◆ for the transfer project</li> <li>◆ for the sending site</li> <li>◆ for the product</li> </ul> </li> <li>◆ Well-established process</li> </ul>	<ul style="list-style-type: none"> <li>◆ Clear and communicated vision</li> <li>◆ Sufficient transfer time</li> <li>◆ Step-wise transfer</li> <li>◆ Well-established process</li> </ul>	
<b>People</b>	<ul style="list-style-type: none"> <li>◆ Competent and available</li> </ul>	<ul style="list-style-type: none"> <li>◆ Competent and</li> </ul>	<ul style="list-style-type: none"> <li>◆ Available and capable or</li> </ul>

	receiving resources	motivated sending resources ♦ Competent, available and active receiving resources	trained receiving resources ♦ Mature receiving organization ♦ Available support from the sending site
--	---------------------	--	---

### Illustrations from a Case Study

In order to illustrate how the alleviating factors can be applied in practice, we illustrate them on a case study. Here, the factors from Table 2 are put into context and hence product maturity, complexity and so on are judged in the specific case. This means that the factors are defined within the case study from the company perspective. For this purpose, we selected an on-going product transfer that started in January 2009 and finished in November 2009. The evaluation was performed retrospectively.

**Table 3.** Illustration from a case study

	Decision	Transfer	Post-transfer
<b>Product</b>	☹️ Immature product ☹️ Complex product ☹️ Compound product ☹️ Very little documentation 😊 No phase out plans	-	☹️ High market pressure ☹️ High number of customers 😊 Critical documentation ☹️ Product architecture has some complexities
<b>Process</b>	☹️ Announced decision 😊 Clear vision of the end state for the transfer project, the sending site and the product ☹️ Not established process	😊 Clear vision ☹️ Announced deadlines ☹️ Aggressive schedule 😊 Step-wise transfer 😊 Well-established process	-
<b>People</b>	😊 Gaps in competence ☹️ Half of the required resources are missing	☹️ Competent and motivated sending resources 😊 Active ramp-up of resources and competence in the receiving site ☹️ Insufficient initiative of the receiving site	😊 Receiving site trained 😊 Receiving site is young 😊 On-site and remote support from the sending site is ensured
<b>Total</b>	☹️	☹️	☹️
<b>Actions taken</b>	<ul style="list-style-type: none"> <li>• Developed documentation</li> <li>• Performed risk analysis</li> <li>• Planned the details</li> <li>• Actively recruited</li> </ul>	<ul style="list-style-type: none"> <li>• Promoted existing people</li> <li>• Performed hands-on training</li> <li>• Executed trial operation</li> </ul>	<ul style="list-style-type: none"> <li>• Transferred people with the product</li> <li>• Prolonged remote support</li> </ul>

The case study demonstrates that although the discussed factors alleviate software transfers, the initial unfavourable circumstances do not necessarily lead to inevitable failure. Facing disadvantageous circumstances, an organization may choose to address the risks, although observations suggest that this often comes for a certain cost. In particular, to mitigate the risks of failing (actions taken), the sending site invested in developing critical documentation, prolonging sending resources and transferring key experts to the receiving location. In summary, the circumstances in the decision phase were not optimal. However, the countermeasures secured the schedule of the transfer.

### Strategies that Facilitate Software Transfers

The strategies presented in this article combine advice for successful transfer and actions that can be taken to address the unfavorable circumstances in Table 3.

## **Decision**

Our investigation suggests that transfer decisions are often related to political or strategic reasons that are taken on the highest organizational level. Despite the limited ability to affect the decision process or result, there are strategies closely related to the way decisions are made, announced and discussed, that can facilitate the transfer.

### **Strategy 1: Evaluate the Product Feasibility**

Transfers are often expected to generate revenue and hence it is important to evaluate the feasibility of transferring a specific product. Transfers should not be too costly, too slow or result in a failed performance. Therefore selection of a product for a transfer shall be granted respectful attention and involve those levels of management that are capable of judging product complexity and maturity.

As illustrated by the alleviating factors, product immaturity, complexity, size, dependability from other products or components, and poor documentation can make transfers infeasible, because of the costs of employing human resources with specific competence or effort necessary for training. Also products with short life cycles may never reach their point of return on investment due to the additional transfer costs.

A Product Manager estimated that a full transfer (recovered performance and no involvement from the sending site) takes five to six years, and therefore transfer decisions shall have intentions for a longer duration than this to allow reaching the economic benefits.

### **Strategy 2: Establish the Transfer Process**

To change is to suffer loss of certainty, the comfort of the known and the familiar, the sense of competency, security and status that goes along with the existing order of things [9]. Therefore it is critical to establish a clear vision of the transfer regarding process, product and people to convince those affected that the gains will be greater than the losses. Otherwise fears, uncertainty and doubts could impede the change.

