
Radhika Jain and Sandra Richardson
Department of Management Information Systems
Fogelman College of Business and Economics
The University of Memphis, Memphis, TN 38152
Email: {rjain, srchrdsn} @ memphis.edu

Abstract

The software development industry spends more than half of its budget on maintenance related activities. Software testing provides a means to reduce errors, cut maintenance and overall software costs. Given the importance of software testing it is surprising that there is very little emphasis on formal software testing education. This suggests that most software testers are then either self-taught or they acquire needed skills on the job through informal and formal knowledge sharing mechanisms. We explore the role of knowledge management and specifically the various knowledge sharing and transfer mechanisms that enable software testers to acquire software testing skills. This is a multi part study. This paper reports our preliminary findings from the first phase. First phase data was collected at a large multi-national company. We expect that these guidelines can help organizations to devise a program for improving skills of their employees involved in software testing and institute a culture of effective use of knowledge transfer mechanisms.

1. Introduction and Motivation

Organizations are cognizant of the increasing expenditure on software maintenance. One of the reasons for this disproportionate cost is the lack of intensive focus on software testing. Effective software testing can reduce software maintenance costs significantly, improve software quality, and thereby increase customer satisfaction (Basili et al., 2001). Often, testing is performed with limited planning by stakeholders with inadequate training for software testers. Despite the well-understood concerns of software testing, the state of software testing practices and testing education remains lackluster. Survey of software testing practices in industry suggests that testers seldom receive the continuing training necessary to effectively do their jobs (Berard, 1996; Glass, Collard, Bertolino, Bach, & Kaner, 2006). This current state of software testing raises the primary concern of “How do software testers acquire new skills or improve their existing skills to become more effective at what they do?”

To address this concern, we investigate the existing communication, collaboration, and coordination patterns among various testing stakeholders at a large multinational organization, hereafter referred to as TransCo. We also examine these same mechanisms between TransCo and its offshore testing vendors where much of its software testing is performed. The result of this investigation is a greater insight into the existing processes of knowledge transfer that take place as software testers perform various activities and effectiveness of knowledge transfer mechanisms.
in the skills acquisition.

Informal mechanisms of knowledge transfer like these are being increasingly recognized by organizations as strategic to retain and transfer crucial knowledge. Examination of existing knowledge transfer patterns reveal how testers manage expectations and improve communication with managers, developers, and clients. It is our primary goal in this project to understand various such knowledge transfer (KT) mechanisms and identify those that are most effective in improving skills of software testers.

2. Theoretical Background

We draw primarily upon two streams of research literature. These include theories of knowledge transfer and software testing.

2.1. Knowledge transfer in organizations

In order to improve the skill sets of software testers, it is necessary that testers have a deep and broad understanding of current software testing processes and practices, their patterns, and implications. Developing this kind of understanding requires effective development of knowledge management tools on the part of the organization.

"Knowing is not a static embedded capability or stable disposition of actors, but rather an ongoing social accomplishment, constituted and reconstituted as actors engage the world in practice.” (Orlikowski, 2002, pg. 249). This implies that knowledge is both an active and social construct and its transfer cannot be understood in absence of an understanding of the organizational knowledge management structures, and the embedded systems employed to transfer knowledge effectively, efficiently, and creatively.

Knowledge management systems are important assets in organizations today. Defined as systems that support creation, transfer and application of knowledge in organizations, these systems play a critical role in knowledge transfer between individual workers resulting in reduction in both errors and costs as well as leading to the creation of creative solutions and problem solving. An additional result is an increase in organizational learning.

One outcome of knowledge management is the transfer of knowledge from one individual or unit to another. Knowledge transfer is defined as “the process through which one unit (e.g., individual, group, department, or division) is affected by the experience of another (Argote & Ingram, 2000).” Ability to transfer knowledge among different stakeholders through a variety of mechanisms such as technology, personnel movement, and blueprints (Slaughter & Kirsch, 2006) has been considered crucial to the development of strategic capabilities, and thereby influential factor in defining organizational performance (Argote & Ingram, 2000).

Several mechanisms have been proposed to impact the transfer of knowledge. The degree of proximity between the parties involved in such relationship is proposed to impact knowledge transfer (Slaughter & Kirsch, 2006). Hansen (2002) characterizes proximity between parties as knowledge networks. The shorter the network the more likely knowledge will be obtained as long path lengths lead to information distortion.

