TOWARDS A KNOWLEDGE TRANSFER MEASUREMENT FOR SOFTWARE REQUIREMENTS

José Jairo CAMACHO
Universidad Nacional de Colombia
Bogotá – Colombia
jjcamachov@unal.edu.co

Jenny Marcela SANCHEZ-TORRES
Universidad Nacional de Colombia
Bogotá – Colombia
jmsanchezt@una.edu.co

Ernesto GALVIS-LISTA
Universidad del Magdalena
Santa Marta – Colombia
egalvis@unimagdalena.edu.co

There has been an increasing interest about knowledge transfer (KT) in software engineering last years, but, less in software requirements (SR) and even less in KT measurement. The purpose of this paper is to make an approach to KT measurement for SR. A mapping from the KT process steps against the software requirements process steps was made, looking for a customized KT process according SR particularities then an approach to metrics were defined for each step. Classic SR metrics are both quantitative and qualitative, none of them related directly with knowledge transfer but with knowledge codification and knowledge sharing. This paper presents the SR process as a KT process obtaining KT oriented FACTORS in one approach to KT measurement in SR.

Keywords-component; Knowledge Transfer Process; Knowledge Management; Software Engineering; Software Requirements; Software Metrics

I. INTRODUCTION

Software Engineering has been recognized as a knowledge intensive application discipline(Rus & Lindvall 2002),(Dingsøyr et al. 2009) and (Ward & Aurum 2004). For this reason, in the last decade there has been an increasing interest about knowledge management in software engineering. In particular, the processes of knowledge codification and knowledge sharing have received most attention and they have been researched in diverse ways.

Research done about Knowledge Transfer, KT, in software engineering had been more related to the handling of software knowledge along and among software development organizations, focusing in factors affecting knowledge transfer and the levels of the transfer inside software organizations (i.e. KT between multinationals, projects, teams and people).

Even when KT in software engineering has been studied, there is a lack of research about KT throughout the detailed sub process of software engineering process, which could be seen according SWEBOOK as: software requirements, SR, software development, testing and maintenance. Since SR is the first sub process at the beginning of software projects, the goal of this paper is to describe how the KT happens in SR and it is done mapping KT and SR elements in a matrix. SR is represented as software elicitation, analysis, specification and validation from the SWEBOOK point of view which was elected due to their effort to summarize what is known about the software process. Finally, we want to gain insights on how KT process could be measured, so indicators are proposed for each mapped process step. More specifically, the research questions for this study are:

1. How does the KT take place in SR?
2. Which are factors affecting KT in each stage of SR?

In developing this paper, we start with a background about KT in section II. Next in section III the mapping method is described. Section IV presents the results of the mapping and metrics proposed so research question are resolved. And finally, in section VI the conclusions of this study are presented.

II. ATHEORETICAL BACKGROUND ON KNOWLEDGE TRANSFER

A. Knowledge Transfer concept

On the one hand, knowledge has been defined as the information and experience grouped usefully in some context(Alavi & E. Leidner 2001), and literature shows a consensus about the taxonomy which represents knowledge as tacit and explicit(Nonaka 2007). On the other hand, transfer means to pass an element form one side to another(Watson & Hewett 2006) and(Borgatti & Cross 2003), so, knowledge
transfer means to pass useful information and experience from one context (project) to another (inside or outside of an organization).

Nevertheless, such transfer, according to some authors, cannot be done (Krogh 2003) due to the fact that knowledge is personal and unique. Every time knowledge passes from tacit to explicit, new knowledge is generated so it is different from the previous one (Garavelli et al. 2002). In this way, the exactly KT cannot be possible.

It should be noted that KT is different from knowledge sharing (Kumar & Ganesh 2009) (Argote & Ingram 2000), since the fact that a person shares knowledge does not mean that he/she already did a transfer. Consequently, entity A (person, business unit or company) transfers knowledge to entity B, just when B is able to apply it in a useful way in its own context. By the same token, it can be said that only sharing knowledge has occurred.

Knowledge sharing is important as a KT enabler, but sharing alone is not enough to make transfer occurs. This is remarkable because, until now, the greatest advances in knowledge management applied to software engineering have been done at the level of knowledge sharing using knowledge codification (Garavelli et al. 2002), (Gosain 2007), (Farenhorst & Vliet 2009) and (Souza et al. 2010).

