

Requirements Engineering: Tool support for the mobile analyst

Simon Andrea Meier, simonandrea.meier@uzh.ch

University of Zurich, Binzmühlestrasse 14, 8050 Zürich, CH.

Abstract. Requirement elicitation provides theories and methods for inquiry in the context. Context inquiry is important because it helps to get the right requirements by going to the work context itself and understand what is needed at the field where the software is applied. The seminar paper introduces tools and methods that support the mobile analyst to elicit requirements in the working context. It describes a further developed technique applying the concepts of ART-SCENE for mobile devices. This method is then called ART-SCENE CoRE. One of the most established methods and other mobile requirement elicitation tools will be shown and compared. In the final part of this work, these techniques will be put in the context of requirement elicitation theory. The results of this paper is that mobile requirement elicitation techniques need to instil their connections to their roots in requirement elicitation theory. Sometimes electronic tools need to be backed up to paper and pencil because they have as well great advantages. Also this seminar paper tried to show a new method. But it could be elaborated that this idea lacks implementation-details and that the established methods in this field have the greater advantages.

Keywords: requirement elicitation, mobile tool and process support, context inquiry

1 Introduction

In the academic research field of requirements engineering exists the sub-field of requirements elicitation. This is the systematic finding, gathering and summarizing of requirements of the available sources of requirements. In this field there is a particular interest in contextual inquiry, which describes techniques that gather requirements in the context where the requirements occur. For this field there were techniques developed that support the discovering of requirements by a scenario-driven approach. This means that there exists a technique that gathers requirements for a problem to solve, a product to develop or a product to improve by defining different scenarios in advance. These scenarios describe how the work flow is imagined to be and these scenarios are later tested in the applying or working context of the problem, the end product, the using employee or the end user.

In the first section of this work the mobile analyst will be introduced. Ethnography, taught theory of the classical requirement elicitation techniques and methodical elicitation will be recapitulated.

In the second section existing tools for the mobile analyst will be shown. In the third section possible process support is elaborated. This elaboration forms in combination with the previous section the tool support for the mobile analyst.

In the fourth part advantages and drawbacks of the tool support will be discussed out of which a conclusion will be made for the last part.

2 Mobile Analyst

In this section the mobile analyst will be described.

2.1 Ethnography

The mobile analyst is a requirement engineer that uses the methods of ethnography. The word ethnography is Greek, where "ethnos" stands for population and "graphein" stands for describing. Ergo, ethnography is the description of populations and groups. In requirement engineering (RE) ethnography is used for example, to describe a group of software-users or employees that interact with a computer system. In their early paper "Presenting Ethnography in the Requirements Process" Hughes et al present techniques how to present results gathered from ethnography in requirement engineering: "Our presentation techniques rely on the use of three viewpoints; the work setting, the flow of work and the organisational and social perspectives on work evident in the study" [14]. This shows that ethnography in the requirement engineering has to do with the working context. A mobile analyst in the sense of ethnography is therefore a mobile analyst of a working context. Mobile comes from the Latin word "mobilis" and stands for movable. Thus we can describe the mobile analyst as a movable analyst of a working context. In the paper "A methodology for the selection of requirements engineering techniques" by Jiang et al ethnography is summarised as a RE process that has its most common area of application in requirement elicitation. This defines our researched target further to a movable analyst of a working context, using methods of requirement elicitation. Therefore Chapter 4 of the script Requirements Engineering I, "Requirement elicitation and analysis" [3] will be recapitulated in the following section.

2.2 Requirement elicitation techniques

In this section we put a focus on the methods and techniques of requirement elicitation that could support the mobile analyst in analysing the working context for which he has to collect requirements for a new software.

Requirement elicitation is: "Searching, capturing and gathering requirements from the available requirement sources". [3].

In order to search, capture and gather the right requirements it is important to understand the wishes and needs of the customer. Out of this the requirement engineer has to recognise the customer problems and show possibilities how to solve it. There are several techniques from ethnography that help the engineer to recognise the customers problem:

- Interviews
- Observation of end-users
- Workshops
- Role-games
- Analysis of examples
- Surveys
- Comparison with other solution
- Prototyping

For the interested reader Jiang et al present a list with further RE techniques [5]. Next to the recognition of problems and the capturing of concrete requirements for a software, requirement elicitation involves secondary processes that support the main-activities and contribute to good engineering of software. These activities are:

- Task clarification
- Analysis of risks
- Forming of consensus
- Recognition of conflicts and dissolving
- Animation for creativity at all involved

We see that the underlying activities of these techniques are:

- Talking with people
- Observing peoples behaviour
- Observing peoples process-flows
- Studying process documents
- Program software solutions

2.3 Methodical elicitation

Glinz [3] also shows four elicitation methods:

- Object analysis
- Decomposition analysis
- Event-Reaction analysis
- Scenario analysis

Object analysis: Under this approach the context field is analysed into objects which are the candidates for the model elements. This structure oriented method is especially used when the context is described with texts or when there is any other description about the context available. The grammatical analysis of a text, describing the requirement environment, leads with this method to nouns, adjectives and verbs. These are the candidates for the requirement objects that tell about the context where the software has to be implemented.

