Jani Salomaa

Translation Process of Multilingual Web Sites

Helsinki Metropolia University of Applied Sciences

Master of Business Administration

Entrepreneurship and Business Competence

Master's Thesis

26.11.2013

Helsinki Metropolia University of Applied Sciences

Author	Jani Salomaa
Title	Translation Process of Multilingual Web Sites
Number of Pages	42 pages
Date	26 November 2013
Degree	Master of Business Administration
Degree Programme	Entrepreneurship and Business Competence
Specialisation option	Information Technology Management
Instructor(s)	Janne Lautanala Senior Director Pia Koskenoja Principal Lecturer

The main goal for this master's thesis was to improve Avanade's capability to manage translation of multilingual web applications. For this purpose I created translation process, synchronization tool, and translation list. The translation process helps Avanade to reduce the time used for managing translation. The synchronization tool automates the process of copying words manually and the translation list provides real time access to all translated items for the customer. The customer can translate and modify existing translations whenever they want.

The translation process of multilingual web applications has been manual work in most cases. There has been several slow and error prone phases in the process. I have studied how we can automated these process phases. I implemented a process and tool that help Avanade to manage translation process and improve the speed of the process and customer satisfaction. We managed to improve process speed 30 times in some work phases of the translation process.

I used following theoretical frameworks as basis of this study. These frameworks allowed me to create improved translation process and synchronization tool. Ikujiro Nonaka's Knowledge Creation Framework is a theory that defines how we can manage and share tacit knowledge. It describes different means of describing knowledge based processes and what kind of steps there are. Software quality allowed me to measure and study how to improve software development. It provided me the methodology for measuring the impact of the improved process to our daily work.

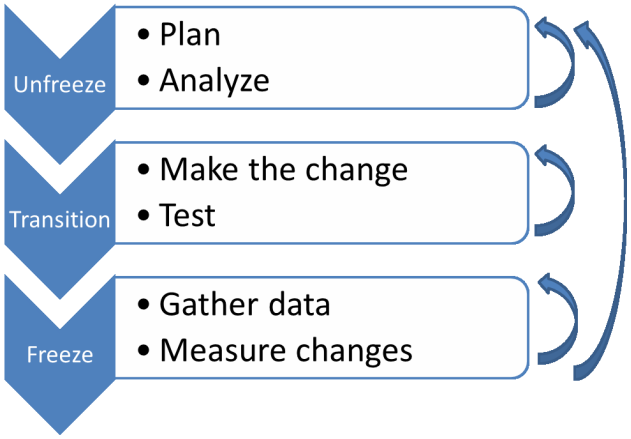
This thesis concentrated on studying Avanade work. I excluded client side improvements from the study. This is next possible subject that I need to study in the future. I also noticed some changes in developer's way of working. This could be also be something that needs to be studied in the future.

Keywords	translation, multilingual, project

Contents

Executive Summary

This thesis is action research study. It means that everything starts from the problem. There is a problem that needs to be solved. During planning phase student tries to find several theories that can be used to improve the status quo. In analysis phase the most suitable theories are selected and the others are discarded. After unfreeze phase starts the transition phase where improvements based on selected theories are implemented. After transition comes a freeze phase where student measures and validates the results.



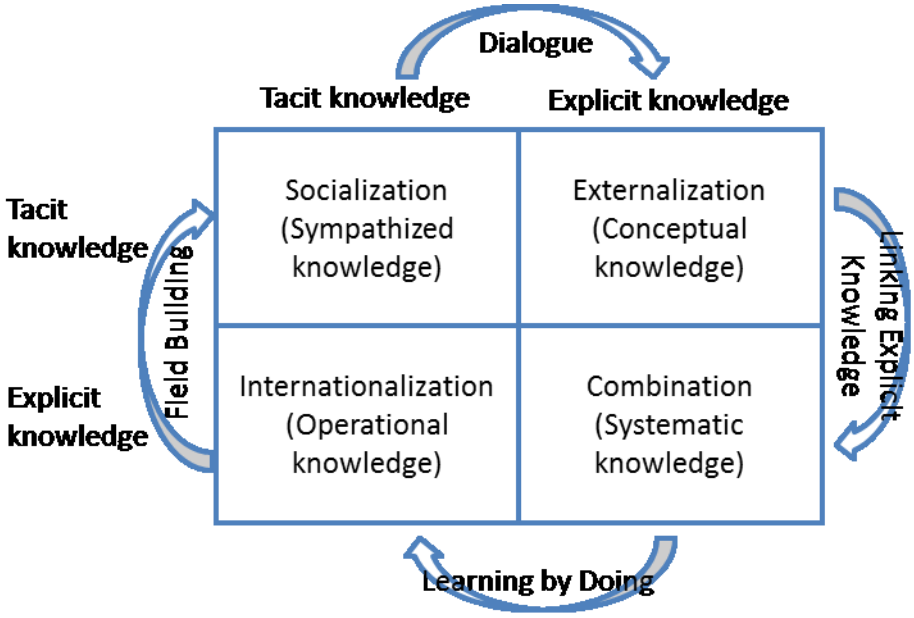
Action research phases based on three phases of Lewin's change theory (Lewin 1958)

In this thesis I talk about translation of the multilingual web sites and introduce automated synchronization tool that will allow developers to concentrate on developing web sites instead of copy pasting translated words which is really tiring and boring task to do. Developer has to copy words that need to be translated from development application to Excel sheet. Then the Excel file is sent to the customer. After this the customer distributes Excel file internally to people who are translating the words. Normally each person makes translations to their own file and then sends it back to the person who requested these translations. Now one file has multiplied into several files each containing translations for one language. Then all the copies of the original file are sent back to the developer. After developer has received the files, he copies translations from multiple Excels to the development application.

Whole process is full of manual work such as sending emails and copy – pasting words between different programs. This is still manageable if the number of words is kept to

the minimum, but in normal situation there can be over 1000 words that need to be translated in a single site and the number of words increases when the site gets new features. At the same time managing words that need to be translated is getting harder, because the developer has to find new words among 1000 old words without any help from the development application. The application can't tell which new words are and which are old. This means that some of the words will be left out from the Excel file and some of the words have already been translated, because developer thinks that they were the new words.

I studied knowledge management theories and selected Ikujiro Nonaka's Knowledge Creation Framework as my main theory. His theory is about communication and knowledge sharing (Nanoka, Takeuchi 1991). It helped me to formulate idea of moving all translated words to SharePoint and automate synchronizing words from SharePoint to development environment.



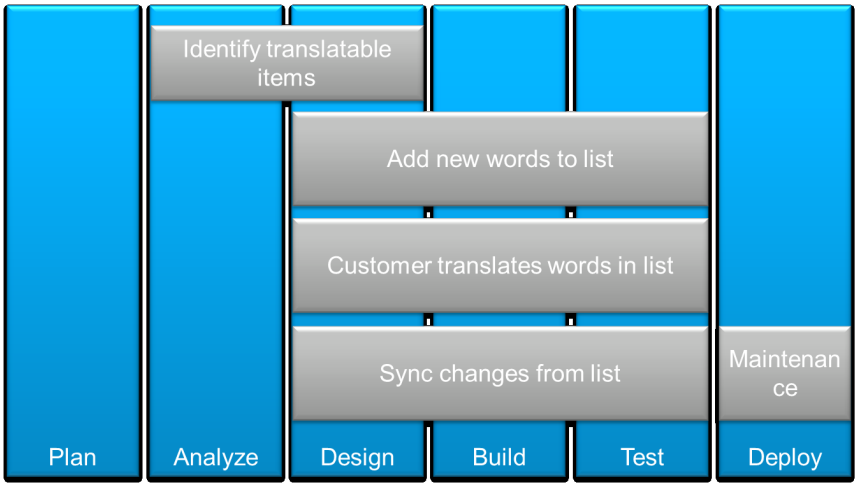
SECI model from Knowledge creation framework (Nanoka, Takeuchi 1991)

My process tries to improve two most problematic areas in translation of multilingual sites. The first area is creating and distributing Excel files. This requires a lot of manual work and coordination between people from different organizations and countries. With this new model all translated words will be created directly to SharePoint where customer can monitor them and translate new words when they are added by the developer. In optimal case all words will be translated "in real time". This however is not realistic, but using SharePoint instead of Excel makes a huge improvement to the old way of working.

SharePoint offers full access to all translated words and the customer can make changes to any of the words whenever they want. At the same time it offers different views to the same content. Each word can belong to one or multiple areas. This allows translator to see all words he/she needs to translate in one view.

The second area is automating manual developer work of copying words from development environment to Excel and then back to development environment. This is a perfect example of automating task that shouldn't require human work. This task is really simple, boring, and time consuming, but it requires lot of manual work from the developer. Instead of using developer doing this task, we are utilizing synchronization tool that synchronizes words between development environment and SharePoint.

My supporting framework is the software quality which helped me to formulate questions and metrics for measuring change effect of the translation process. These meters helped me to measure and improve translation process to its final form. Process starts from the analyze phase and covers all the next steps of the software development process.



Translation process alignment with Avanade development process.

