Outsourcing and Knowledge Management in Software Testing

Katja Karhu, Ossi Taipale, Kari Smolander
Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing
PL 20, 53851 Lappeenranta, Finland
katja.karhu@lut.fi, ossi.taipale@lut.fi, kari.smolander@lut.fi

The objective of this empirical study was to explore outsourcing in software testing and shape hypotheses that explain the association between outsourcing and knowledge management. First, a survey of testing practices was conducted and 26 organizational units (OUs) were interviewed. From this sample, five OUs were further selected for an in-depth case study. The study used qualitative grounded theory as its research method and the data was collected from 41 theme-based interviews. The analysis yielded hypotheses that included that the business orientation of an OU affects outsourcing of testing and the knowledge management strategy, outsourcing seems to be more effective when independent testing agencies have enough domain knowledge, and outsourcing verification tasks is more difficult than outsourcing validation tasks. The results of this study can be used in developing the knowledge management strategy and as guidance in making outsourcing decisions.

Keywords: software testing, outsourcing, knowledge management, grounded theory, case study.

1. INTRODUCTION

The costs of testing of a software project or product are considerable. Kit (Kit, 1995) states that the systems we build are even more complex and critical, and more than 50 % of the development effort is frequently spent on testing. In our earlier study (Taipale et al., 2006a), outsourcing of software testing was evaluated as one of the items that increase the efficiency and reduce the costs in software testing. In the same testing survey (Taipale et al., 2006a), the respondents evaluated personnel costs as the highest cost item (29 OUs of 30) in software testing. The testing automation costs were evaluated as the second highest cost item (27 OUs of 29). Because outsourcing in software testing is associated with the both cost items, we decided to observe it more closely.

The literature of software testing expresses both advantages and disadvantages of outsourcing (Kaner et al., 1999). Independent software testing agencies are commissioned by organizations to evaluate their software products. In the literature the benefits of independent testing agencies are emphasized, because it is believed that product reliability will be better if testing is done by a fully independent testing agency (Kaner et al., 1999). They are seen as to provide a neutral perspective on the system to be tested. On the other hand, Kaner et al. (1999) have listed numerous disadvantages of using independent testing agencies. Many of the disadvantages also apply to an in-house testing organization as well. The main problem concerning knowledge management when working with independent testing agencies is that an independent testing agency does not have access to all the artifacts, such as design documents and source code, and has a limited amount of product knowledge.

Knowledge is viewed as a source of competitive advantage for organizations (Argote and Ingram, 2000, Spender and Grant, 1996). Outsourcing necessitates making knowledge explicit (Osterloh and Frey, 2000). Foray (2004) notes, that when knowledge is codified it becomes transferable, independently of people in whom tacit knowledge is embedded. Tacit knowledge is knowledge which is not or can not be articulated and is usually embedded in the employees. According to Nonaka (1994), if the knowledge has tacit components, the knowledge transfer may require numerous individual exchanges. Explicit knowledge is codified and transferable in artifacts, such as documents. To effectively acquire knowledge and utilize the knowledge created in an organization, a suitable knowledge management strategy is needed.

According to Hansen et al. (1999) there are two primary strategies for knowledge management: the codification strategy and the personalization strategy. In the codification strategy knowledge is carefully codified and stored in databases, where it can be accessed and used easily by anyone in the company. In the personalization strategy the knowledge is tacit, closely tied to the person who developed it, and is shared mainly through direct person-to-person contacts.

The organizational knowledge, which in this case is testing knowledge (such as test cases, user documents, the product to be tested, as well as the results of the tests), has to be transferred between organizations. According to Szulanski (1996), the transfer of organizational knowledge can be quite difficult to achieve. This is because knowledge resides in organizational members, tools, tasks, and their sub networks and, as Nonaka (1995) shows, much of the knowledge in organizations is tacit or hard to articulate.

The physical distance between development and testing may create new challenges for knowledge transfer. Cohen et al. (2004) found in exploring the organizational structure that testers and developers ought to co-locate. When testers and developers worked in separate locations, communication, as well as personal relationships, was impaired, unlike when both groups worked in close proximity.

In this study, our specific objective is to understand the practice of outsourcing in software testing, and based on that understanding, derive hypotheses for testing outsourcing and knowledge management. The grounded theory research method outlined by Glaser and Strauss (1967) and later extended by Strauss and Corbin (1990) was used. Grounded theory was selected because of its ability to uncover the issues from the practice under observation that may not have been identified in earlier literature (Glaser and Strauss, 1967).