Decomposition analysis: In the decomposition analysis problems are decomposed into partial problems. This leads to a hierarchical model of problems to solve. The leaves of this problem-tree are actual requirements that the requirement engineer can write in the requirement specification. These requirements are therefore implementing-tasks for the developers.

Event-Reaction analysis: In this analysis for each event there has to be drawn the behaviour of the system through interactions with the system that provokes reactions.

Scenario analysis: The idea of this is the analysis of the logical sequence of interactions between actors and systems. The outcome of this are process chain models. Glinz defines a scenario as: "Is a ordered set of interaction between partners. Usually a system and the external participants" [4].

In addition to the analysis done from the customers point of view it is also useful that the customers stakeholders are analysed too. This is necessary in order to find the key persons that influence the problematical environment. With a mixture of the recapitulated methods and techniques from Glinz, the requirement engineer defines the targets and future capabilities of the new software. His key-drivers are the problems, visions and the aimed benefit of the customer. The requirement engineer is responsible for using the right methods and techniques in order to find the problems in the customer's environment that can be solved with software.

The latest papers in research show that one of the most frequent cited method and tool, that supports the mobile analyst uses scenarios to elicit new requirements. Practically, scenarios are written down in list items of activities for a specific work flow. Thus it is important to capture the appropriate scenarios. Modelling scenarios or use-cases can be done with the methods shown by Glinz [4].

Methods for modelling single scenarios or use-cases:

- Free text
- Structured text
- State-Charts
- UML
- Activity diagram
- Interaction diagrams

These methods help to understand how the scenarios, the mobile analyst will look at, were generated. Further we will look at techniques that can support the mobile analyst in eliciting requirements from the generated scenarios. In there we will focus on techniques derived from scenario analysis.

3 Mobile tools

As soon as the mobile analyst has a concept of the working context, with the help of scenarios or process chain models, he can use technologies to go to the place where the scenario take place. In situ, he can compare the modelled scenarios with the actual happenings. Differences between model and reality are new, potential requirements. This section will introduce tools that support the mobile analyst to do exactly this.

Paper and pencil Coming back to the roots of requirement elicitation, writing down requirements on paper is the classical technique. In this section it is investigated how the transition from paper to electronic device happened and what the underlying reasons might be. Experiences with the new mobile devices will show that the traditional techniques still have some very good qualities. Regarding our ecological environment and the effects of our power plants on it, it could, in appropriate situations, be reasonable to use again paper and pencil. In the end we always must keep in mind that when laptops, smart phones and other electronic devices run out of battery, the elicitation of requirements has to be backed up on paper.

Evolution of paper Recapitulating, writing down requirements on paper pre-conditions several activities: First it makes the requirement analyst think about the context of the requirements. He is collecting material from several sources like interviews, observations, existing documents and own experiences when helping and working in the process. In the past the requirement engineer was often part of the working process but had the additional freedom and time that he could think about how to reach a special target by changing something in the work chain and putting it into software. Therefore the way how he captured the requirements was not so important. It was rather important that he did it. Paper was the most intuitive and easiest way. When the development of software proceeded it was more and more a question of detail and subtle changes in software. It was not only about writing down what a customer needed, it became more important how he needed it and with what features. This lead to extensions in requirement elicitation techniques. One of the biggest problem that remained with paper eliciting was that the requirements could only be written down statically. In the past this was sufficient because requirements were stated bigger. Today they are rather subtle and more detailed.

Secondly it is asked what the differences and long lasting values of the paper eliciting technique are, comparing to the electronic versions. Requirements on paper are longer lasting, because they don't depend on any interpreting machine that parses them. On the other hand paper elicitation has its drawbacks in the automated recognition of conflicts. It's structure is static, quite open and its degree of freedom is mostly unrestricted. This has influences on the communication it allows between engineer and customer. Due to the paper form, both parties are talking to each other eye to eye and are able to write down notes. From the

notes it can be observed that each person has its own keywords. Most of the conflicts occur when one keyword that has been written down on the paper is not understood or accepted by the other party. The paper both parties are working on can be analysed. A high number of written words or graphs stands for a more dominant speaker. Keywords that are not very much elaborated might have been issues that were discussed heavily but no agreement has been found for. The lack of agreement is the reason that the keyword is not well elaborated on the working-paper. Another indicator of negotiation troubles are keywords that are written in an unusual handwriting. Maybe the person writing it was under stress thinking about possible issues and consequences the discussion about the keyword would arise.