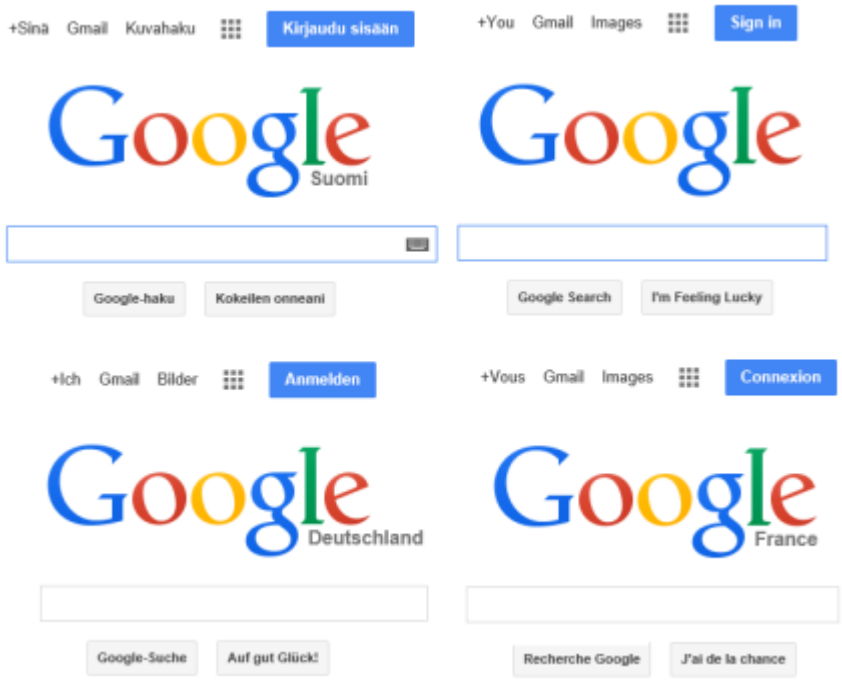
I collected information about the process for six months. My meters show that this process was a success. It improved our ability to provide a better service to our customer in all measured areas. We managed to decrease the number of synchronization related errors from 3% to 0.2%. We also use about 30 times less time while we synchronize words to development environment. At the same time we save approximately 8 hours every month when we don't have to collect the words from the development environment to be translated.

We can promise much better delivery times for translations than before. We managed to deploy a set of translations to test the environment six hours sooner than customer informed their translator to translate 50 words. Before we started using the translation process, I would have estimated the whole process to take two to four calendar days. It wouldn't have taken that long as actual billable hours, but coordinating and finding free time to do would have required that much extra hours.

1 Introduction

Designing and implementing multilingual sites is an often repeated activity in Avanade. Avanade is a joint venture company owned by Accenture and Microsoft established in 2000. We are specialized in Microsoft technology based IT consulting. We have 20 000 employees in 20 different counties. Our main target group is large global customers who operate in multiple countries.

Like every modern IT company Avanade has globally distributed projects. This adds additional challenges to our work. We have to manage knowledge and people in local and global levels. Therefore we need to implement new ways to share knowledge even though we might be alone at the customer site and our nearest colleague might be 10 000 kilometers away.



Picture 1 Example of multilingual site

Our task is to create multilingual web applications. These applications such as public web site for corporation or ecommerce applications have different functionalities like the feedback forms. These forms and other functionalities contain elements for example buttons and links. Each button and link needs to be localized so that the same application can be used in multiple countries. If you localize the site, you need to translate texts

(Sinä, You, Ich, and Vous texts in example) and also images (Google symbol in example), because they often contain localized content. As you can see from example above really simple looking sites require quite lot translations. In this picture you can seven elements that need to be translated for all target countries and languages. This is an example of really simplistic site, but ecommerce sites may have hundred different pages with complex functionality and one page can even have hundred different elements that need to be translated.

One might wonder why all this hassle, because you can use Google to translate each site automatically. This may be true, but it would work only with public sites. You can't use public internet services to translate intranets with classified information. Each time you translate a site, you need to send the content of the site to translation service and this is something that can't be allowed for confidential information. Current internet translation services are far from perfect and their translation quality is much poorer than the translation of a trained professional.

1.1 The problem description

Translation of multilingual sites normally contains lots of manual work while people are translating words and phrases in different geographical locations. After translating words is complete the programmers have to transfer translated words to the format that computer programs can understand. The biggest problem in this phase is that normally people who transfer words do not have any knowledge about the languages they are transferring. This may sound strange, but quite often we have to deal with situations where our client has offices in every continent. This means that we may have to transfer different European languages and also languages like Arabic, Chinese and Japanese.