The interviewees in this study represented companies that produce and/or test automation or telecommunication software. Their products can be described as mission critical and they are used in real-time environments. The study included four theme-based interview rounds. We personally visited companies and carried out 41 tape-recorded interviews.

The paper is structured as follows: The research process and the grounded theory method are described in section 2. The analysis results are presented in section 3. Finally, the discussion and conclusions are given in section 4.

2. RESEARCH PROCESS

According to Seaman (1999), a grounded approach enables the identification of new theories and concepts, making it a valid choice for software engineering research. We followed the process of building a theory from case study research described by Eisenhardt (1989) and its implementation example for information technology (Paré and Elam, 1997). Principles for an interpretive field study were derived from Klein and Myers (1999). Other example studies included studies by Carter and Dresner (2001) and Smolander et al. (2005).

2.1 Selecting Case OUs

The population of this study consisted of organizational units (OUs). The standard ISO/IEC 15504-1 (ISO/IEC, 2002) specifies an OU as a part of an organization that is the subject of an assessment. An OU deploys one process or more having a coherent process context and operating within a coherent set of business goals. An OU is typically a part of a larger organization, although in a small organization, the OU may be the whole organization.

The reason to use an OU as an assessment unit was that we wanted to normalize the effect of the company size to get comparable data. The population consisted of OUs from small, nationally operating companies to large internationally operating ones. Based on the available information of the OUs, we estimated the criticality of their produced or tested software. The objective was to select OUs with software criticality above average (Taipale et al., 2006c) to ensure that the process contents would be comparable. The criticality of the produced or tested software was measured during the first interview round. Criticality of the products was estimated by asking how severe problems the faults in their products can cause (Taipale et al., 2006c).

The initial sample consisted of 26 OUs. From this first sample, five OUs were selected as the case OUs. The sampling was theoretical (Paré and Elam, 1997) and the cases were chosen to provide examples of polar types (Eisenhardt, 1989). The goal of theoretical sampling is not the same as with the probabilistic sampling. The researcher's goal is not the representative sample of all possible variations, but to gain a deeper understanding of analyzed cases and identify concepts and their relationships. The case OUs are described in Table 1.

TABLE 1: Description of the interviewed OUs.

OUs	Business	Company size[1] / Operation
All 26 OUs including cases from A to E	Automation or telecommunication domain	The average size of an OU was 75 persons.
Case A	MES producer and integrator	Large / International
Case B	Software producer and independent testing agency	Small / National
Case C	Process automation and information management provider	Large / International
Case D	Electronics manufacturer	Large / International
Case E	Independent testing agency	Small / National

[1] SME Definition (EU, 2003)

2.2 Data Collection

The data collection consisted of four interview rounds. The interview rounds and the roles of interviewees in the case OUs are described in Table 2. Snowball sampling was applied in arranging the interviews. This meant in practice that an interviewee recommended the next interviewee. To reduce bias caused by the researchers, the interviews were conducted by two researchers.

The first interview round was made in all 26 OUs and it contained both structured and semi-structured questions. The objective of this interview round was to understand the basic practice of testing, identify case OUs (representative polar points) for the next rounds, and identify problems and improvement proposals for testing. The interviewees were managers of development or testing or both. In some interviews there was more than one interviewee present, e.g. a manager of development and a manager of testing. Such interviews usually lasted more than one hour. The questions of the first round concerned general information on the OU, processes, knowledge transfer between development and testing processes, and the development environment of the OU.

The interview rounds 2 to 4 were made in the five case OUs. The interviewees of the second round were managers of testing. In some interviews, managers of development were also present. The duration of the interviews varied between one and one and half hours. The objective of the second interview round was to achieve a deeper understanding of software testing in practice. The questions were semi-structured and concerned problems in testing, the use of software components, the influence of the business orientation, knowledge transfer, schedules, organizations and knowledge, testing automation, and economy.

The interviewees of the third round were testers and the interviewees of the fourth round were systems analysts. The interviews in these rounds were also semi-structured and concerned the work of the interviewees, problems in testing (e.g. increasing complexity of the systems), the use of software components, the influence of the business orientation, knowledge transfer, schedules, organizations and knowledge, and testing automation. The interviews lasted about one hour.

The themes between the interview rounds remained similar, but the questions evolved from general to detailed. Before proceeding to the next interview round, all interviews were scripted and analyzed because new ideas emerged during the data analysis. These new ideas were reflected in the next interview rounds. The data collection process of all 41 interviews generated a transcription of 946 pages. The themes for each of the interview rounds are available at http://www.it.lut.fi/project/anti/.