3.1 Smart-phones

Syed et al describe that observation, interviews, and ethnographic techniques have become very popular elicitation techniques and that the commonly used devices to perform these activities are desktop computers, laptops, video cameras, and audio recorders. But they also mention that these devices are large in size and difficult to move. "So the scenario happens like that analysts and stakeholders need to go near the devices rather than supporting devices are brought near them" [13]. One solution for the problem of mobility are smart phones. Smart phones combine different tools like audio recording, camera and electronic notepad. So the supporting devices by Syed are combined in one tool such that they are ready and available on-site for the mobile analyst.

Camera The camera in a smart-phone can be used to take images of situations or scenes where a software is in use. For example it can take a picture of a person trying to scan a barcode that is hard to read. The picture can show why the code is difficult to read and how the bar-code-reading software has to approach the problem. This can be useful for the developers who can recognise circumstances like weak printing, shadowing light situations or difficult positioning that lead to exceptions in the software. With this knowledge it is than easier for them to find the right solutions, building better intelligence in the background-logic. Showing an example of this: Due to they recorded pictures the developers could come up with the idea of a fuzzy-logic that tries to complete inconsistent bar-code-information.

Audio recording The microphone of the smart-phone can be addressed with different mobile software. For example the application for the I-phone from Apple is called "Language-Memos" which is stored under "Serviceprograms" and is a installed by default on the phone. With just one touch on the red-button the I-phone starts a memo-recording. This can be used for example in situations where a requirement engineer is doing a walk-through. There he follows a list of activities, also called a scenario. E.G. he is trying to use a banking-software-prototype, let's say a tool to send of an offer to the customer. Therefore he follows

the steps of his scenario, for example: He first types in the name of the customer, his birthday and city. Then he starts a query where he tries to find an appropriate product for the customer and at last he sends off the offer as a PDF-attachment via e-mail. At the same time the "Language-Memo" application of the I-phone is running in the background so that he can speak out comments of the usage. For example he could be saying that in the current birthday-selection-field it is not possible to enter the 29. of February 2012, or he could be mentioning that the queries can only be done in SQL and that this is not understood by the majorities of the bank employees. He could also be saying that the attachment of PDFs is not always working and that this has to be fixed by the developers. When finished with the review later on, the requirement engineer can listen again to the recording and decide how to improve the problems. Similar to the I-Phone standard application there is a standard memo application and a quite popular application called "Astrid" on Android.

Electronic notepad On smart-phones there also exists application that imitate a notepad. With this a requirement engineer can write down requirements as single notes on his smart-phone. The advantage of this is that the notes are in machine letters and can therefore be read more easily. The noted requirements can be edited and sorted because for every note there is an automatic generated time stamp. Evernote [1] is a tool that works on a broad range of devices and makes the electronic savings of notes possible.

3.2 Tablet PCs

A tablet PCs such as the iPad, the WePad or the Archos9 can be seen as a mixture between a smart-phone and a Laptop. It extends the smart-phone with a bigger display and more calculation power but it is not as big as a laptop. On the other hand it has no keyboard but a touchpad. The tablet PCs could be an ideal requirement elicitation tool, because it is very well suited for reviews of graphs, code or prototypes.

4 Mobile processes

In this section processes are presented that use the shown mobile tools in order to provide mobile tool support for the mobile analyst. These processes take the previous tools as a lower level infrastructure, combine with them appropriate software and form a new service, the tool support for the mobile analyst.

4.1 Mobile Scenario Presenter

The Mobile Scenario Presenter by Seyff [7] is a scenario-driven technique for discovering and documenting stakeholder requirements. It is based on the analysis-methods Art-Scene and Art-Scene Core.

Art-Scene is a method that analyses requirements trade-offs via scenario evaluation. It supports analysts and stakeholders by automatic transformation of use-cases to scenarios (Structured list of example-activities).

Art-Scene core extends the art scenario evaluation by supporting it with contextual requirements elicitation (CoRE). It adds the contextual activities, on-site scenario validation and on-site scenario walk-through to Art-Scene. Art-Scene CoRE is the appropriate process support for the mobile analyst because the contextual activities support requirement elicitation in the work place. Thus in the following Art-Scene CoRe is elaborated.

ART-SCENE CoRE From ART-SCENE there was a software implemented with the name "ART-SCENE presenter". The Art-SCENE presenter is a web-based tool that supports requirement engineers. "The tool is based on CREWS-SAVRE and provides different functions for viewing scenarios and the requirements generated for them. It offers functions to add, edit or delete events, comments, and requirements" [10]. It makes scenario evaluation for software possible. The idea of the software is a screen view divided in the middle. The left side shows the different steps of a flow of actions e.g. going into the alps and using a ski-tour software. On the right side it shows scenarios that could arise for a marked step of the work flow from the left side. To find alternative scenarios that the software doesn't provide yet the requirement engineer uses the ART-SCENE presenter. He models the usage of the ski-tour software by drawing use-cases. The presenter has an automatic translator of use-cases into a list of steps. When the requirement engineer models all use cases and uses ART-SCENE he gets for every use case a list of steps that the user will do when he uses the ski-tour software. Now he can evaluate the workflow by following the ART-SCENE presenter. One of the goals is to raise "what if"-questions and to think about alternative scenarios that could occur in the workflow. This method makes it possible to find new requirements in an effective and convenient way.