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <data name="example_gmail" xml:space="preserve">
    <value>Gmail</value>
  </data>
  <data name="example_images" xml:space="preserve">
    <value>Images</value>
  </data>
  <data name="example_lucky" xml:space="preserve">
    <value>I'm feeling lucky</value>
  </data>
  <data name="example_search" xml:space="preserve">
    <value>Google Search</value>
  </data>
  <data name="example_signin" xml:space="preserve">
    <value>Sign in</value>
  </data>
  <data name="example_you" xml:space="preserve">
    <value>+You</value>
  </data>
</root>
```

Name	Value	Comment
example_gmail	Gmail	
example_images	Images	
example_lucky	I'm feeling lucky	
example_search	Google Search	
example_signin	Sign in	
example_you	+You	

Figure 1 Example of translation file

As you can see from the example, translation files are not very user friendly. Development environment provides better user interface for developers (on right), but customers don't have the development environment and they would have to use this file in its raw format and sending translation files to customer would be out of question. Small mistake in editing this file would result big problems in the development environment.

One of the biggest time consumers in IT project is rebuilding functionality and fixing poor quality implementations. If we can improve this with better translation process, it will save us and our customers a lot of money. According to Jones (2008) average amount of defects found from software can be something from 2 to 10 per each functionality. This means that if software has 50 different functionalities we can easily estimate that we have something from 100 to 500 defects. This is the total number of software defects. If we estimate that 5% of all defects are translation related, we can estimate that we have to fix translation related defect 5 to 25 times. Each fix means that customer employee for example from Russia will send email to their project manager, who will forward this email to me. I will forward email to developer in India, who tries to located correct word from the list of 1000 words and then replaces it with a new word. Then acknowledgement email will be send back to this chain for the Russian employee who will verify that change has been successful. The process of changing one word for one language may take from one day to one calendar week. In some cases one of the intermediates forgets to send an email and original requestor sends another request again after couple days and asks why this word is still wrong.

1.2 Research questions

Translation process of multilingual sites contains several steps and some of the steps are manual and error prone. This process requires intense collaboration between different people working in different countries and different companies. In some cases the customer has outsourced actual translation work to the third party and the process becomes even more complicated. I will try to improve current process and in order to do it I need to find answers for these questions:

- How to have better quality (less errors) in translation and transfer tasks?
- What are the steps that have the highest defect potential? How we can lower the defect potential of these steps?
- What kind of collaboration is needed between Avanade and the customer in translation process? What are the means of sharing information?

1.3 Research method

This study is action research. Action research is a research initiated to solve a recognized problem. Action research involves the process of actively participating in an organization change situation whilst conducting research. The word action research was first used by Kurt Lewin in his book Field Theory in Social Science. (Lewin 1951)

Action research is a participatory, democratic process concerned with developing practical knowing in the pursuit of worthwhile human purposes, grounded in a participatory worldview which we believe is emerging at this historical moment. It seeks to bring together action and reflection, theory and practice, in participation with others, in the pursuit of practical solutions to issues of pressing concern to people, and more generally the flourishing of individual persons and their communities.
(Reason, Bradbury 2008)

A primary purpose of action research is to produce practical knowledge that is useful to people. Action research is about working toward practical outcome, and also creating new forms of understanding. While doing action research you must participate to everyday life of the people you are investigating, it means that you are doing participative research. (Reason, Bradbury 2008)

When you do action research, you aim to change things. It contains three phases unfreeze, transition and freeze introduced by Kurt Lewin. Each phase helps organization to adapt and accept change. It's important that the organization and people have knowledge about the change and in each phase it's vital to communicate about change to help people to involve process as much as possible. During unfreezing phase communicating the results is even more important. (Lewin 1947)

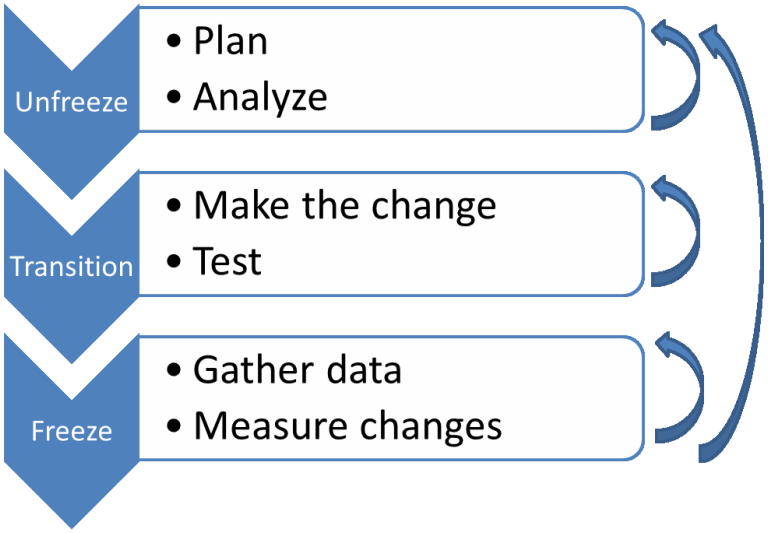


Figure 2 Action research phases based on three phases of Lewin's change theory (Lewin 1958)

During this process different ideas and theories have been tested. Each idea and theory starts in unfreezing phase. This is the first time when the individual or group becomes aware of a need to change. During changing phase the situation is diagnosed and new knowledge is explored and tested. During freezing phase new knowledge is evaluated and if accepted, added to the thesis. New information may change existing information and feedback loop starts again from the start. (Lewin 1958)

1.4 Theoretical Framework

I have chosen Ikujiro Nonaka's and Hirotaka Takeuchi's Knowledge Creation Framework as the base framework. It's based on Nonaka's article published on Harvard Business Review in 1991 where he used concepts of tacit and explicit knowledge. The word tacit knowledge was first introduced by Michael Polanyi in his book called Personal Knowledge: Towards a Post-Critical Philosophy. (Polanyi 1958, Referenced Nonaka, Takeuchi 1995)

I also use Software Quality Management secondary theoretical framework. It helps me to focus on software development aspects of the thesis. SQM contains three layers:

- Software Quality Assurance (SQA) layer
- Software Quality Plan (SQP) layer
- Software Quality Control (SQC) layer

This thesis will focus on SQA and SQC layers, because my aim is to understand and improve knowledge transfer during the implementation phase of the project.

1.5 Measuring and validating results

There are three measurements for this study.

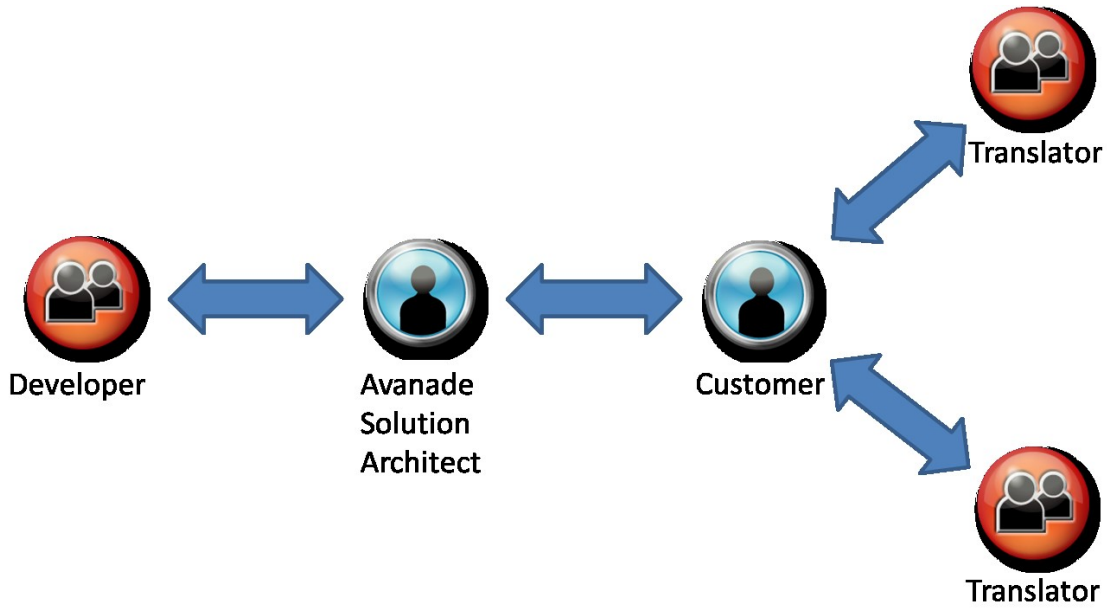
- How many defects happened during transfer process? This means how many errors developer did when he copied translatable words from Excel to Visual Studio. This measures quality of the delivery. We can measure how many change requests customer made before the translation process was used and then we can measure how many change requests customer made after we started using the translation process.
- How much we can save time with the translation process? The idea of the process is to automate boring and time consuming part of the translation, where developer copies words from Excel sheet to Visual Studio. It's boring, time consuming and error prone. Automation of this process saves a lot of time and allows developers do something meaningful. We can estimate how long it takes when the developer transfers words manually and how long it takes to do it with synchronization tool.
- How much faster we can react to the customer requests than before? This has direct relationship to client satisfaction. This can be measured by interviews of the key personnel.

Our projects operate using the same model, so I can safely assume that my results from this study are going to be valid in our other projects at least in Nordic region. Possible differentiators in projects models and customer behavior may differ so that all the results may not be applicable to other regions.

2 Unfreeze

2.1 Description of the original process

Translation of multilingual site has been manual process where the programmer creates an Excel sheet where all words or phrases are listed. Then Excel sheet is send to Avanade Solution Architect who coordinates translation with the customer representative. Client translates words according to their standards. In a normal situation these steps are repeated several times, because translating requires knowledge about the context. Without context it's really hard to understand the real meaning of the word. Each time a developer gets a list of words, he/she has to copy those words manually from an Excel to the development environment, because there is no integration between those applications. This is a really time consuming and monotonous task. It's also really difficult to avoid making mistakes in the process, because the number of translated words is normally really big and normally the person who copies the words from Excel to Visual Studio doesn't understand the language he/she is copying. This task should be automated, because there is no reason why it would need to require human interaction.

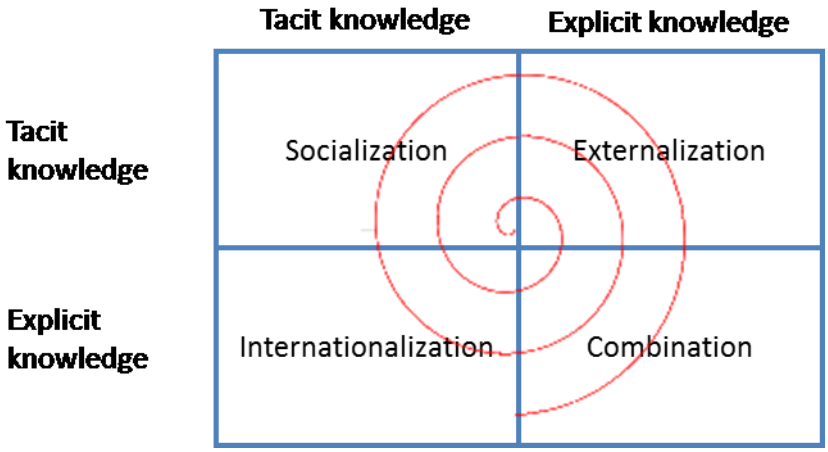


Picture 2 Original translation process

There is no single person who is responsible of translating a whole site. The translation process requires a co-operation between people who are translating the content, because they need to use same words to describe the same term. Some of the terms used are defined in corporation vocabulary and these terms should be exact.

SECI model describes concept of the translation process even though translation of the site doesn't require knowledge sharing as is. Developer sends a list of words to translate to target people. Then the first person starts translating words. After a while she/he needs some advice and he/she needs to contact her colleague. Then she may send first draft version to another colleague and he will change some words then they will discuss with different options and may come up totally new word. After they are happy to translation they send the result back to the developer who will add translated words to the system.

After first translation the customer may want to change some words to describe a feature better. This time they may use different ways to tell what they want to change. If project is still in the development phase, they may raise an error ticket or send an email to a developer directly. The problem with these methods is that there is no single point of contact that will take care of the translations. Some of the words may be lost, because one email can contain lots of other information and the developer just misses the request to change a word. Then we will spend time for tracking why some words are not translated or are translated wrong. There are number of reasons why this may happen and most of the times its waste of money, time and resources.



Picture 3 SECI model and spiral of knowledge creation (Nonaka, Nishiguchi 2001, 18)