2.3 Data Analysis

The grounded theory method contains three data analysis steps: open coding, where categories of the study are extracted from the data; axial coding, where connections between the categories are identified; and selective coding, where the core category is identified and described (Strauss and Corbin, 1990). In practice, these steps overlap and merge because the theory development process proceeds iteratively.

TABLE 2: Interviewees and interview rounds.

OU / Round	Case A	Case B	Case C	Case D	Case E
1) Structured/Semi-structured	R&D and testing manager	Testing manager	Testing manager R&D manager	Testing manager R&D manager	Testing manager
2) Semi-structured	R&D and testing manager Testing manager	Testing manager	Testing manager	Testing manager R&D manager	Testing manager
3) Semi-structured	Tester	Tester	Tester	Tester	Tester
4) Semi-structured	Systems analyst	Systems analyst	Systems analyst	Systems analyst	Systems analyst

The objective of the open coding was to classify the data into categories and identify leads in the data. The process of grouping concepts that seem to pertain to the same phenomena is called categorizing and it is done to reduce the number of units to work with (Strauss and Corbin, 1990). The open coding of the interviews was done using the ATLAS.ti -software (2005). The open coding process started with “seed categories” (Miles and Huberman, 1994) that contained the essential stakeholders, phenomena, and problems. Seaman (1999) notes that the initial set of codes (seed categories) comes from the goals of the study, the research questions, and predefined variables of interest. In the open coding, new categories appeared and existing categories were merged because especially in the beginning of the analysis new information sprang up.

The objective of the axial coding was to further develop categories, their properties and dimensions, and causal conditions or any kind of connections between the categories. The dimensions of a category represent the locations of the property or the attribute of a category along a continuum (Strauss and Corbin, 1990). Each phenomenon represented by a category was given a conceptual name (Strauss and Corbin, 1990). The created categories contained the business orientation of the OU, outsourcing activity, involvement of testing in the development process, knowledge transfer between development and testing, knowledge management strategy, and domain knowledge. Our inductive data analysis of the categories included Within-Case Analysis and Cross-Case-Analysis, as specified by Eisernhardt (1989). We used the tactic of selecting dimensions and properties, and looking for within-group similarities coupled with inter-group differences (Eisenhardt, 1989). Each chain of evidence in this interpretation was established by having sufficient citations in the case transcriptions.

The objective of the selective coding was to identify the core category (Strauss and Corbin, 1990), a central phenomenon, systematically relate it to other categories, and generate the theory. Strauss and Corbin (1990) write that sometimes the core category is one of the existing categories; at other times no single category is broad enough to cover the central phenomenon. In that case the central phenomenon must be given a name. In this study the core category was outsourcing in software testing.

The general rule in grounded theory is to sample until theoretical saturation is reached. This means, until (1) no new or relevant data seem to emerge regarding a category; (2) the category development is dense, insofar as all of the paradigm elements are accounted for, along with variation and process; (3) the relationships between categories are well established and validated (Strauss and Corbin, 1990).

The data was analyzed after each interview round to collect new issues, and to see if it was worthwhile to continue the data collection procedure. The theoretical saturation was reached during the fourth interview round, because new categories did no more appear, categories were no more merged, shared, or removed, attributes or attribute values of the categories did not change, and relationships between categories were considered stabile i.e. the already described phenomena begun to repeat in the data.

To ensure the validity of the study, four researchers took part in the data analysis. The bias caused by researchers was reduced by combining the different views of the researchers (observer triangulation) (Denzin, 1978).

3. RESULTS

In the following the results of the analysis are discussed. The cases and their characteristics were depicted by the dimensions that explain the similarities and the differences between the case OUs. The observed categories and their dimensions are presented in Table 3.

TABLE 3: Categories and their dimensions for the case OUs

OU Category	Case A	Case B	Case C	Case D	Case E
Business orientation	50 % services, 50 % products	75 % services, 25% products	66 % services, 33% products	100 % products	100% services
Outsourcing activity	Low	Offers testing services	Low	High	Offers testing services
Involvement of testing	Medium	Late	Medium	Medium	Late
Knowledge transfer	Partly informal and flexible.	Partly informal and flexible.	Partly informal and flexible.	Formal, planned, and transparent.	Planned and flexible.
Knowledge management strategy	More personalization.	More codification.	More personalization.	More codification.	Codification and personalization.
Domain knowledge	High	Low	High	Low	Low

Business orientation described the ratio between products and services of the OU's turnover. The construct was derived from Sommerville (1995) who divides software products into two broad classes, generic products and customized products. Generic products are stand-alone systems produced by a development organization and sold on an open market to any customer who is able to buy them. Customized products are systems commissioned by a particular customer and developed specially for that customer by a contractor. We complemented and widened Sommerville's (1995) classification with a finer granulation and added a purely service-oriented dimension (consulting and subcontracting) at the other end (see Figure 1).