Vision is however not sufficient for ensuring successful execution of the transfer. Without a clear process, people get frustrated and either fail to execute the change or reject the change due to a lack of confidence in the organizational leadership [9]. It is thus essential to establish a transfer plan early.

### **Strategy 3: Evaluate the Readiness for the Transfer**

Any transfer is dependent on the readiness of the receiving site to take over the responsibility. Unavailability of the right people with the right competence at the decision phase means that the organization shall reconsider site strategy or carefully plan the deadlines. In addition, it is observed that poor readiness for the transfer, in particular in the case of complex or immature software products may become a threat to budget and schedule.

## **Transfer**

The actual transfer starts when the decision is made and the product is selected. Transfers in the studied organization are executed as separate projects with their own resources, management, plans, budgets and schedules. The focus falls on evaluating the probability of achieving the goals set in the decision phase.

### **Strategy 4: Avoid Rushed and Ad-hoc Execution**

Our findings indicate that due to limited involvement of development staff deadlines set for transfers in the investigated organization are often very optimistic. A simple explanation runs – transfers are seen as a process of building a receiving site as a mirror of the sending site and then switching the development. However, due to the knowledge intensive nature of software development, this process is significantly challenged by the difficulties of transferring knowledge and experience across geographic, temporal and socio-cultural distance. Thus, transfers shall be sufficiently planned and carefully executed.

A Tech Lead from India warned that transfers should not be pushed or rushed. Due to unavoidable impact that a transfer has on the whole organization, transition shall be organized step-by-step.

### **Strategy 5: Ensure Availability, Capability and Motivation of the Resources**

It is often challenging to find the right people in the right time, thus recruitment and training may be required throughout the transfer. Transfers are not terminated, if the receiving site is short of resources or competences. However, knowledge cannot be shared if the receiving site is not in place.

When the process of knowledge transfer is initiated, the motivation and capability of the sending site becomes vital. Naturally resistance can be expressed due to threats associated with training what may be perceived as competitors. In the studied case it was avoided by ensuring the future assignments and thus motivating the sending site. It is also worth noting that the employees had only a two-year history with the product, thus there were no hard feelings that are usually associated with transferring a product that people feel emotionally attached to.

Finally, successful transfers in the studied organization are characterized by an active leadership from the receiving site. Although the so-called “push transfers” are regarded as inefficient, “pull transfers” are challenging to establish. This often requires strategic selection of transfer management and sometimes a cultural shift.

### **Post-Transfer**

The key challenge for making a transfer successful in the long run is found to be related to sustainable efficiency and quality in post-transfer operation.

#### **Strategy 6: Ensure Product Maintainability**

Along with the people-related factors, our observations indicate that product characteristics may significantly affect its maintainability. Products with low market activity and fewer customers are easier to maintain, as the amount of work in the first month of the post-transfer development can be easier controlled. Interviewees from the studied organization also argue that a transfer cuts the history and generations in the development of the product. Thus, a complex architecture increases the threat of failure, because the new product owners often tend to add functionality without reengineering as they climb up the learning curve. Unfortunately, keeping the sending resources with the transferred product that could alleviate this threat is not always feasible. Therefore, product documentation plays an important role.

A Product Manager from Sweden noted that products evolve. This puts high demands on inexperienced teams, as they are not only required to keep the product “alive”, but also have capabilities to maintain and improve it, in particular when it comes to design.

#### **Strategy 7: Ensure Efficiency**

When the transfer project reaches the cut-off date, the receiving site becomes responsible for further development. Our observations however suggest that transfers are never complete. Knowledge and experience cannot be 100% transferred. Despite managerial intentions, the first year of independent performance is naturally critical and often associated with decreased efficiency. Therefore, disconnection from the sending organization shall be performed gradually and carefully. It can be as well achieved by ensuring post-transfer resources at the sending site or travelling to the receiving site to support the initial development through coaching. The maturity of an organization is also reported to act as a “safety net” that can support people in operation. In other words, it is essential to ensure an opportunity to obtain the necessary support from the sending organization after the transfer.

A Software Engineer from Sweden urged that even a year after the transfer is accomplished the receiving site calls and asks questions, because some unique problem situations may occur only once in several years.

### **Other types of transfers**

This paper has been focused on offshore insourcing, but many of the strategies may be applicable in other situations too, for example in outsourcing or transfers within a site.

**Outsourcing:** In outsourcing there is less control of recruitment, transfer processes, and knowledge management. Thus, outsourcing most likely adds some challenges and hence strategies to master it efficiently.

In summary, we believe that our results are likely to be useful in other global scenarios. However, more factors might be required to cover specific contexts and the relative importance of alleviating factors may change. For example, it is likely that knowledge transfer to an external entity will be supported by less positive behavior at the sending site. Similarly, organizations are limited in mitigating external employee turnover problems.