This spiral of knowledge shows the iterative way of translating words. When the words have been translated for the first time, people in target country will notice that some words are incorrect or they have spelling mistakes. Then they make some changes and the developer replaces some words in target system. After a while people are happy with all the words. The process of translating words doesn't end here. In fact after this it becomes much harder, because the customer wants to improve and enhance existing site

with the new functionality. These enhancements require new words and gathering these new words from the list of thousand words becomes really difficult and time consuming task.

This task is not getting easier because there are multiple persons responsible translating words in different countries. In some cases they are scattered all over the world. At the same time there are multiple developers working with their own assignments, each requiring translations. This requires huge amount of coordination between Avanade and the customer responsible persons.

2.2 The start of the transformation

When I started working with this client, we were starting a big upgrade for the existing web application. This application was originally implemented by other company and it was our job to maintain and further develop it. This application is used in seven different web sites that use seven different languages. Now we have about 1 600 words in our translation tables. This means that total number of translated words is well over 10 000.

If the client wants to change one or more words weekly, this adds additional stress to maintenance team. The main problem in changing a word to translation table is that the person who actually does the transferring to computer understandable format does not understand the target language and therefore it's possible that they will accidently select a wrong line and change a wrong word. When people make these changes several times a week, the possibility something is accidentally changed increases.

We also started a new project which had a huge impact on the whole application. This required about 200 new translations for all language. We noticed that there were some old words that are not translated, because that particular functionality wasn't used in all countries. This required some additional work, because we didn't have any idea what words are translated correctly and what still required translation. During this time I had an idea about allowing the customer to translate the words themselves. This would help all participants, because customer could easily translate the words they wanted and they wouldn't have to make a change request every time. This would also allow us to concentrate of more interesting work instead of copy pasting words from email to our development environment.

3 Knowledge Creation Framework

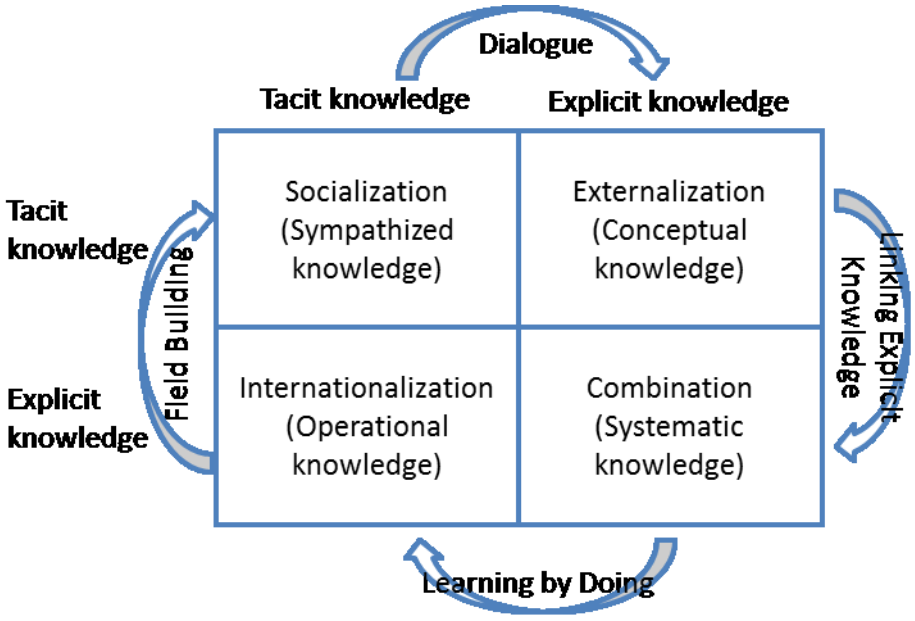
The Knowledge Creation Framework contains three models that share the same pattern. These models are:

- SECI,
- BA, and
- Knowledge assets.

In my thesis I will concentrate on the first two (SECI and BA).

3.1 SECI model

SECI stands for Socialization, Externalization, Combination, and Internalization. Each phase represents one of the four modes of knowledge conversation. SECI model defines framework for communication and knowledge sharing between people who work in different places and in many cases far from each other.



Picture 4 SECI model (Nanoka, Takeuchi 1991, 62, 71, 72)

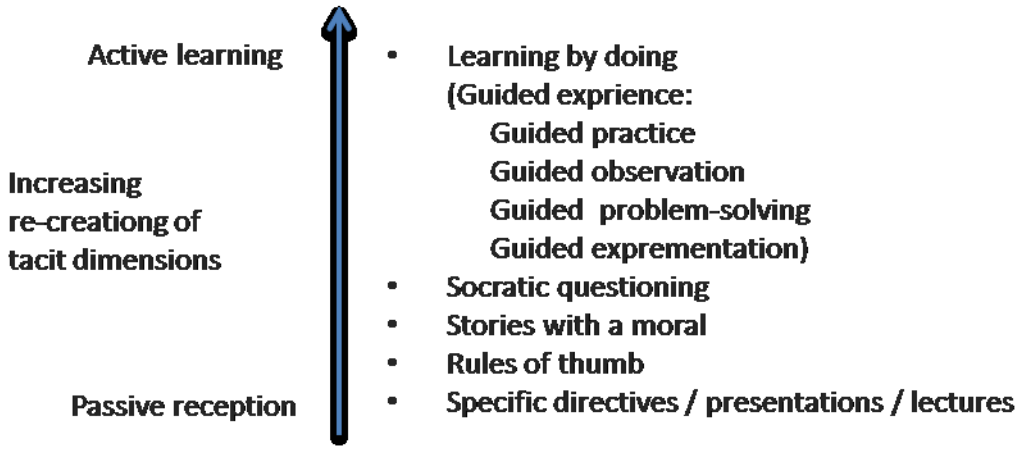
3.1.1 Socialization

Socialization is a process of sharing experiences, mental models and skills. An apprentice can acquire tacit knowledge directly from the master without using language by watching and imitating master's work. This is the most common example of socialization.

The key to acquiring tacit knowledge is experience. It's really hard to share tacit knowledge without any kind of shared experience. (Nanoka, Takeuchi 1991, 62)

Socialization yields what can be called to Sympathized knowledge, such as shared mental models and technical skills. (Nanoka, Takeuchi 1991, 71) In Avanade case this might mean skills like understanding pricing model of the customer when we are creating new web site. Inside Avanade we share technical skills like creating the SharePoint applications. This shared technical knowledge allows me to discuss with our Indian colleagues and understand each other.

There are many different ways of transferring tacit knowledge. Harvard Business School has created a model where different modes of knowledge transfer are aligned with activeness of the recipient. Tacit dimension defines how much information can be expressed as explicit form like documents and presentations. When the number of tacit dimensions increases, the more difficult is to transfer knowledge from one person to another with passive learning techniques (Ichijo, Nanoka 2007, 63-65). In practice socialization involves capturing knowledge through physical proximity. Knowledge is acquired from outside the organization though direct interaction with customers (Nonaka, Nishiguchi 2001, 16).



Picture 5 Modes of knowledge transfer. Harvard Business School Press

In consulting work our goal is to understand the client's needs and transfer those needs to working software. The number of tacit dimensions is often high, because our customer cases are complex. I'm most cases it's impossible to use learning by doing, because it takes too much time even though it might be the best way to gather insight knowledge about the customer company and their processes. The most common way of gathering

information from the customer is having workshops with predefined agenda. Each workshop concentrates on one or more parts of the project and during this workshop we try to gather the needed information about the project requirements.

Capturing tacit knowledge embedded within the organization by walking around inside the organization is another process of acquiring knowledge. (Nonaka, Nishiguchi 2001, 16) This is something we do quite often in our company. It may not be literal walking around the company, but more virtual activities like joining to one of the communities of practice. In these communities, you will meet people who are interested in same things and this helps you to learn something new and increase your tacit knowledge at the same time.

3.1.2 Externalization

Externalization is a process of articulating tacit knowledge into explicit concepts. The most common way to doing this is writing tacit knowledge to some form of model, concept or hypotheses. When we conceptualize an image, we express its essence mostly in written language. The written language is often inadequate, inconsistent, and insufficient. Such discrepancies and gaps between images and expressions help to promote reflection and interaction between individuals. Among the four modes of knowledge conversion, externalization holds the key to knowledge creation, because it creates new, explicit concepts from tacit knowledge. (Nanoka, Takeuchi 1991, 64 - 66)

When we try to convert tacit knowledge into explicit knowledge, we have to use metaphors, analogies and models. These three, reflection and interaction between individuals help us to create something new. New ideas are often offer a key to better business, which is the ultimate goal for all our projects. When we start working with our customer we have a series of workshops. In these workshops we use different techniques to define the purpose and requirements of the project outcome. One of the possible methods to gather customer requirements is fit/gap analysis.

Metaphor is a way of perceiving or intuitively understanding one thing by imaging another thing symbolically. Metaphor is important tool to create something new, but it also can be misinterpreted easily. Metaphor must be processed and harmonized by analogy. Analogy helps to reduce the unknown and highlight the commonness. Metaphor is more intuitive and holistic when analogy is more rational and structured. Thus analogy helps

to understand the unknown through the known and bridges the gap between an image and a logical model. After metaphor has been polished by analogy, we can create explicit concepts. These concepts can be modeled. In a logical model, no contradictions should exist and all concepts and propositions must be expressed in systematic language and coherent logic. (Nanoka, Takeuchi 1991, 66 - 67)

3.1.3 Combination

Combination is a process of systemizing concepts into a knowledge system. This mode of knowledge conversion involves combining different bodies of explicit knowledge. Individuals exchange and combine knowledge through different media. Reconfiguration of existing knowledge through sorting, adding, combining, and categorizing of explicit knowledge can lead to new knowledge. (Nanoka, Takeuchi 1991, 67)

In practice combination relies on three processes:

1. Explicit knowledge is collected from inside or outside the organization and then combined.
2. The new explicit knowledge is disseminated among the organizational members through presentations and meetings.
3. The explicit knowledge is edited or processed in the organization to make it more usable.

(Nonaka, Nishiguchi 2001, 17)

Combination is teamwork effort where collaboration between team members is vital. Instead of old manual teamwork we have knowledge teamwork. Teams are fundamental units in modern organizations and there is increasing recognition that collective work in teams is the most efficient way of creating knowledge. Knowledge teams have three main aspects.

1. In contrast to manual teamwork, pure knowledge teamwork raises productivity of the team only if different knowledge is dispersed among different people. If all knowledge workers in team have the same knowledge, one person could do the whole job almost entirely alone.
2. The result of the joint knowledge work is at least in part new explicit knowledge that can easily disseminated and further developed by all members of the company. This new knowledge is seen as feedstock of competitive advantage.

3. Replacing a person in large project is normally a time consuming task. True knowledge about customer and their field requires many years of work with particular customer or other customer in same field. This means that key people in project can be replaced, but only with high cost and it's not done easily.

(Ichijo, Nanoka 2007, 160)

These differences between manual and knowledge team work dictates our work more than we normally think. People have tacit knowledge and it cumulates during long projects. It means that we need to share knowledge between Avanade and the customer, but also between Avanade personnel who are working in same project.

3.1.4 Internalization

Internalization is a process of embodying explicit knowledge into tacit knowledge. It closely related to "learning by doing". When experiences through socialization, externalization, and combination are internalized into individuals' tacit knowledge bases in the form of shared mental models or technical know-how, they become valuable assets. For explicit knowledge to become tacit, it helps of knowledge is verbalized or diagrammed into documents, manuals, or oral stories. Documentation helps individuals internalize what they have experienced and enrich their tacit knowledge. (Nanoka, Takeuchi 1991, 69)

During internalization process a reader should analyze and comprehend the meaning of the information. This process leads to a creation of new knowledge. We don't learn new knowledge as discrete, isolated facts. If we do, we are not going to remember them long. The information we learn best is material that can be integrated with knowledge we already have. (Thong 2002)