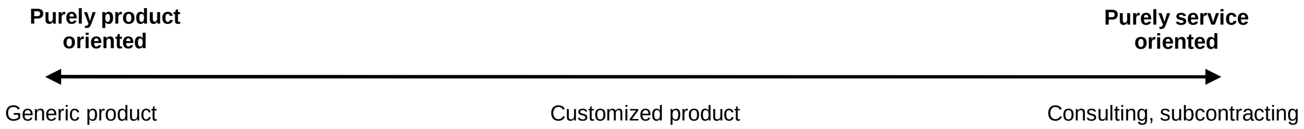

FIGURE 1. Business orientation.

Outsourcing activity describes how actively an OU either used or offered testing services. The involvement of testing defines how early or late testing became involved in the software development process. The knowledge transfer between development and testing processes is depicted with the knowledge transfer dimension. The knowledge management strategy describes whether an OU was more inclined towards the personalization or the codification strategy. The amount of domain knowledge that was needed in testing is described by the domain knowledge dimension.

3.1 Description of the Case OUs

Case A: A MES Producer and Integrator
In Case A, products represented 50% and services 50% of the turnover of the OU. Case A developed and tested manufacturing execution systems (MES). Testing was partly integrated with software development, but system testing was organized as a part of the quality assurance organization. Using outsourced testing resources had been considered, but not yet implemented. Close co-operation between development and testing was emphasized. The involvement of testing in requirements specification was complicated because the customers included companies with different in-house standards that caused special requirements complicating the requirements testing.

Knowledge transfer between development and testing processes was flexible because most of the developers and testers worked physically close to each other. The effect of distance in knowledge transfer was growing because Case A had founded a new development center. The distance between testing and development locations in this new situation was recognized as a barrier to knowledge transfer. Codification of knowledge was not emphasized earlier because developers and testers co-located, but the situation was changing because of the new development center. The need for formal documentation and communication was emphasized in knowledge management. Problems of knowledge management included that documentation was not up to date, troubleshooting documentation was inaccurate, and the provided schedule and release information from the testing organization to development was insufficient.

Domain knowledge was emphasized in testing, because testing customized MES systems required know-how of their customers' systems and application area.

Case B: Software Producer and Independent Testing Agency
Case B was mostly a service oriented OU. Products represented 25% and services 75% of its turnover. Case B tested its own software products and offered testing services to external customers as well. Testing was partly integrated with software development and partly adapted to the processes of external software development. Case B offered testing services mainly to a large internationally operating organization which manufactured high technology products.

In providing testing services knowledge transfer depended on planned and formal meetings, as well as on codified knowledge. The codification strategy was more emphasized because e.g. test plans, test runs, and test results were reported using the templates of the customer's quality system. This in essence meant that the knowledge transfer was controlled by the customer.

Domain knowledge was more emphasized when testing the company's own products and less emphasized when testing external software. Problems of knowledge management included a lack of codified knowledge and finding the right balance between personalized and codified knowledge.

Case C: Process Automation and Information Management Provider
In Case C, services represented two thirds and products one third of the turnover of the OU. The testing resources were partly integrated with software development. System testing formed a centralized testing organization with a testing technology center. Case C did not use outsourced testing resources. Close co-operation between development and testing was seen important.

The testing resources were involved in development often before system testing. The early involvement of testing also caused problems because developers could not give enough advice to the testers. Knowledge transfer was seen flexible because testers and developers were able to communicate face-to-face. The effect of distance was low, but it was growing because of a new development center. The distance between development and testing, and the fact that testers do not understand the technical language of the developers were found as barriers to the knowledge transfer. The codification of the knowledge was less emphasized, but it was growing in importance because knowledge had to be transferable for the new development center.

Domain knowledge was emphasized in testing, because testing customized automation and information management systems required know-how of their customers' systems and application area.

Case D - Electronics Manufacturer
Case D was a purely product oriented OU. High product orientation required high-quality end-products, because recalls in serial production are very expensive. The tasks of testing were clear and they were carefully specified. Case D concentrated on the core business and outsourced testing resources to a great extent.

Re-testing increased the amount of testing, and avoiding duplicate testing seemed to be important for Case D. Other factors that affected knowledge transfer were not of as high an importance. Testing was involved in development rather late, but it was informed of the upcoming testing tasks. Case D emphasized the measurement function of testing and tried to avoid the testing of unfinished products. Testing worked as a separate function and was not integrated in product development. Knowledge transfer between development and testing was planned, formal, and transparent as specified by the quality system. The effect of distance was medium because development centers were located around the world, but knowledge transfer was supported by the available technology and the quality system, which emphasized the codification of knowledge.