Organizational knowledge resides in reservoirs making transfer difficult, and even more difficult to transfer across different contexts (Argote and Ingram, 2000. This view of knowledge transfer identifies members, tools, and tasks as the three elements of knowledge transfer (Argote and Ingram, 2000). Understanding the interaction between these elements within an organization reveals the effectiveness and accuracy of the knowledge being transferred and the ease, or lack of, with
which knowledge is transferred.

Employee motivation is identified as a crucial element for knowledge transfer in organizations (Osterloh and Frey, 2000). This view can take on an economic flare and reveals that “effective internal knowledge sharing and transfer systems provides motivations to both the provider and seeker.” (Lin et al., 2005, pg. 199).

Knowledge transfer is also characterized by the type of knowledge exchanged; specifically explicit and tacit knowledge (Osterloh and Frey, 2000). The distinction is important when considering how to capture knowledge in a knowledge management system with the goal of transferring knowledge throughout the organization. In addition, Osterloh and Frey (2000) contend that different types of knowledge (explicit and tacit) are closely tied to motivations.

Effectiveness of knowledge transfer itself is affected by knowledge-related, motivational, and communication related factors (Ko, Kirsch, & King, 2005) and the frequent use of various mechanisms together in-combination rather than relying on the single use of a single mechanism (Slaughter & Kirsch, 2006). Very little research in this area has focused on the impact of knowledge transfer mechanisms on the learning outcomes specifically impact on the processes by which software testers acquire and/or improve their existing skill set.

2.2. Software testing

Review of extant literature on software testing lifecycle (STLC) identifies various software testing activities and ways in which these activities can be carried out in conjunction with the software development process. This literature also identifies various skills that software testers need to possess in order to perform activities effectively in a given phase of STLC. Similar to SDLC, STLC also suggests the phases of analysis, design, implementation, execution, and evaluation in software testing lifecycle. The V-model, which is the most popular testing model, provides a basis for the identification of various testing activities. Based on the V-model, Vijay (2001) and Waligora and Coon (1996) suggest the need to conduct testing in parallel with many of the SDLC phases so that testing efforts in later stages can be minimized.

Table 1 below summarizes testing activities and skills needed by the phases of STLC. It provides us with a preliminary channel to identify testing activities during which various stakeholders in the software development and testing could potentially interact. We intend to use these activities to guide the development of our preliminary interview questionnaire.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Activities</th>
<th>Skills needed</th>
</tr>
</thead>
</table>
| Analysis | • Define scope and strategy for testing.  
• Refine specification. | • Read and understand a specification  
• Identify deficiencies in a specification  
• Convert specification to a decision table  
• Devise a testing strategy based on the specification |
| Design | • Derive test cases.  
• Organize test effort. | • Develop an adequate set of functional and non-functional test cases.  
• Identify redundant test cases  
• Develop boundary test cases  
• Write a test procedure/script |
### Implementation
- Develop machinery needed to conduct tests.
- Develop test drivers
- Create test data sets

### Execution
- Run test as controlled experiment.
- Capture test results
- Perform testing according to test procedure
- Document test results
- Use test drivers

### Evaluation
- Verify results against expectations.
- Take follow-up action such as reporting, debugging.
- Identify program errors from test results
- Debug program given description of bug
- Debug test drivers/ procedures

## 3. Research Methodology

We take a two-phased approach in accomplishing the goals of this research. We employ qualitative and quantitative research methodologies sequentially. In phase 1, we use case study methodology and qualitative data analysis techniques in order to identify knowledge transfer mechanisms. With the use of survey research methodology in phase 2, we intend to triangulate our findings from our case study in phase 1. This paper reports our preliminary findings from phase 1. Our data gathering phase began with interviews with appropriate testing personnel, in a large Fortune 500 organization, TransCo, to identify and better understand the issues related to software testing knowledge transfer in software testing. We also interviewed stakeholders from the offshore testing vendor, VendorCo, utilized by TransCo.

1. Interviews included a numerous open-ended questions focusing on understanding the testing process followed, the training processes currently used for software testers, and existing knowledge management initiatives;
2. Our key informants included testing process stakeholders such as testing managers and test leads from TransCo as well as VendorCo. In total, we interviewed 5 informants (3 TransCo and 2 VendorCo employees)
3. Interviews were conducted in December 2006 and February 2007.

Follow-up interviews will be conducted to clarify initial findings and to expand the coverage, as needed. After this initial data gathering phase, we expect to develop an initial model of knowledge transfer mechanisms, factors influencing these mechanisms, and their effectiveness on skills improvement and acquisition.

## 4. Data Analysis and Results

Our initial analysis of transcripts of interviews of 5 software testing stakeholders identified several interesting topics and issues that need to be investigated further. In the following paragraphs we discuss these findings.