In practice, internalization relies on two dimensions:

- Explicit knowledge has to be embodied in action and practice. This means that process of internalizing explicit knowledge uses concepts or methods about strategy, tactics, innovation, or improvement.
- Explicit knowledge can be embodied through simulations or experiments to trigger learning by doing. New concepts or methods can be learned in virtual simulations.

(Nonaka, Nishiguchi 2001, 17)

3.1.5 Knowledge spiral

Knowledge creation is a continuous process. Knowledge is created through a continuous and dynamic interaction between tacit and explicit knowledge. Knowledge is created through each mode of knowledge conversion interact with each other in the spiral of knowledge creation. (Nonaka, Nishiguchi 2001, 17)

Avanade's project model has six phases: Plan, Analyze, Design, Build, Test, and Deploy. Action research and SECI model both have similar features to ACM. All have iterative work model where we have reflective period where we can verify our progress and change direction if needed. In ACM we can go back for example from design to analyze, if we notice that we have forgotten something or something has changed. Within each ACM phase we have several tasks. Each task has several input and output documents that convey the knowledge between tasks and ultimately between different phases of the project. When each task starts the employees working on current task will read the input documents. This task involves combination and internalization phases from SECI model. An employee must understand the content of the input document before he can start to write output document. When he writes the output document he gathers new knowledge (socialization) and then shares tacit knowledge he managed to gather during previous phases to output document (externalization). When next task starts the knowledge spiral starts again.

3.2 Ba

Ba provides better understanding for translation process where words are translated, verified and retranslated several times during lifespan of the application. This can mean that one word will be translated several times or word can be translated then deleted because the original functionality no longer serves its purpose. Then it may reappear again when it's required in new functionality again.

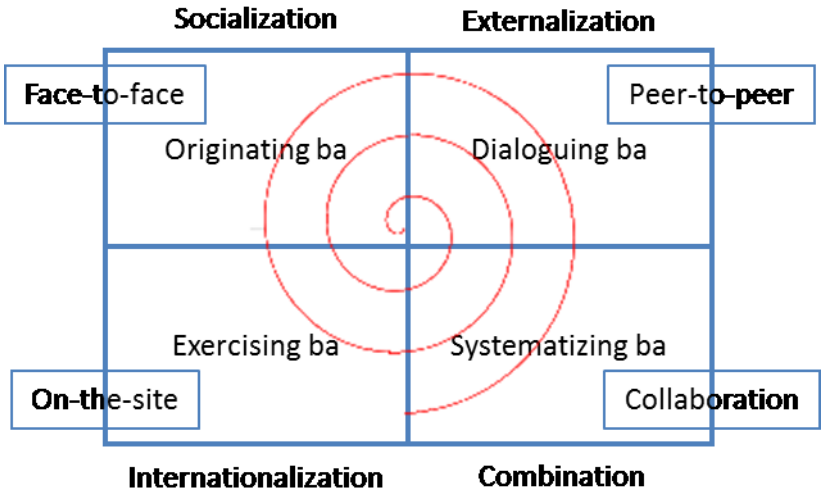
The original concept of ba was originally proposed by the Japanese philosopher Kitaro Nishida in his book called "Fundamental Problems of Philosophy: The World of Action and the Dialectical World" and further developed by H. Shimizu in his article "Ba-Principle: new Logic for the Real-Time Emergence of Information" on Holonics magazine. The

Japanese word “ba” can be translated to English word “place”. This place doesn’t necessarily mean any physical place. It can also be virtual, mental, or any combination of these. (Nonaka, Konno 1998)

Since knowledge is intangible, without boundaries, and dynamic and there are no way of stocking it, it has to be used where and when it’s needed to create value. To use and create knowledge effectively and efficiently, it’s necessary to concentrate knowledge at a certain time and space. The space is called Ba. Ba is platform for knowledge creation where knowledge is created, shared, and used. It functions as a medium for the resource concentration of the organization’s knowledge and of the individuals who own and create such knowledge. (Nonaka, Nishiguchi 2001, 18-19)

The most important aspect of ba is interaction. This means that knowledge creation process is collaborative process where multiple individuals join to this process by sharing, recreating and amplifying knowledge. Ba is a shared time and space for emerging relationship among individuals and groups to create knowledge. (Nonaka, Nishiguchi 2001, 19)

There are four types of Ba: originating, dialoguing, systematizing, and exercising. Each type supports a particular mode of knowledge conversion between tacit and explicit knowledge and offers a platform for a specific step in the knowledge spiral process. (Nonaka, Nishiguchi 2001, 19)



Picture 6 Four types of Ba (Nonaka, Nishiguchi 2001, 20)

3.2.1 Originating ba

Originating ba is the world where individuals share feeling, emotions, experiences, and mental models. An individual sympathizes or further empathizes with others, removing the barrier between self and others. Originating ba often is the primary ba from which knowledge-creation process begins, and this ba is associated with the socialization process. Face-to-face experiences are the key in converting tacit knowledge into tacit knowledge. (Nonaka, Nishiguchi 2001, 20)

3.2.2 Dialoguing / interacting ba

Dialoguing ba is more consciously constructed than originating ba. Selecting people with the right mix of specific knowledge and capabilities for a project team is critical. Through dialogue individuals' mental models and skills are created to common terms and concepts. Individuals share the mental model of others, but also reflect and analyze their own. Dialoguing ba is the place where knowledge is explicitly made, thus it's associated with the externalization process. (Nonaka, Nishiguchi 2001, 20)

3.2.3 Systematizing / cyber ba

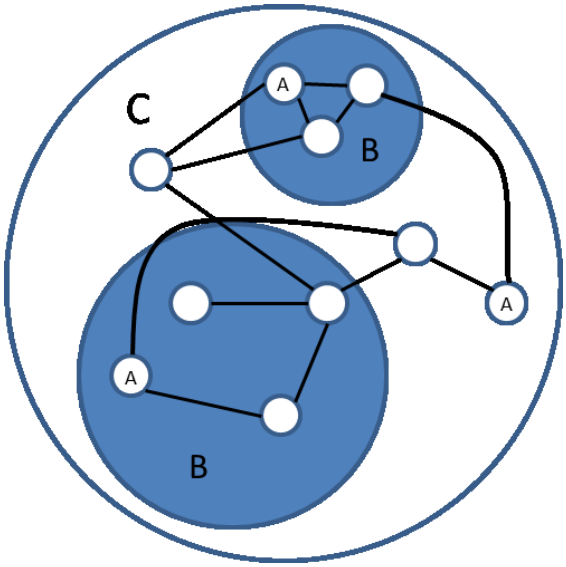
Systematizing ba is a place of interaction in a virtual world instead of sharing of space and time in reality. Here, combining new explicit knowledge with existing information and knowledge generates and systematizes explicit knowledge through justifying the concept throughout the organization. Thus, the systematizing ba is associated with the combination phase. The combination of explicit knowledge is most efficiently supported in collaborative environments. (Nonaka, Nishiguchi 2001, 21)

3.2.4 Exercising ba

Exercising ba supports internalization by facilitating the conversion of explicit knowledge to tacit knowledge. Focused training with senior mentors and colleagues consists primarily of continued exercises that stress certain patterns and the working out of such patterns. The interaction that takes place in exercising ba is on-the-site, which means that it shares time and space. Exercising ba synthesizes the transcendence and reflection through action, while dialoguing ba achieves this through thought. (Nonaka, Nishiguchi 2001, 21)

3.2.5 Ba, collaboration and networks

If we assume that Avanade's Ba would be our SharePoint environment, there are at least one other company that will participate to our project, customer. In most cases there are several companies that will have their own expertise area. This means that knowledge transfer between different participants is a necessity. Each company has their own Ba and the knowledge transfer between different Bas should be possible.



Picture 7 Networking within and across Ba (Brännback 2003)

Picture 7 shows the different ontological dimensions, where A represents the individual level of Ba, B project team level and C project level. When project starts it normally contains only small amount of people from Avanade and customer. This is a phase when originating ba starts for the first time and knowledge creation starts. After project starts we need to find out the solution for multiple dimension ba problems.
(Brännback 2003)

4 Software Quality

Software quality helps to understand what kind of problems there are in software development and how to improve quality of the development. This is important because the poor quality costs for all participants.

4.1 What is quality?

The International Organization for Standardization (ISO 9000, second edition, 2000) defines quality as the degree to which a set of inherent characteristics fulfills requirements. Quality can be used with such adjectives as poor, good, or excellent.
This definition contains three key terms: requirements, characteristics, and degree. Requirements can be stated by a customer in a made-to-order scenario or by product specifications in a commercial off-the-shelf product scenario.
Quality is characteristics refer to the capability of the deliverable or, in other words, the robustness (fitness) of the product (Chemuturi 2011, chapter 1). This is the definition of the quality I will use in this thesis. It defines the most important aspect of the quality needed in the software development projects, which is how well the software fits and implements the required features. Even if we wouldn't have any code errors, we couldn't have quality delivery if the software doesn't fit for the purpose it was meant to be used.

4.2 Software Quality

Software quality is defined as conformance to explicitly state functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

There is little question that the above definition could be modified or extended. If fact, a definitive definition of software quality could be debated endlessly. But the definition stated above does serve to emphasize three important points:

- Software requirements are the foundation from which quality is assessed. Lack of conformance to requirements is lack of quality
- A mature software-process model defines a set of development criteria that guide the manner in which software is engineered. If the criteria are not followed, lack of quality will almost surely result.

- There is a set of implicit requirements that often goes unmentioned (e.g., the desire for good maintainability). If software conforms to its explicit requirements but fails to meet implicit requirements, software quality is suspect.

(Thayer, Christersen 2005, Chapter 6)

4.3 Software Quality Assurance

There is a misconception in the software development industry that quality assurance means testing. Testing is testing; quality assurance encompasses inspection (verification), testing, and standards. In the glossary of the Capability Maturity Model Integration (CMMI®) model document for development (version 1.2, 2006), quality assurance is defined as "a planned and systematic means for assuring management that the defined standards, practices, procedures, and methods of the process are applied" (Chemuturi 2011, chapter 1). These test documents are normally derived from requirements. Each requirement can be divided into several test cases and these test cases can then be used for validating customer required functionality and it's suitability to the customer.

One occurrence of note that had a significant impact on the evolution of the concepts of quality was the transformation Japanese manufacturing organizations underwent. This transformation contained new ideas like quality control circles, zero defects, and right first time. Quality is not just the responsibility of the quality department alone; it is an organizational issue. This realization led to the development of the concept of total quality management, which requires the entire organization be involved in achieving quality—not just in terms of deliverables, but in every activity of the organization. (Chemuturi 2011, chapter 1)