Domain knowledge was less emphasized, because testing tasks consisted of measurements, that were similar for most of the software releases.

Case E - Independent Testing Agency
Case E was a purely service oriented OU offering testing services to external customers. Its customer base was more heterogeneous than the customers of Case B. Working as an external testing organization required the adaptation to the processes of the customers. Early involvement of testing was fundamental for the OU because it also offered assistance in the planning of testing.

The knowledge transfer between customer's software development and the independent testing agency was mostly described as being active and clear. Knowledge transfer was specified as the task of the contact persons.

The effect of distance was significant because of the work as an external testing organization. We observed that in Case E also the knowledge transfer between customers and the OU was controlled by customers, and it further emphasized codified knowledge. Codification of the knowledge was emphasized, but the level of the customer documents was not always satisfactory. Domain knowledge was slightly emphasized, because it was seen as a competitive advantage.

3.2 Summary and Hypotheses

The summary of the factors that affect outsourcing and knowledge management in software testing is expressed in Table 4. The resulting hypotheses were shaped according to Eisenhardt (1989). The central idea is that researchers constantly compare theory and data iterating toward a theory which closely fits the data.

TABLE 4: Factors affecting outsourcing and knowledge management in case OUs

OU Category	Case A	Case B	Case C	Case D	Case E
Business orientation	Customized products based on uniform product kernels.	Independent testing agency and software producer.	Customized products based on uniform product kernels.	Generic products.	Independent testing agency.
Outsourcing activity	Case A has considered testing outsourcing, but not yet used.	Offers testing services.	Case C does not use outsourced testing resources.	Case D concentrates on core business. Testing is outsourced to a great extent.	Offers testing services.
Involvement of testing	Strives to earlier involvement.	Customer does not prefer earlier involvement.	Strives to earlier involvement.	Avoided testing unfinished products, but testing was informed of upcoming testing tasks.	Some customers would prefer earlier involvement, but it has not yet been possible.
Knowledge transfer	Testers and developers co-locate → hinders outsourcing	Through documentation and meetings. Mainly controlled by the customer.	Testers and developers co-locate→ hinders outsourcing	Knowledge transfer is formal → enables outsourcing	Through contact persons. Slightly different protocols with different customers. Mainly controlled by the customers.
Knowledge management strategy	Lack of codified knowledge → hinders outsourcing	Adaptation to customer's codification strategy.	Lack of codified knowledge → hinders outsourcing	Codification procedures clearly defined by the quality system → enables outsourcing	Adaptation to each customer's codification strategy.
Domain knowledge	Testing MES systems requires domain knowledge → hinders outsourcing	Testing activities require only general level knowledge of the system to be tested.	Testing of automation and information management systems requires domain knowledge → hinders outsourcing	Testing tasks are clearly specified and do not require specific domain knowledge → enables outsourcing	Knowledge of the customers' software development processes is viewed as more important than domain knowledge.

Hypothesis 1: The business orientation of an OU and the knowledge management strategy affect testing outsourcing.
The product oriented OU D outsourced testing resources to a great extent.

> "During the last year we decided that testing is not our core business. We decided to use outsourced testing resources." – Testing manager, Case D.

In Case A testing outsourcing had been considered, but they had come to the conclusion that it is not possible with their current processes. Case C had not considered outsourcing, because they had a strong in-house testing organization. In Cases A and C also the co-location of developers and testers was considered important.

According to the testing manager of the independent testing agency E, in product oriented OUs the contents of testing is more repetitive, compact and easier to specify (and therefore codify), which enables the outsourcing of testing. Companies making customized systems are not seen as potential customers for the independent testing agencies, because the systems are unique and the testing tasks are not repetitive.

> "In product oriented companies the testing process is more compact and easier to specify. They are more potential customers. Companies making customized systems are not so willing, or should we say potential to outsource their testing." – Testing manager, Case E.

On the other hand, also OUs developing customized products could benefit from hiring testing professionals from independent testing agencies, because they may have more knowledge about planning of testing, as well as new testing methods and tools.

> "We give a clear financial advantage to the customers. They can hire external testing services when they need and not keep the testing resources on payroll all the time. Then they also know that they get testing professionals." -- Testing Manager, Case E

Hypothesis 2: OUs ought to adjust the knowledge management strategy according to the business orientation.
We concluded that product oriented OUs emphasize codification and OUs developing customized systems emphasize personalization. The codification of knowledge is more systematic in product oriented OUs because of e.g. the repetitive development and testing tasks. The knowledge management strategy of independent testing agencies depends on the knowledge management strategy of their customers.

