

A FRAMEWORK FOR EFFECTIVE KNOWLEDGE TRANSFER OF SOFTWARE COMPONENTS ACROSS INTERNATIONAL DEVELOPMENT SITES WITH FOCUS ON OWNERSHIP

Krishnamurthy T.V
krishnamurthy.tv@nsn.com
Nokia Siemens Networks

Narasimha Nayak.
narasimha.nayak@nsn.com
Nokia Siemens Networks

ABSTRACT

The Era of the Global Software Organization is well and truly underway. More and more companies are locating their development sites across different countries to take advantage of the talent pool available to effectively leverage a distributed global workforce. In this scenario companies, which are able to strategically and effectively transfer applications and components across sites will be clearly at an advantage. This may help free up some of the strategic resources in the current sites and also utilize some newer resources in some other sites. While an effective transfer will reflect positively on the redeployed resources and continuing efficient functioning of the systems, a bad transfer shall surely drain additional resources in maintaining the redeployed system. In this paper we discuss some of the issues facing Effective Knowledge Transfer of Software Components and articulate a Framework that attempts to solve these issues with focus in detail on ownership transfer. We are sure that the same will strengthen the fundamentals of the Global Software Organization immensely and also pave the way for future research in this area.

Keywords:

SCKT, KT, Ownership, Handover, Source, Recipient

1. INTRODUCTION

The Global Software Organization is extremely successful and has actually been flourishing for quite a while. Promotional lines such as “The Global Conglomerate”, “The Multinational Giant” etc have become quite ubiquitous. In fact Corporations and organizations have been so successful in distributing and managing work across global sites that it would appear to be a weakness these days for an Organization to be present in only one country, i.e. in not being global. This is especially true for the software development organization. This can be also be verified with the speed at which organizations are acquiring global presences and adding international resources by way of mergers and acquisitions, thereby consolidating their positions and establishing themselves as clear global entities.

Some of the reasons that have been attributed to this distributed workforce phenomena are market presence, pressure from competition, differences in labor rates, availability of specialized talent in specific locations and of course optimization of resources. Also being transnational gives added advantages in terms of not being affected adversely by business climate changes in any particular country. Thus the teams also act as a basis for redundancy planning in case of eventualities. Thus we may find more and more complicated large software projects being developed with individual components assigned to teams in different countries.

While all this may present a rosy picture for multi-site multi-national software development, the same becomes an Achilles heel when companies want to shift some of these components between these sites for business reasons. This is because although knowledge management has been researched very well, we believe research is still getting organized in the area of knowledge transfer itself and in particular for the case of software component transfer. Of particular interest to us is the case of transfer of ownership of the software component and the timelines for doing the same. While smooth transfer of components enhances the organization’s resource utilization, botched up transfers end up placing tremendous stress on the teams that aren’t able to cope up to the transfer and hence burdens the organization and its management with additional support effort and drains both actual resources and the confidence of the organization.

Hence in this paper, we basically try to understand the impediments to a successful Software Component Knowledge Transfer (SCKT) and ways to mitigate or avoid these. The key focus to us is the knowledge build up for the software component and the ideal way for the “recipient” to accept the ownership transfer. Also we check how the success can

successfully after the transfer and keeps blaming the source and the project for not having, in its eyes, been given a fair deal during the transfer. The recipient may keep pointing out to alleged technical or process or project details that were missed during the transfer. The recipient may also raise issues regarding the whole transfer process; much after the same has been done. We propose steps in our framework that maintain record of transactions during the transfer that detail every session with conclusion that are agreed upon by both source and the recipient. This saves lots of effort from the project management later in terms of arbitration between teams.

3. THE FRAMEWORK FOR EFFECTIVE SCKT

In addressing the issues explained in the last chapter, we have come up with a framework of activities with key focus identified, which governs the whole knowledge transfer for the software component. The major part of our proposal is governed by a 3 phase knowledge transfer process, *a pre-transfer phase*, followed by *the transfer phase*, which is followed again by a *post-transfer phase*.