Quality management hasn't been the most important part of the content management projects. These projects are quite often driven by corporate communications or HR type organization units with no or very little knowledge about software development projects. From their perspective good looking web site is the most important thing and quality means pixel perfect layout, not technically solid application. They wake up for quality when project is about to go live and in most cases it's too late to make any changes to applications. The biggest problem is that if there are no room for testing, applications will be installed to production without proper testing. This can be acceptable if application is

not business critical, but if company expects zero bug delivery without any quality management they will be disappointed. Fortunately quality knowledge has been increasing in these type of projects.

4.4 Software Quality Plan

The software quality plan is an outline of quality measures to ensure quality levels within a software development effort. The plan is used as a baseline to compare the actual levels of quality during development with the planned levels of quality. If the levels of quality are not within the planned quality levels, management will respond appropriately as documented within the plan. The plan provides the framework and guidelines for development of understandable and maintainable code. A plan also provides the procedures for ensuring that quality software will be produced or maintained in-house or under contract. These procedures affect planning, designing, writing, testing, documenting, storing, and maintaining computer software. (Lewis 2009, Chapter 2)

Software quality plan is a document that describes different standards, practices, and metrics, how these should be applied, monitored and assured. It contains chapters for each audience and helps them to understand what their responsibilities in software projects are. Quality plan is toolset for all participants of the project. It contains documentation, checklists, and roles for each person and phase of the project. There are five steps to implement good software quality plan:

1. Document the quality plan. In this step quality plan is written. Plan can contain or more documents and it should be treated as living document that should be updated during project.
2. Obtain management acceptance. Management participation is necessary for the successful implementation of a quality plan. Management is responsible for both ensuring the quality of a software project and for providing the resources needed for software development.
3. Obtain development acceptance. Because the software development and maintenance personnel are the primary users of a quality plan, their approval and cooperation in implementing the quality plan are essential. The software project team members must adhere to the project quality plan; everyone must accept it and follow it.

4. Plan for implementation of the quality plan. The process of planning, formulating, and drafting a quality plan requires staff and resources. The individual responsible for implementing a quality plan must have access to these resources.
5. Execute the quality plan. The actual process of executing a quality plan by the software development and maintenance team involves determining necessary audit points for monitoring it.

(Lewis 2009, Chapter 2)

4.5 Software Quality Control

Software quality control is the set of procedures used by organizations to ensure that a software product will meet its quality goals at the best value to the customer, and to continually improve the organization's ability to produce software products in the future. (Clapp 1995, 7)

Based on the definition, we can observe the following:

- Software quality control is a process carried out by one or more organizations. In my case this means that Avanade and customer organization must be part of this process.
- The goal of software quality control is to deliver a satisfactory end result to the customer at the best cost. This means that there can't be quick wins, because they will ruin customer relationship and software quality.
- The goal of software quality control for an organization also includes learning from each development so that software quality control will be better the next time. This should lead to self-learning team that won't repeat same mistakes every time again and again.

(Clapp 1995, 7)

These five steps to software quality control have been observed in the course of software management consulting in leading corporations:

1. Establish a software quality metrics program. Leading-edge software companies can measure both software quality and productivity with high precision. With software quality metrics program's full-scale measurements they can receive early warnings and cure problems before they will cause any bigger problems.

2. Establish tangible executive software performance goals. Now that software can be measured, it is possible to establish tangible, pragmatic performance goals for both software quality and productivity. Since the two key aspects of software quality are defect removal efficiency and customer satisfaction, reasonable executive targets would be to achieve higher than 95 percent efficiency in finding software bugs and higher than 90 percent “good” or “excellent” customer satisfaction ratings.
3. Establish meaningful software quality assurance. Companies that concentrate on software quality have higher productivity, shorter development schedules, and higher levels of customer satisfaction than companies that ignore quality. Since the steps needed to achieve high quality include both defect prevention and defect removal, a permanent quality assurance organization can facilitate the move toward quality control.
4. Develop a leading-edge corporate culture. Business activities have a cultural component as well as a technological component. The companies that tend to excel in both market leadership and software engineering technologies are those whose corporate cultures reflect the ideals of excellence and fair play. There is no external source of corporate culture. The board of directors, the CEO, and the senior executives are the only people who can forge a corporate culture, and it is their responsibility to do it well.
5. Determine your software strengths and weaknesses. More than 200 different factors can affect software productivity and quality. Each company must find their own combination of strengths to excel and attract leading talents from the work force available.

(Jones 2008, Chapter 5)

4.6 Measuring software quality

Two very important measurements of software quality that are critical to the industry are defect potentials and defect removal efficiency. All software managers and quality assurance personnel should be familiar with these measurements, because they have the greatest impact on software quality and also software costs and schedules of any known measures. (Jones 2008, Chapter 5)

The phrase “defect potentials” refers to the probable numbers of defects that will be found during the development of software applications. As of 2008 the defect potential of software includes five categories of defects:

- Requirements defects
- Design defects
- Coding defects
- Documentation defects
- Bad fixes (secondary defects accidentally included in repairs of prior defects)

(Jones 2008, Chapter 5)

As of 2008, the approximate U.S. averages for defects in these five categories have been measured in terms of defects per function point and rounded slightly so the cumulative results are an integer value. Note that defect potentials should be measured with function points and not with lines of code. This is because most of the serious defects are not found in the code itself, but rather in requirements and design. The defect removal efficiency measured against each of the five defect categories is approximately as follows:

Table 1 Avarage defect potentials in US 2008 (Jones 2008, Chapter 5)

Defect Origin	**Defect Potential**	**Removal Efficiency**	**Defects Remaining**
Requirements defects	1.00	77%	0.23
Design defects	1.25	85%	0.19
Coding defects	1.75	95%	0.09
Documentation defects	0.60	80%	0.12
Bad fixes	0.40	60%	0.12
	5.00 (∑)	**85% (average)**	**0.75 (∑)**

The measured range of defect potentials extends from just below 2.00 defects per function point to about 10.00 defects per function point. Defect potentials correlate with application size. If all defects are calculated, total number of defect potential is 5.00. As application sizes increase, defect potentials also rise. (Jones 2008, Chapter 5)

The phrase “defect removal efficiency” refers to the percentage of the defect potentials that will be removed before the software application is delivered to its users or customers. As of 2008, the U.S. average for defect removal efficiency has been about 85 percent. If

the average defect potential is 5.00 bugs or defects per function point and removal efficiency is 85 percent, then the total number of delivered defects will be about 0.75 defects per function point. However, some forms of defects are harder to find and remove than others. For example, requirements defects and bad fixes are much more difficult to find and eliminate than coding defects. (Jones 2008, Chapter 5)

Note that the defects discussed in this section include all severity levels, ranging from severity 1 “show stoppers” to severity 4 “cosmetic errors.” Obviously it is important to measure defect severity levels as well as recording numbers of defects. (Jones 2008, Chapter 5)

To have better understanding what kind of defects falls into each category here are some examples:

- Level 1: If user buys a product from the web site, he will get wrong product, because web site sends wrong product id to delivery system.
- Level 2: If user buys a product from the web site, system fails to send update to delivery system and operator must resend order manually.
- Level 3: Content editing functionality on the site crashes in some cases and content editor must retype the content again.
- Level 4: Title of the page has wrong color.

As we can see from the list, some of the defects are more important to fix and some of the defects may even prevent taking new system into production, before the most critical defects has been fixed.

5 Implementation

5.1 Description

Implementation starts with analysis and planning according Lewin’s change theory. This theory has been one of the most used theories in change management and it can be used in action research study. When I studied SECI and BA, I realized that we had similar problems than example companies in Nanoka’s books. They needed to share knowledge to make better products. We wanted to share translations to all required people to get site translated. Then I used software quality principles to estimate the possible impact of

the improvement. These helped me to formulate the first ideas of the synchronization tool.

We can map steps of the translation process to SECI model. Socialization happens when the developer creates a list of words that needs to be translated. In this step we create shared model and after each translator contributes their knowledge, as a form of translating words to Excel, the externalization happens. When customer combines all translated Excel files to Avanade, happens combination. Then the developer transfers all translated words to application and this phase can be called as internalization.

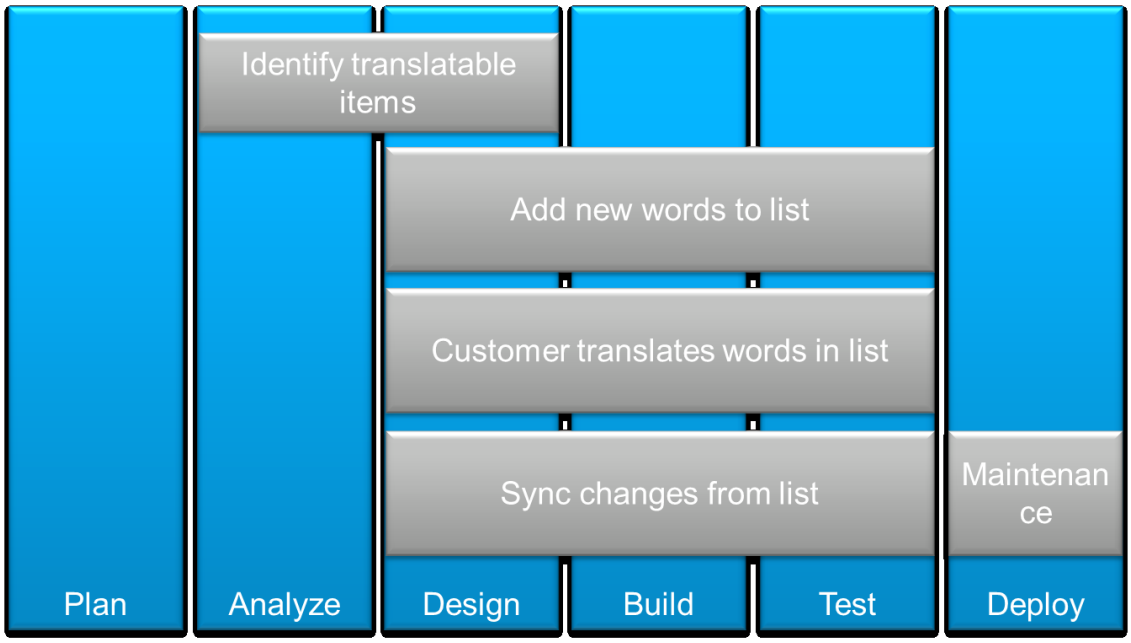