Case D had clearly focused on the codification strategy throughout the entire organization, and it was specified by their quality system. Codification of knowledge was seen to improve the knowledge transfer.

> "Making test results (documents) available organization wide is an efficient way to avoid extra work and costs in development and testing." – Testing manager, Case D.

The case OUs A and C, that developed customized systems, relied more on the personalization strategy. In the OUs A and C there was more face-to-face communication between development and testing, instead of documentation.

> "Identifying a bug together with the tester is easier than trying to explain the same thing in writing or on the phone, especially in a foreign language." – Systems analyst, Case C

However, the significance of codification was growing. The distance between developers and testers affected the codification of knowledge. If developers and testers were not co-located, the need for codification was higher.

> "We have started to develop software in a new subsidiary (in a foreign country). They have to document their work well." – Tester, Case C.

The case OUs B and E followed the codification strategy more consistently, because the interaction with their customers required formal documentation.

Hypothesis 3: Outsourcing seems to be more effective when independent testing agencies have enough domain knowledge.

If testing requires more tacit domain knowledge, which is available only in-house, testing outsourcing might not be beneficial. Testers' domain and product knowledge was more emphasized in service oriented OUs A and C than in product oriented OU D because testing customized systems demanded a broader expertise.

> "Some of our testers have worked as automation designers, and in some testing tasks you have to be an automation designer in order to understand how the system should be tested." – Testing manager, Case C.

Independent testing agencies may not have enough domain knowledge.

> "I think that knowing the processes of the customers is more important than domain knowledge. If our knowledge has not been adequate, then we have turned to the customers to get more expertise." – Systems analyst, Case E.

On the other hand, it is beneficial for an independent testing agency to gain deeper knowledge of the systems in order to have a longer term partnership with their customers.

> "We have to know the customers' systems. We gain a noticeable competitive advantage when we know the domain, the software, and the systems. – When we get more knowledge about the customers' systems, they will be more committed to us." – Testing Manager, Case E

Hypothesis 4: Outsourcing verification tasks is more difficult than outsourcing validation tasks.

Verification ensures that software correctly implements a specific function and it answers to the question: are we building the product right. Basic verification methods include inspections, walkthroughs, and technical reviews. Checklists are used as the tools of the verification.

Validation ensures that software that has been built is traceable to the customer requirements. Validation answers to the question: are we building the right product. Validation consists of (1) developing tests that will determine whether the product satisfies the users' requirements, as stated in the requirements specification and (2) developing tests that will determine whether the product's actual behavior matches the desired behavior as described in the functional design specification.

Verification requires access to the design documents and source code. Independent testing agencies do not have access to all the knowledge concerning the systems which they test. Therefore, validation tasks are easier for the independent testing agencies and less risky for the customers.

> "Our role is just to test and notify our customer which tests did not pass, and give some suggestions on why it happened. We do not know the components so well that we could give detailed information. In a way our role is not even to get that information, but just to do the tests. It is the customer's job to find out the real reason." – Testing manager, Case B.

4. DISCUSSION AND CONCLUSIONS

The objective of this qualitative study was to explore outsourcing in software testing and shape hypotheses that explain the association between outsourcing and knowledge management. The analysis was done using the grounded theory method.

The result of this study consisted of four hypotheses: 1) The business orientation of an OU affects testing outsourcing, 2) OUs ought to adjust the knowledge management strategy according to the business orientation, 3) outsourcing seems to be more effective when independent testing agencies have enough domain knowledge, and 4) outsourcing verification tasks is more difficult than outsourcing validation tasks.

According to this study, the business orientation of an OU affected testing outsourcing. The product oriented OUs seem to have more possibilities of using outsourced testing than the OUs developing customized systems. In the product oriented OUs the development and testing processes seem to be more repetitive, and the need and the scope for outsourcing is more predictable. In OUs developing customized systems, early involvement with software development seems to be more important. According to Blomqvist et al. (2002) outsourcing is preferred when (i) the degrees of uncertainty and complexity are minor and the danger of opportunistic behaviour is small, (ii) there are many potential partners available, and (iii) transactions do not need any specific investments. The contents of testing in product oriented OUs fits better to the requirements expressed by Blomqvist et al. (2002), because the testing tasks are easier to specify.