KT is more than mere codification because it demands more than building “knowledge” bases (data and information) (Kumar & Ganesh 2009). Those bases ended being only data repositories because those bases are used just to code the knowledge, however KT is related to a human process and it could only be generated through cognitive process inside people’s mind (Carayannis 1999).

B. Knowledge Transfer as a Process of Knowledge Management

KT is one of the most important processes for knowledge management (Kuhn & Abecker 1997), their activities are mainly three, gather the knowledge from a source, code it through a channel, and pass it to a receipt (Albino et al. 2004). KT inside the knowledge management could be seen as a final process, because after create, store and share the knowledge, only when transfer occurs knowledge management makes sense and could be said that is useful (Argote & Ingram 2000) and (Kumar & Ganesh 2009).

KT process could be depicted as a source of knowledge who has explicit or tacit knowledge and a receipt who has to interpret the knowledge so it is able to apply knowledge transferred. It is important to note that for transfer success, Knowledge codification at the source must to be done, because the knowledge at the source, even if explicit, has to be codified in an object with significance for both source and receipt.

C. Knowledge Transfer Barriers and Enablers

KT has barriers and enablers inside an organization, such barriers are related to economic, cultural and social capital (Liebowitz & Suen 2000). Depending on the type of organization some factors are more relevant, for instance in multinational environments, culture differences are more important than in small and medium environments (Ambos & Ambos 2009), (Gera 2012) and (Tichá & Havlíček 2007).

The barriers/enablers factors influencing KT usually come from hypothesis about people behavior, is used to presume that for KT occurs, a predisposition of the people to share knowledge should exist (Karlsen et al. 2011). Typically, factors always have two components, one organizational and one technologic. The organizational is related to the behavior of source and receipt of knowledge at different levels (multinationals, projects, teams and people), some factors include: communication mechanisms, trustiness, commitment, reciprocation, identity, shared goals, culture distance, language distance and geographic distance (Ambos & Ambos 2009; Duan et al. 2010; Garavelli et al. 2002), (Chen et al. 2008) and (Aurum A. 2008). The technological factors are pertaining to tools helping KT process at levels mentioned above, some tools like knowledge bases and ontologies are mentioned as cultural distance helpers, while some others like VoIP help
communication (Chen & Lovvorn 2011) and (Ambos & Ambos 2009).

III. METHOD

KT process stages were mapped against SR process steps. On the one hand the reference for the KT stages was taken from relevant authors according a literature review about KT in SR and their measurement, on the other hand the SWEBOOK was taken into account, for the SR process, because it is an effort from the computer society to characterize what is known about software engineering process, and promote a consistent view of SR. Finally, factors affecting KT were organized for each step mapped.

The literature review was defined as follows: step 1, a set of papers were taken from the result of a search equation in SCOPUS, over title, abstract and key words, the search equation used is as follows: (“Knowledge transfer” or “knowledge sharing”) and (“Software requirements” or “requirements engineering”) and (“metric” or ”measurement” or “indicator”); step2, a quick review over the results of step1 was performed, the review include the reading of abstract, introduction and conclusions, the criteria used for the selection was the pertinence about SR and KT measurement; step3, a full reading over the papers resulting form step2 was done, from this reading a group of KT stages were defined and each KT stage was linked against SR steps, from SWEBOOK, in order to find which KT stage and SR step matches better, the linking was done according common functions and goals from KT-SR; step4, a group of factors affecting the mapping were defined, according authors from step3 papers.

IV. RESULTS

After being applied the methodology explained above, the results found are as follows: step1, resulted in 373 papers; step2, resulted in 49 papers; step3, resulted in four dimensions where (Szulanski 2000), (Schwartz 2007), (Minbaeva 2007), (Goh et al. 2008) and (Simonin 2004) are the principal references, step3 is detailed in section A. Responding the questions proposed in the introduction, the KT and SR process are mapped, thus, making visible how KT takes place in SR process, and answering first question in the introduction. The step4 resulted in 7 factor groups that are depicted in section B, answering the question two proposed in the introduction.

A. The knowledge transfer process inside Software Requirements process.

Starting with (Szulanski 1996) who states that KT has four stages: Initiation, implementation, “ramp-up” and integration, other authors start using the word KT process adding or modifying steps like: information acquisition, documentation, transmission, source and receiver perception (Verkasalo & Lappalainen 1998), gather the knowledge from a source, code it through a channel, and pass it to a receipt (Albino et al. 2004), Idea creation, sharing, evaluation, dissemination and adoption (Levine & Gilbert 1998).