I have worked many years designing and implementing multilingual web sites and translating required texts has always been some kind of problem. In most cases it has been short period problem when we have created a new site for customer, but in my current assignment we are working with existing site where we are constantly developing new features. This means constant requirement for translations from multiple sub projects and the development teams.

To solve this problem I have created a process to help both the developers and the translators to improve their work efficiency and quality. This process has three parts: synchronization tool, SharePoint based translation list and documentation. Tool is used by the developers to synchronize changes between translation list and programming software. SharePoint is a ba for Avanade and customer where they can exchange information. SharePoint list contains all languages we need to support. List provides simple and familiar user interface for translator to add translations to new items and modify existing items. They don’t have to worry about sending translations to developer anymore, because they can edit words by themselves. It makes sending emails obsolete and helps everybody in their tasks. With the synchronization tool developers can easily get the latest translations from the list and at the same time they can send new translation items to the list that requires translation. The whole process takes couple seconds and it eliminates the transferring translated words to development platform totally. This means that developer can’t make any mistakes anymore, because there are no manual works. This improves the overall quality.

5.2 Translation Process

Translation process can be aligned to any software development methodology. This means that it's easy to use in all software development projects. During analyze and design phase solution architect or business analyst identifies all elements that need to be translated with the customer. These can be added to translation list at the same time when technical design document is written. In ideal case all words have been translated before the development even starts.

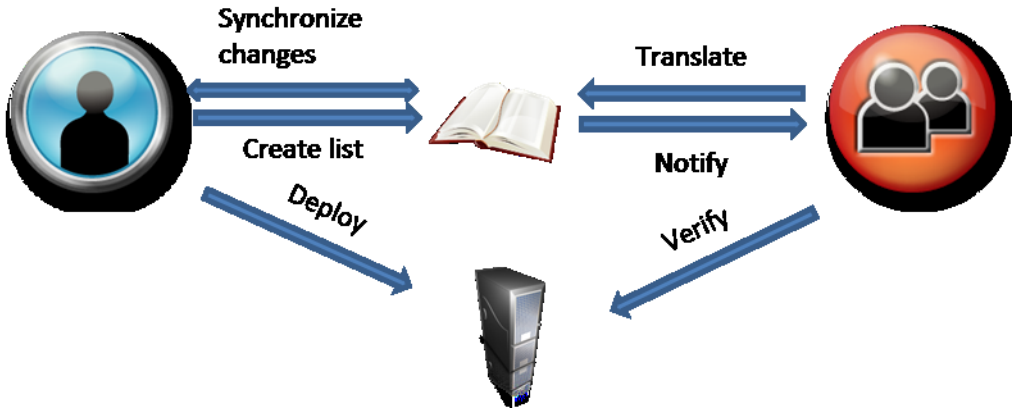
This helps development a lot, because developers can verify that there is enough room for each language. In many cases this can be difficult afterwards when layouts have been finalized. In any case it's much more expensive to make changes that late. It requires more work and additional testing when developer has to make changes to working code.



Picture 8 The translation process phases aligned to development model used in Avanade

Translation process starts when developer creates the translation list and synchronizes translated items to the list first time. This phase initializes the list and fills it with needed information. After initialization phase the developer makes final adjustments to the list like assigning proper rights for necessary people. When list is ready developer can add additional layout images for each translatable item. These layout images help translator to see better in which context each translatable work is. From layout image translator can see surrounding words and in some cases whole page or element.

Now list is ready and translating words can begin. Each translator can translate their words when it's best for them and there's no need for coordination between people anymore. Developer can synchronize changes whenever he needs to and each time they will get the latest translations. When developer deploys next version to web server, all changes to translations will be deployed at the same time and the result is visible for every translator. Then they can make changes if some of the words don't fit and these changes will get deployed to test environment again in next version deployment. This process of translating and synchronizing changes continues whole life cycle of the web site.



Picture 9 Translation process

5.3 Synchronization tool

Synchronization tool is used to automate transfer task from Excel or email to programming environment. This transfer task is normally the biggest reason for translation defects, because developer seldom can verify that the text he copies actually contains the right content. This may sound odd, but content can be in different language or even in language that does not use Latin characters. Some of the defects are almost unnoticeable, but some of the defects may even lead to monetary loss, because the meaning of the content changes and end user may be eligible for refunds or some other compensation.

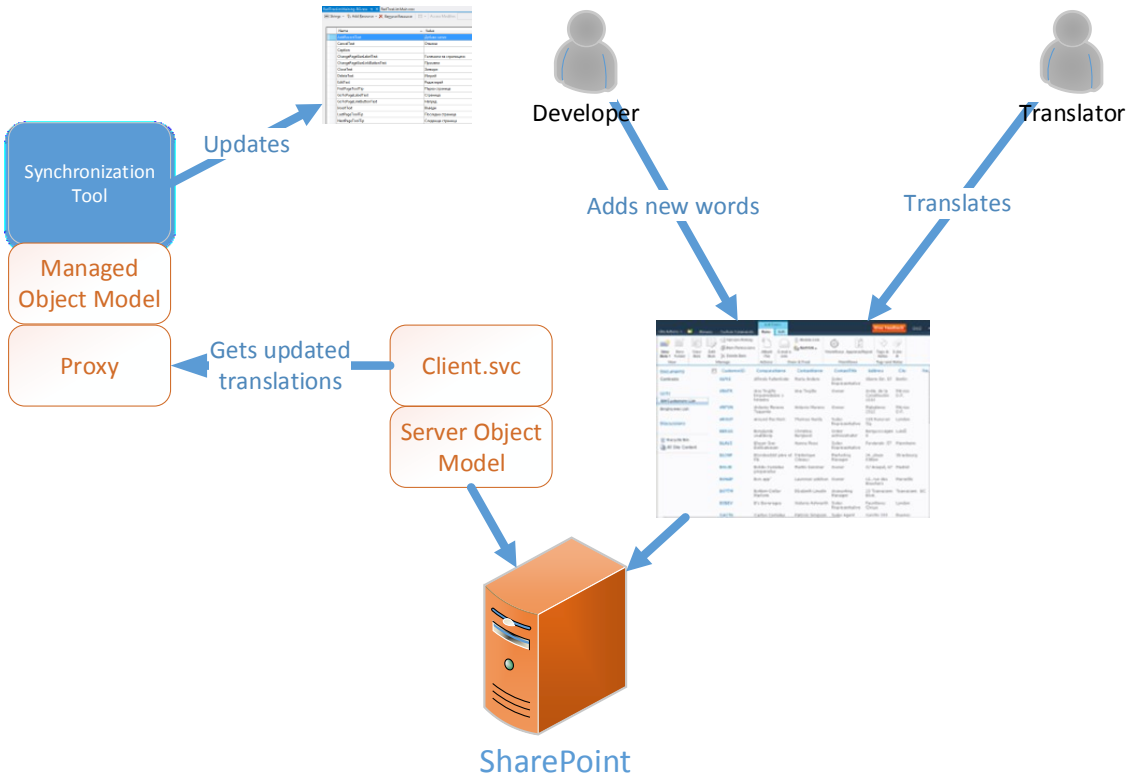


Figure 3 The synchronization tool in the translation process

Figure 3 describes synchronization tool. It uses SharePoint Client Object Model to communicate with Translation list. Developer and translator use normal SharePoint user interface to add new words or to translate words. With COM synchronization tool can work with SharePoint even they wouldn't be in same network.

5.4 Transition

Transition phase is according to Lewin's change theory the phase were change will take place. During transition phase the organization starts using new ways to work. It also means that process is constantly changing and improving based on the feedback of the organization. It's also a phase were organization must be trained to work differently. If this part of the transition is not done properly, organization may totally ignore the change and keep working with old way.

5.4.1 Development and testing

I started developing synchronization tool in summer 2012. I used my own personal development environment for first version of tool. It allowed me to test and create new

versions without any limitations. When the first working version was finalized, I started testing tool in Avanade project workspace. It was perfect place to test, because it uses same version of SharePoint as my client uses. It was also my first test where tool and Translation list were on separate computers. It allowed me to test network latency and other remote connection related problems.

I have run some tests with it and now it seems to be ready for first customer test. I have chosen my current customer to be the first user of this translation process. During December 2012 we will start using translation process in daily work. I will expect some changes to the process and tool after customer starts using it. I will make necessary changes to process and system.

I started testing translation process in December 2012. Between December and February 2013 I did testing in our own SharePoint workspace. I used real customer resource files and everything seemed working fine. After final tests I discussed with customer and we agreed that they would create new workspace for hosting Translation list. It was important that customer has unlimited access to Translation list, so they can access to it without any problems.

5.4.2 Moving from testing to production

Middle of March a customer informed me that they had created a site where we could create translation list and start using it for translating real production site resource texts. I created Translation list to production site and then I loaded all resource texts from production environment to list. After this day we used only SharePoint translation list for updating resource texts.

I had a meeting with my Indian team and gave them a presentation about the process and tool. It's important that they accept the tool for their normal process, because it won't work unless I have their commitment to this process. I was actually little surprised, because the feedback from India was really positive. I was expecting little bit transformation resistance, because this was quite far from normal process they have used so far.

During the first month of production use we noticed that there were some problems with the tool. It didn't notice if there were multiple resource texts with same key. This type of incident happened when two people were working on different tasks that were using

same resource texts and they both tried to add same resource texts to Translation list. I had to make some fixes to the tool that it could understand that there could be multiple translation items with same key. If the synchronization tool finds two items with the same key, it will notify developer, so he/she can remove extra resource items from the Translation list. After this fix the synchronization tool has worked fine. My Indian team helped me to improve process and tool with their work. They found all problems we had during production environment testing period.

5.4.3 Production

During the first month of production use I modified process a little bit based on the feedback from my Indian team. I also modified the documentation of the tool and the process based on the feedback. Now I have formulated best practice rules how this tool and process should be used. These help users to work with translation list. They are not mandatory steps, but make some of the tasks easier and more intuitive for people who make actual translation work.

5.5 Improvements

These are couple improvement I noticed during transition phase. They were small additions to whole process and they help developers, managers and translators to work better and easier. These can be used to monitor the translations of each country and they help translators and managers to monitor new and modified items in the list.

5.5.1 Best practice: Add the category column

The category column is important to categorize translatable words better. When words are categorized translators can find words related to certain area better. We did two projects during spring and summer. I created categories for those projects and it allowed me to create a custom view for all words related to those projects. It helped customer to translate all needed words. If category column is created, it should be multiple choice field. Some of the words can belong to several categories and they should visible in every category.

5.5.2 Create views

Use category column to create a custom view for each project or area. A view shows only the words related to then project and it makes the task of the translator lot easier. They can see all the words related to single area in one page and then they can be sure that there are no other words they need to translate. If new words needs to be added, they will be visible in a same view.

Create also date based views. Translator can see all words that have been added for example within the last month. Views can also be used by developers to show all words that has been modified during the last week. The customer can also follow the work of translators with a custom view that shows all words that haven't been translated yet. Views help all users of the translation list to be more effective in their daily work.

5.5.3 Use a datasheet view

A datasheet view allows translator to translate all needed words without changing page. Page automatically saves all the words translated when user moves to a new row. A datasheet view is almost identical to Excel sheet so it's familiar to users and very seldom requires any kind of training.

5.5.4 Use alerts

Translators, web masters and developers can order alerts from translation list. Alert tells if there are new words that needs to be translated in the list. It also can tell if there are new translated words ready to be deployed to the test environment. Alerts can be send immediately or daily. In normal situation daily alerts work really well, because the words don't need to be translated immediately. It's sufficient that all the words can be translated for example couple days after they have been added to the list.