We found that OUs ought to adjust the knowledge management strategy according to the business orientation. Only the product oriented Case D had clearly focused on a codification strategy throughout the entire organization, because it was specified by their quality system. The case OUs A and C, that developed customized systems, relied more on the personalization model. Hansen et al. (1999) discuss business orientation from the competitive strategy point of view. They divide companies according to their standardized or customized products. Companies that follow a standardized product strategy sell products that do not vary much, if at all. According to Hansen et al. (1999) a knowledge management strategy based on codification fits companies that create standardized products. In companies that sell customized products and services, most of the work goes toward meeting particular customers' unique needs. Because those needs can vary substantially, codified knowledge is of limited value. Companies that follow a customized product approach should therefore consider the personalization model.

Independent testing agencies made an exception. The case OUs B and E followed the codification strategy more consistently, because the interaction with their customers required formal documentation. Hansen et al. (1999) view that the company should focus on only one of the knowledge management strategies. Even though the independent testing agencies were service oriented, their knowledge management strategy was more inclined towards codification. This does not seem to directly fit to the description that companies providing services should focus only on the personalization strategy.

Outsourcing seems to be more effective when independent testing agencies have enough domain knowledge. In Cases A and C domain knowledge was viewed as important in testing. In Case D the testing process was standardized and the need for specific domain knowledge in testing tasks was lower. Independent testing agencies do not necessarily have an adequate amount of domain knowledge. Beath and Walker (1998) state that outsourcing is most effective when the supplier's existing knowledge is adequate to the tasks implied by the project. Domain knowledge is tacit and context-dependent in nature, therefore making it difficult to transfer and codify.

According to our last hypothesis, outsourcing verification tasks seems to be more difficult than outsourcing validation tasks. Independent testing agencies usually focus on system testing, and are not concerned with the software's internal processing (Dustin et al., 1999). System testing consists mainly of validation tasks. Verification tasks require more knowledge transfer between the customer and the independent testing agency because artifacts needed in verification include e.g. source code and design documents. This may pose a potential risk to the customer. Teece (1986) notes that while codified knowledge is easier to transmit and receive, it is more exposed to industrial espionage and the like. Verification is also more closely involved in many stages of software development, and requires close co-operation with several instances inside the organization. It is therefore harder to define verification as an independent and separate function, which can be then given as a task to an independent testing agency.

As a conclusion, outsourcing and knowledge management are very much interconnected. Osterloh and Frey (2000) note that outsourcing necessitates making knowledge explicit. Also when knowledge is codified it becomes transferable, independently of people in whom tacit knowledge is embedded (Foray, 2004). When making decisions about testing outsourcing, also issues related to knowledge management should be taken in to account. Cowan et al. (2000) state that codification will provide high benefits in 1) stable systems characterized by specific requirements of knowledge transfer and communication, such as delocalization and externalization, i.e. outsourcing., 2) stable systems where advances and novelties mainly proceed from recombination, reuse and cumulativeness, i.e. standardized systems, and 3) systems that require extensive memory and retrieval capabilities (e.g. firms and organizations that have long product or process development cycles or high rates of personnel turnover).

To guarantee the validity of the study we used probability sampling when selecting the sample for the first interview round and theoretical sampling when selecting case OUs. Robson (2002) lists three threats to validity in this kind of research: reactivity (the interference of the researcher's presence), researcher bias, and respondent bias and strategies that reduce these threats. To reduce these biases, the interviews were conducted by two researchers and the data was analyzed by four researchers (observer triangulation). We also compared the data (data triangulation) and the results (method triangulation) of this study with our earlier quantitative (Taipale et al., 2006c) and qualitative studies (Taipale and Smolander, 2006).

The limitation of the study is the number of the case OUs. It is obvious that increasing the number of cases could reveal more details. However, our target was not to create a comprehensive list of factors related to outsourcing and knowledge management, but to cover the most important factors from the point of view of our case OUs.

In addition, due to the lack of reporting space, some other relevant findings, such as the influence of testing schedules are not covered here because it is discussed in detail in (Taipale et al., 2006b). The results of this study can be used in developing the knowledge management strategy and as guidance in making outsourcing decisions.