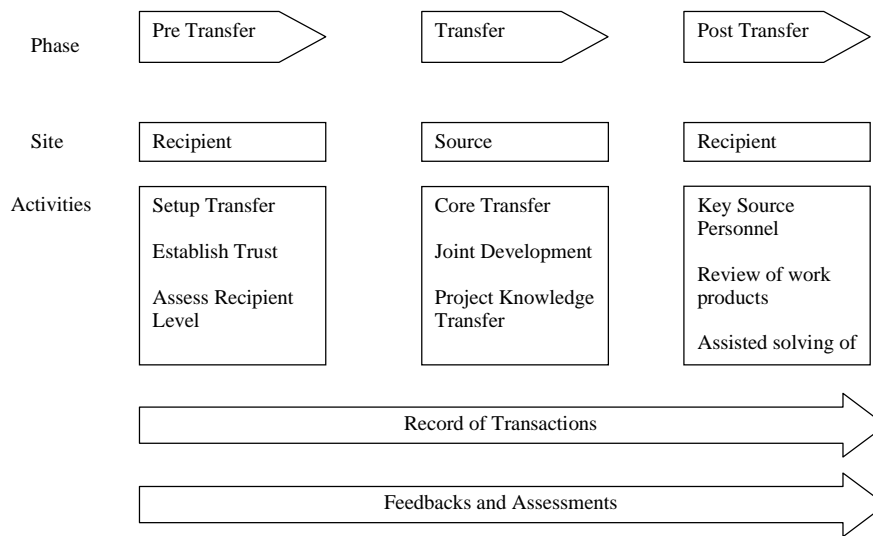


Figure1. Framework Overview

3.1 Overview of the Framework

The following table gives an overview of the three-phase proposal and the major points of focus of each phase

Table1. Framework Overview

Phase	Overview
Pre-Transfer (at recipient site)	Setup for transfer, establish trust, and assess current recipient level and key recipients for transfer. Transfer basic knowledge.
Transfer (at source site)	The actual transfer of core component knowledge, along with key recipients participating in actual development. Optional transfer of project knowledge too.
Post-transfer (at recipient site)	Complete the knowledge transfer by supporting recipients with key source personnel to review and assist the recipients work during this phase.

4.2 Pre-Transfer phase

These activities are conducted at recipient site.

Table3. Pre-Transfer Activities

Activity	Focus
Breakdown training areas	Identify a top down breakdown of the knowledge structure.
Identify key trainers (from source)	These trainers prepare training materials and also assessment tests
Prepare training plan (at source site)	Requires detailed planning with management and key trainers
Present training plan(at recipient site)	Familiarize recipients and get their approval or buy-in
Conduct assessments (at recipient site)	To get current standings of trainees in the knowledge areas.
Identify Key recipients(at recipient site)	These high potential recipients shall be the KT enablers, and play active role during the Transfer phase in bridging the source and the recipient.
Conduct initial knowledge sessions(recipient site)	Theses sessions will ensure the basic knowledge required for the transfer gets covered here. This could also be augmented with an example application being planned to be developed at the recipient site. Key documents left back to augment the learning.

4.3 Transfer phase

These activities are conducted at source site.

Table3. Transfer Activities

Activity	Focus
Revise and recap current knowledge.	Make recipients present the current understanding and assess their grasp.
Component knowledge Assessment	Assess recipient grasp of component knowledge by quizzes on component knowledge documents. (shared with recipients in pre-transfer)
Conduct Active core knowledge sessions	Conduct actual component knowledge transfer sessions in detail covering requirements, design, code and test infrastructure.
Conduct project process knowledge sessions (optional)	This is an optional part needed only when the recipient is also entering the project newly and hence needs to be trained in processes followed in the project too.
Integrate key recipients into source team for active development of a feature.	Give ownership of a part of the component to be developed for a particular feature / release to the key recipient(s), for them to actively understand and grasp the complexities (product and process) and also assess their progress / readiness.

at the recipient's site and scope of his work was made very clear. He was to act as liaison and help solve issues, either by offering expert advice or by rerouting to concerned colleagues in source / other sites. Here too the record of transactions was kept. Also source colleagues reviewed recipient's work in terms of design documents, and fault corrections to ensure that no major issues were found.

6. DISCUSSION ON THE FRAMEWORK AND CONCLUSION

The framework was derived based upon study of issues that our site faced with some of the earlier transfers. Also thought was given to actively make the transfer fruitful for both the source and the recipient.

Though we came across literature dealing with details on the source side of the issues [2] we did not come across material that dealt with knowledge transfer with ownership in key focus. But we have found in our experience that it is a common phenomenon for software components to get transferred between sites. And our framework gives us good leverage to systematically transfer the knowledge and also assess the transfer while the same is happening so that the ownership transfer is really successful.

It was also acknowledged in our project that the transfer that we did was tremendously successful. We do hope further research can take up from our framework and pave the way for more successful component transfers.

7. REFERENCES

Georg, Disterer. *"Individual and Social Barriers to Knowledge Transfer"* Proceedings of the 34th Hawaii International Conference on System Sciences -2001

Saonee Sarker, Suprateek Sarker, Darren Nicholson, Kshiti Joshi, *"Knowledge Transfer in Virtual Information Systems Development Teams: an Empirical Examination of Key Enablers"*, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)

Szulanski, G., *Exploring internal stickiness: Impediments to the transfer of best practice within the firm*, Strategic Management Journal, 17 (Winter 1996), pp.27-43.