SWEBOOK divide SR in seven topics: SR fundamentals, Requirements process, elicitation, analysis, specification and validation, Practical considerations and SR tools. But, only four are going to be considered which are the related with the strictly SR process: elicitation, analysis, specification and validation.

In short, there are four dimensions for the knowledge transfer to occur.

1). Initiation: where the decision to KT and information acquisition is done by gathering the knowledge from a source, in a software context it is supposed that the source is motivated enough to share their knowledge because the source is the client who need the software, at this point the KT for software requirements differ from classical KT in organizations, because the receiver of the knowledge (i.e. software analysts) doesn’t intend to apply such knowledge but to build a software specification. This first step match with software elicitation stage for SR, because is where the first approach to business knowledge stake holders is made, those stake holders initiate the sharing of their knowledge and KT starts.
2) **Implementation**: is about the formal flow of knowledge from the source to the receipt, first software specification which could be seen as the source of knowledge codification occurs, the elicitation step ends and start the analysis of such first requirements, implementation cease or diminish with the software specification because is where the receipt starts using the transferred knowledge (requirements). At this point implementation step for KT differ from classical KT in organizations, because the receipt isn’t going to use the knowledge in his behalf, but for a software specification analysis.

3) “**Ramp-up**”: in this step initial knowledge codification and knowledge dissemination ends, software requirements are fully analyzed giving as a result the formal initiation of a software specification document. Consistence and conjecture of requirements are being evaluated. The software specification serve as a basis for agreement between customers and contractors on what the software product is to do as well and what it is not expected to do.

4) **Integration**: begins after the receipt achieves satisfactory results with the transferred knowledge, the knowledge is adopted and the perception of source and receiver happens. In the software requirements context is about the software specification end, the awareness of needs and ambiguity are evaluated, starting and ending the validation stage of software requirements, resulting in the final software specification.

Table 1.KT in SR will show the mapping, where KT steps appears in the first column and SR stages take place in the first row, E for elicitation, A for analysis, S for specification and V for validation.

<table>
<thead>
<tr>
<th>Table 1.KT in SR. Dimensions of KT vs steps of SR.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization.</strong></td>
</tr>
<tr>
<td>*Information acquisition.</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

B. **Factors affecting each KT and SR mapping.**
Based on (Szulanski 2000), (Schwart 2007), (Minbaeva 2007), (Goh et al. 2008) and (Simonin 2004) work, the next factors are defined for each mapping.

Factors F1, the Initialization and Elicitation are affected by the willingness to initiate transfer and propensity to share which are related with: acknowledgement and attribution, disseminative capacity, interpersonal connection and motivation of the source.

Factors F2, the Implementation and Elicitation are affected by the ease of transfer which is related with stickiness at initiation, stickiness at implementation, motivation, the awareness of need, the ability to transfer, the ambiguity of knowledge, the retentive capacity and modifiability of requirements.
Factors F3, the Implementation and Analysis are affected by the available time/access of source and receipt, reliability of the source, motivation of the receipt, ambiguity of knowledge, awareness of availability, absorptive capacity of receipt, understandability of requirements and its verifiability.

Factors F4, the Ramp-up and Analysis are affected by the requirements degree of conjecture.

Factors F5, the Ramp-up and Specification are affected by requirements internal consistency.

Factors F6, the Integration and Specification are affected by the available time/access of the receipt and source, the awareness of need from the source and the ambiguity of knowledge.

Factors F7, the Integration and Validation are affected by the correctness and completeness of the specification.

V. CONCLUSIONS

A view of SR as a KT process was done using a mapping between KT stages, here called dimensions of analysis, and SR steps, here called aspects.

The mapping serve as an approximation of KT transfer measurement because shows how SR take place in each KT dimension and which factors influence each match.

Knowledge transfer for software requirements is quite different from classical approach to KT, because transfer of a full body of knowledge is not intended, instead of that, needs who relate with some business knowledge are transferred.

Since KT for SR is a special case, classical metrics used for knowledge transfer should not be applied. For instance, metrics oriented to intellectual capital, number of people trained does not make sense for any SR stage.

Further work will include specific indicators for each factor.

VI. BIBLIOGRAPHY


Chen, J.-S. & Lovvorn, A.S., 2011. The speed of knowledge transfer within multinational enterprises: the role of