REFERENCES

ATLAS.ti - The Knowledge Workbench. (2005). Scientific Software Development.
Argote, L. and Ingram, P. (2000) Knowledge Transfer: A Basis for Competitive Advantage in Firms. *Organizational Behavior and Human Decision Processes.* **82,** 150-169.
Beath, C. M. and Walker, G. (1998) Outsourcing of Application Software: A Knowledge Management Perspective. Thirty-First Annual Hawaii International Conference on System Sciences. IEEE Comput. Soc, Los Alamitos, CA, USA.
Blomqvist, K., Kyläheiko, K. and Virolainen, V.-M. (2002) Filling a gap in traditional transaction cost economics: Towards transaction benefits-based analysis. *International Journal of Production Economics.* **79,** 1-14.
Carter, C. R. and Dresner, M. (2001) Purchasing's role in environmental management: Cross-functional development of grounded theory. *Journal of Supply Chain Management.* **37,** 12-28.
Cohen, C. F., Birkin, S. J., Garfield, M. J. and Webb, H. W. (2004) Managing Conflict in Software Testing. *Communications of the ACM.* **47**.
Cowan, R., David, P. A. and Foray, D. (2000) The Explicit Economics of Knowledge Codification and Tacitness. *Industrial and Corporate Change.* **9,** 211-253.
Denzin, N. K. (1978) *The research act: A theoretical introduction to sociological methods.* McGraw-Hill.
Dustin, E., Rashka, J. and Paul, J. (1999) *Automated software testing: introduction, management, and performance.* Addison-Wesley, Boston.
Eisenhardt, K. M. (1989) Building Theories from Case Study Research. *Academy of Management Review.* **14,** 532-550.
EU (2003) European Commission.
Foray, D. (2004) *Economics of Knowledge.* The MIT Press, Cambridge, MA.
Glaser, B. and Strauss, A. L. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research.* Aldine, Chicago.
Hansen, M. T., Nohria, N. and Tierney, T. (1999) What's Your Strategy for Managing Knowledge? *Harvard Business Review.* **77,** 106-116.
ISO/IEC. (2002) *ISO/IEC 15504-1, Information Technology - Process Assessment - Part 1: Concepts and Vocabulary.*
Kaner, C., Falk, J. and Nguyen, H. Q. (1999) *Testing Computer Software.* Wiley, New York.
Kit, E. (1995) *Software Testing in the Real World: Improving the Process.* Addison-Wesley, Reading, MA.
Klein, H. K. and Myers, M. D. (1999) A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly.* **23,** 67-94.
Miles, M. B. and Huberman, A. M. (1994) *Qualitative Data Analysis.* SAGE Publications, Thousand Oaks, CA.
Nonaka, I. (1994) A Dynamic Theory of Organizational Knowledge Creation. *Organization Science.* **5,** 14-37.

Nonaka, I. and Takeuchi, H. (1995) *The Knowledge-Creating Company.* Oxford University Press, New York.

Osterloh, M. and Frey, B. S. (2000) Motivation, Knowledge Transfer, and Organizational Forms. *Organization Science.* **11,** 538-550.

Paré, G. and Elam, J. J. (1997) Using Case Study Research to Build Theories of IT Implementation. the IFIP TC8 WG International Conference on Information Systems and Qualitative Research, Philadelphia, USA. Chapman & Hall.

Robson, C. (2002) *Real World Research, Second Edition.* Blackwell Publishing.

Seaman, C. B. (1999) Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering.* **25,** 557-572.

Smolander, K., Rossi, M. and Purao, S. (2005) Going beyond the Blueprint: Unraveling the Complex Reality of Software Architectures. the 13th European Conference on Information Systems: Information Systems in a Rapidly Changing Economy, Ragensburg, Germany.

Sommerville, I. (1995) *Software Engineering.* Addison Wesley, Essex, England.

Spender, J.-C. and Grant, R. M. (1996) Knowledge and the Firm: Overview. *Strategic Management Journal.* **17,** 5-9.

Strauss, A. and Corbin, J. (1990) *Basics of Qualitative Research: Grounded Theory Procedures and Techniques.* SAGE Publications, Newbury Park, CA.

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

Taipale, O. and Smolander, K. (2006) Improving Software Testing by Observing Causes, Effects, and Associations from Practice. the International Symposium on Empirical Software Engineering, Rio de Janeiro, Brazil.

Taipale, O., Smolander, K. and Kälviäinen, H. (2006a) Cost Reduction and Quality Improvement in Software Testing. Software Quality Management Conference, Southampton, UK.

Taipale, O., Smolander, K. and Kälviäinen, H. (2006b) Factors Affecting Software Testing Time Schedule. the Australian Software Engineering Conference, Sydney. IEEE Comput. Soc, Los Alamitos, CA, USA.

Taipale, O., Smolander, K. and Kälviäinen, H. (2006c) A Survey on Software Testing. 6th International SPICE Conference on Software Process Improvement and Capability dEtermination (SPICE'2006), Luxembourg.

Teece, D. J. (1986) Profiting from technological innovation: Implications for integration, collaboration, licencing, and public policy. *Research Policy.* **15,** 285-305.