6 Results

We have used the translation process for six months. There has been no recorded incident of wrong translations in production anymore. So far this translation process has been successful.

I have defined following measurements for this research:

- How many errors happens during transferring words from Excel to Visual Studio?
- How much time we can save with new process?
- How much we improve our delivery capability / reaction for sudden requests?

These meters allow me to monitor the change and how it effect on daily work.

6.1 The number of transfer errors

Before we started using this translation process we got approximately 3 change requests per 100 translatable items. This meant that every time we added 100 words we made a mistake in three words. These mistakes may have been that we accidently copied word from wrong line or didn't copy the whole content of the cell. This happens quite often if the content is whole phrase instead of a word or two. I found 10 different email threads concerning this type of problems. This allowed me to estimate the number of incidents per month. I have estimated that we translate approximately same amount of words every month. This is not totally accurate, but over period of three months it gives quite good approximation. I have estimated that we add 15 new translatable items per month. This means that we will translate 15 word * 6 language = 90 words every month.

After we started using translation process we got one change request from client, but this was an error in installation of the package. We have translated approximately 500 words since translation process stared six months earlier. This means that our current incidents per transferred words is 0.2%

The improvement in error percentage is from 3% to 0.2%. This means we have fifteen times less errors than before.

6.2 The number of saved work hours

It took approximately 10 seconds for a developer to add new word to Visual Studio from an Excel sheet. We have six language so this means that to get "one" translatable item from Excel to Visual Studio a developer has to repeat this 10 second task six times. If number of words is 15 as I have estimated earlier, it would take 15 minutes from a developer to add all required words. This is an optimal case, because developers normally take short breaks and get distracted when they have to do this type of monotonous task.

Now it takes approximately 30 seconds to update same 15 new items from the translation list to Visual Studio. I have tested this quite often, because I can add new words to the translation list one word at the time and then I can synchronize words from SharePoint to Visual Studio whenever I want. Developer adds new word directly to SharePoint and customer can translate words and this doesn't require any actions from Avanade.

The other time consuming task is to collect new words that needs to be translated from Visual Studio to Excel. This is something a developer needs to remember every time they add new translatable items to the application, because there are no way of getting those new words afterwards. This leads to a situation where we have to repeat this task several times during testing period, because we find new words that needs to be translated. This requires quite a lot manual word and sending emails to get all new words translated.

I did small test and I tried to find all new words I added to the application during one day. This took me one hour and 20 minutes to go through all words I added during that day. I had to go through 47 words from 12 different components to locate all new words that weren't translated and needed translations. Now if I need to check what the new words are, I can check it from the translation list within seconds.

The improvement in transferring words is from 15 minutes to 30 seconds. This means that we use 30 times less time in transferring than before. We also don't have to use any time to collect words that needs to be translated from Visual Studio to Excel anymore. This will save approximately 8 hours every month.

6.3 The improvement of the delivery capability

Before translation process we could promise that we can update new translations to test within two weeks. We had to collect new translations from visual studio to an Excel sheet manually. This was quite problematic, because Visual Studio couldn't tell which words where new and which were old. Normally we had to do a collection of new translations from code when then we had to locate matching words from the resource files and copy this information to an Excel sheet. This type of process meant that quite often we didn't locate all required words and we had to repeat this process after we got the first set of words and we noticed that some of the words were still missing translations.

Now we can ask customer to go to translation list and select a new items view. There are all words that has been added within one month. Translators can also order alerts from the list. SharePoint can send immediate alert or daily digest from the changes. This means that the translators can translate new words couple minutes after developers has added new words to list without any human intervention. We can promise that we can update new translations to test within four hours after customer has translated all required words to the translation list.

This gives us realistic possibility to react sudden requests from a customer. Before we had translation process in use, we could promise that we will deploy new translations to production in next release. Now we can promise that we can do daily fixes to translations in test environment if needed.

7 Reflection

7.1 Conclusions

The translation process has improved our capability to provide better quality with better response times than before. There has been one reported translation related incident in the time we have used translation process. It was in June when customer reported that some of words they updated to translation list were not deployed to test environment. This incident was related to installation failure of the package. It was not exactly related to translation process, but it was included to reports because we can't be sure that some of the earlier reported incidents would be related to installation failures.

One of the biggest success stories for the translation process happened in September were customer asked that one of the country sites would translate over 50 words in the morning. In the afternoon I got message from our Indian team leader that they had installed new package to test containing new words. It took about six hours from initial request to translate 50 words and deploy those to the test environment. There were five different persons from three different countries involved to this translation task and all of them had their normal duties and were only reacting when they were informed to do their part of the task.

Translation process can be used in any of the Avanade projects. It has no customer specific requirements. All Avanade projects have access to SharePoint and they can host translation environment at Avanade project site, if customer won't have SharePoint of their own. I have tested synchronization tool with SharePoint 2010 and 2013 versions. It should also be compatible with SharePoint Online, because it doesn't use any of the server side programming.

7.2 Next steps

This thesis deals mainly with Avanade capability. Next step is quite logical. We could study how this process can used to improve customer side problems and how much we can save customer's time in translation management.

The other study could be change in developers work. I noticed in my own work some small changes. Because I didn't have to worry about translations, I could translate some of the texts with internet based services like Google translate and see how well different texts fit to the page. Application should look good in all languages and this may sometimes require some additional work, because there can be quite big difference between different languages. For example a word might be 10 characters long in Finnish and 16 characters long in Russian. This means that application layout must be created in a way that buttons written in each language can fit to the space reserved for them.

References

Brue, Greg 2002, Six Sigma for Managers. McGraw-Hill.

Brännback, Malin 2003. R&D collaboration: role of Ba in knowledge-creating networks. Knowledge Management Research & Practice No.1

Chemuturi, Murali 2011. Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers. J. Ross Publishing. Electronic book.

Clapp, Judith. Software Quality Control, Error Analysis, and Testing 1995. Noyes Data Corporation.

Jones, Capers 2008. Applied Software Measurement: Global Analysis of Productivity and Quality, Third Edition. McGraw-Hill/Osborne. Electronic book.

Kazuo, Ichijo; Nonaka, Ikujuro 2007. Knowledge Creation and Management. Oxford University Press.

Kisker, Holger 15. 3 2012. Turning Data Into Business Value. http://www.forbes.com/sites/forrester/2012/03/15/turning-data-into-business-value/ Accessed 19.3 2012.

Lewin, Kurt 1951. Field Theory in Social Science.Harper and Row. New York. Ref Cadle, James; Paul, Debra; Turner, Paul 2010. Business Analysis Techniques: 72 Essential Tools for Success. Electronic book. Accessed 26.8.2013.

Cameron, Esther; Green, Mike 2009. Making Sense of Change Management: A Complete Guide to the Models, Tools & Techniques of Organizational Change, 2nd Edition. Electronic book.

Lewin,Kurt 1947. Frontiers in group dynamics. Human Relations, Vol. 1. Ref

Lewis, William E 2009. Software Testing and Continuous Quality Improvement, Third Edition. Auerbach Publications. Electronic book.

Nonaka, Ikujiro; Konno, Noboru 1998. The Concept of “Ba”: Building a Foundation for Knowledge Creation. California Management Review Vol. 40 No.3.

Nonaka, Ikujiro; Takeuchi, Hirotaka 1995. The knowledge-creating company. Oxford University Press.

Nonaka, Ikujiro; Nishiguchi, Toshihiro 2001. Knowledge emergence: social, technical, and evolutionary dimensions of knowledge creation. Oxford University Press.

Polanyi, Michael 1958. Personal Knowledge: Towards a Post-Critical Philosophy. Referenced in Nonaka, Ikujiro; Takeuchi, Hirotaka 1995. The knowledge-creating company. Oxford University Press.

Reason, Peter; Bradbudy-Huang Hilary 2008. Handbook of action research: Participative Inquiry and Practice. SAGE. Ref http://faculty.mu.edu.sa/public/uploads/1346012794.621handbook_of_action_research.pdf Accessed 1.10.2013

Scharmer, Claus Otto 1999. Knowledge Creation: A source value. Macmillan Press Limited. Edited by von Krogh, Georg; Nonaka, Ikujiro; Nishiguchi, Toshihiro

Schulmeyer, G. Gordon 2008. Handbook of Software Quality Assurance, Fourth Edition. Artech House. Electronic book.

Thayer, Richard H.; Christensen, Mark J. 2005. Software Engineering, Volume 1: The Development Process, Third Edition. John Wiley & Sons. Electronic book. Accessed 3.10.2013.

Thong, Lip Fei 6.11.2012. Circulation between Combination, Internalization and Externalisation. http://www.cyberartsweb.org/cpace/ht/thonglipfei/explaination.html Accessed 4.4.2012