### 4.1. Knowledge Partitioning

One of the major problems facing TransCo in the area of knowledge management is that existing knowledge is fragmented and spread across its various business units. Adding to this problem is the fact that in many occasions this fragmented knowledge is inconsistent. We identified following two major knowledge partitions:

1. Across multiple projects
2. Between TransCo and VendorCo.

We discuss each of these partitions in detail next.
4.1.1. Across multiple projects. Within a single project we observed that not everyone will be aware of all the aspects of that project. Thus, many times team members, especially new team members, may end up spending time to figure out how to perform certain tasks. Such time is usually spent in locating know-how knowledge for performing given tasks and identifying and locating other stakeholders who may possess such knowledge.

Most software testing even for a single software release involves dependencies with the other systems. Such circumstances have become more commonplace as the systems begin operating in an increasingly distributed environment. In such environments one would need to know about other systems to which system-under-consideration now interfaces to and the nature of those dependencies. Following quotes illustrates these circumstances:

“"It was very consistent (before); we could test point of sale form end to end without worry about if anyone else was doing anything. But now we have moved to a distributed environment so that the call the POS makes now go through the various distributed system, one of them is our rating system, well without getting a rate and now you are at the POS. So right now when we do testing you are dependent on that rating system being up, to be able to test something upstream that has functionality that we want to test...and the people that are dealing with that don’t know at all what is going on with the people in the rates side or what I would call the revenue side. Then you have another big area called shipment, all of the scanning and movement of packages and tracking all of that. Well they don’t know what each other are doing and the testing processes are different, so if I take this testing system down because of normal maintenance or I am putting patches in or whatever, you might affect something over here on automation.”

“So the first thing that came to mind is, it would be helpful to understand how these things are playing and how they interact so people would know, I need to be thinking about rating even though I am a dot.com guy. I need to understand revenue... It would be great to have them [people] be able to see on smaller level how things work together.”

Another factor that aggravates this knowledge partition further is that knowledge about such systems is (if) documented is typically very inaccessible to someone outside the groups. This is not necessarily because people do not want to share, but primarily because such documentation may not be a standardized format, or not available in one central location, or is not documented in a way that another person or group will be able to comprehend.

4.1.2. Between TransCo and VendorCo. Second knowledge partition comes into play with the existence of offshore testing vendor companies employed by TransCo. Given that many of TransCo’s systems are poorly documented, these vendor companies end up spending significant amount of time understanding how such systems work together and what are the application dependencies, in short doing some sort of reverse engineering. In the due process, these vendor companies develop and maintain documentation of these systems. Primary goal of developing such documentation is to train the new team members in VendorCo’s testing team especially in the light of high employee turnover in the Indian software industry and the need to quickly assimilate new team members in the team.

Technically even though such system knowledge documentation belongs to TransCo, TransCo does not have systematic way of incorporating that knowledge into its projects and use it to reduce time-to-test on future projects. Another major concern here is similar to the one pointed out in the discussion of previous knowledge partition, specifically,
unawareness of knowledge availability for those outside that particular relationship.

“…they have to do that because the turnover rate is so high relative to here. And so they are very good at it and they capture a lot of information that I would say that we don’t have but it is our information and I think that tapping into that would be very helpful.”

“We have off shore vendors that we do a lot of work with and contract with them on. They have either web based or other tools that they use to capture information about what we do and they use it for support that they are providing to us. But that is for that vendor. We own all of that data and we can access it anytime we want, but the processes to do that in a systematic way are not there. And I doubt that anybody outside of that specific relationship would even know that that was available.”

Given the existence of these two knowledge partitions, we need to identify guidelines that will allow for smooth knowledge sharing and transfer practices among the involved parties. We are currently working to identify such guidelines. We are also in the process further analyzing the qualitative data collected from interviews to identify other emergent themes that might provide clues on how such transfer should take place.

5. Potential Contributions and Conclusion

The potential contributions of this research are:
- How knowledge partitioning affects knowledge transfer process.
- Identification of factors that influence effectiveness of KT mechanism
- Development of guidelines that can help organizations in designing programs for allowing test employees to acquire the skills necessary to be effective in their work through such knowledge transfer.

It is important to keep the following points in mind: First, the interview data represent the perceptions of those interviewed. The interviewees typically respond in accordance with their view of the overall organizational system, even though this view is often limited in scope and in appreciation for overall organizational complexity. Nonetheless, the data suggest the interviewees’ view of “reality.” Second, while some issues uncovered may be cause for concern, they should also be viewed as opportunities for future improvement efforts.

6. References