**Transfers in co-located environments:** Transferring software work from one development team to another as a phenomenon is not new. For example, such processes exist when transferring a product between a development team to a maintenance team. Internal transfers within a site has its challenges, although some of the challenges related to global transfers do not exist.

## Conclusions

The factors and strategies reported in this article are based on an extensive study of offshore insourcing in Ericsson that aims at evaluating the favourable conditions for transferring software work, the consequences caused by unfavourable conditions and solutions used to mitigate these consequences. Our findings illustrate the challenges faced by a sending site and a receiving site involved in transfer activities. Hereby we indirectly confirm that transfers are associated with difficulties related to decreased efficiency and challenging post-transfer development as pointed out by for example [1] and [10].

A Product Manager from Sweden emphasized that transfers always have consequences, which is natural. Therefore instead of putting blame on the transfers these problems shall be considered from the beginning.

Carmel and Pjia found that during the first few months of knowledge transfer the new offshore individuals are less productive as they “go up the learning curve” [1]. Kommeren and Parviainen report that it took more than five years before the offshore group of Philips in India had enough application domain knowledge to cooperate with the TV software integration center in Belgium effectively [10]. While our findings are hard to quantify, the interviewees provide useful observations confirming that transfers take time. Interestingly, while the product managers estimate a five-six years long recovery for a full transfer, developers expect the receiving site to be independent after the first year.

Our findings lead us to conclude that different products are differently suitable to transfer. We therefore emphasize, that software organizations shall take informed decisions regarding transferring work across locations and carefully plan and undertake the execution of software transfers. The strategies identified can be used in any product-focused software company. Application of these strategies is expected to facilitate the transfer by mitigating the decrease in efficiency and the length of recovery.

## Acknowledgements

We would like to thank Ericsson employees for their interest, active participation and support of this research. This work was funded by a research grant from Ericsson Software Research and the Knowledge Foundation in Sweden.

## References

- [1] Carmel E. and P. Tjia “Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce”. Cambridge University Press, NY, 2005
- [2] Conchuir E O, Holmström H, Ågerfalk P J, Fitzgerald B (2006) Exploring the Assumed Benefits of Global Software Development, In: Proceedings of the 1st International Conference on Global Software Engineering, 2006, pp.159-168
- [3] D. Šmite, C. Wohlin, R. Feldt, T. Gorschek “Empirical Evidence in Global Software Engineering: A Systematic Review” to appear in the Journal of Empirical Software Engineering; DOI: 947w6533p607u036, 2010
- [4] Jahns C., E. Hartmann, and L. Bals, “Offshoring: Dimensions and diffusion of a new business concept”, In: Journal of Purchasing and Supply Management, 12(4), July 2006, pp. 218-231
- [5] Damian D. and D. Moitra, “Global Software Development: How Far Have We Come?”, IEEE Software, Vol. 23, No.5, 2006, pp.17-19
- [6] D. Šmite, C. Wohlin, R. Feldt, T. Gorschek “Reporting Empirical Research in Global Software Engineering: a Classification Scheme”, In: Proceedings of the 3rd International Conference on Global Software Engineering, 2008, pp. 173-181
- [7] Herbsleb, J. D., A. Mockus, T. A. Finholt, and R. E. Grinter, “An empirical study of global software development: distance and speed”. In: Proceedings of the 23rd International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 2001, pp. 81-90
- [8] Mockus A. and D.M. Weiss, “Globalization by Chunking: A Quantitative Approach”. In: IEEE Software 18(2), 2001, pp. 30-37

- [9] Pieters W.G. "Reinvention Strategy: Using Strategic Learning to Create and Sustain Breakthrough Performance", John Wiley & Sons, Inc., New York, 2002
- [10] Kommeren R. and P. Parviainen „Philips experiences in global distributed software development“, Empirical Software Engineering, Vol. 12, No. 6, December 2007, pp. 647-660



## About the Authors



**DARJA ŠMITE** is an assistant professor of software engineering at Blekinge Institute of Technology and an associate professor at University of Latvia. Her research interests include global software engineering with the emphasis on improving distributed team efficiency, coordination and processes. Šmite received a PhD in computer science from University of Latvia.

Contact her at [Darja.Smite@bth.se](mailto:Darja.Smite@bth.se)



**CLAES WOHLIN** is a professor of software engineering at Blekinge Institute of Technology. His research interests include empirical methods in software engineering, software metrics, software quality, and requirements engineering. Wohlin received a PhD in communication systems from Lund University. He is the Editor-in-Chief of Information and Software Technology published by Elsevier. He is a professorial visiting fellow of University of New South Wales in Sydney, Australia, a senior member of the IEEE and a member of the ACM.

Contact him at [Claes.Wohlin@bth.se](mailto:Claes.Wohlin@bth.se)