For example a group of people want to go to Davos and use a ski-tour planing software at home where they define that they want to go to a certain restaurant. On the ski-track the software guides them the right way. But then they decide to try out a restaurant of which they only know the name. Now with ART-SCENE the requirement-engineer can write in these "what-if"-scenarios that the software has to provide a possibility to search for the restaurant and re-plan the route, if they change their mind on the tour.

In this example the software "ART-SCENE presenter" is not 100 percent applicable, because a laptop can hardly be taken to the Ski-Track. Therefore the method is extended with ART-SCENE CoRE for applying this analysis method to smaller mobile devices. The technical implementation of ART-SCENE CoRE is done in the Mobile Scenario Presenter. The Mobile Scenario Presenter is a software that enables the requirements analyst to compare the predefined scenario with the alternatives that could occur, on a smaller screen. "The Mobile Scenario Presenter is an ASP.NET web application, based on the ART-SCENE Scenario Presenter, adapted for mobile browsers running on PDAs. The applica-

tion uses wireless access to the database designed for scenarios and requirements of the ART-SCENE environment and is optimized for Microsoft's Pocket PC OS including the Microsoft Pocket Internet Explorer" [12]. The system has been tested in the London public transport system and one normal scenario was "The customer looks at the countdown display" and one of the alternative question was: "What happens if the customer is younger than expected" [7]. The MSP technology works on PDAs, this distinguishes it from other mobile technologies and most probably MSP can also be applied to mobile phones. A features of the MSP is that it is possible to check in, with a smaller keyboard, new requirements that occur during the workflow evaluation. A feature of this elicitation technique is, that the MSP makes it also possible to record audio-requirements and therefore new requirements can be gathered when actually there is no possibility to write them down on the physical keyboard. The .NET technology behind MSP enables the user to download a word file, where requirements can be put down. While using the word file the communication with the providing server is going on and notes are saved.

ART-SCENE CoRE includes the following activities [9]:

- Scenario generation and walkthrough preparation
- On-site scenario validation
- On-site scenario walkthrough
- Analysis and follow up

4.2 WikiWeb

With the distribution and more common usage of Web 2.0 services, web-services gain the ability of processing more complex tasks. Some of them are even suitable for the usage in RE. One example are WikiWebs. Ferreira and Rodrigues da Silva introduce collaborative requirement engineering via Wikis. They show that Wikis are simple tools that "are based on an open model and make easy edition of workflow possible" [2]. This could help the mobile analyst because he is able to invite participants of his analysed workflow to collaborate on the requirements. For example he could capture, in a first iteration, all the main requirements for a software and put it on a wiki. In a second iteration he could invite the particular employees to the wiki, where they can review the requirements and edit them. Rodrigues and Ferreira argument that the upload of the requirements on the wiki on its own has a positive effect for the requirement analyst: "By comparing and commenting other participants inputs, the user not only becomes aware of what his peers know (the community knowledge), but he also benefits from an unconscious self-assessment effect, hence consolidating his believes" [2]. A tool for implementing a WikiWeb in the requirements-engineers environment is the opensource project "Mediawiki" which is under the GPLv2 licence.

4.3 Micro-blogging

Another Web2.0 service-form is microblogging. Zhao and Rosson research the reasons and forms of using microblogging in order to explore microblog's poten-

tial impacts on informal communication at work. They bring in the idea that informal communication among employees (e.g. water cooler conversation) "play an important role in organizations success and provide a variety of potential benefits supporting for collaborative work" [15]. From this point of view microblogging is a potential tool for collaboration in the working-context. In that case, the generated messages from microblogging could be an important source for new requirements. The requirement engineer could analyse an archive of an implemented internal microblogging server like statusnet [11]. From this analyse he might elicit informal opinions and requirement. Reinhard supports this argument: "Within the last two years a novel mode of communication emerged with microblogging, which could also be advantageous for the use in distributed and co-located software teams as it can be saved persistently and is further kept out of restrictions of formal communication" [8].

4.4 Mobile unit tests

Another idea is the combination of the software development environment with the requirement elicitation process. The idea is that the requirement engineer creates for every software-module idea a Unit test. This could make the whole software generation process more efficient and cheaper. One big advantage of this could be that the communication between requirement engineers and developers take place on the same medium or platform. The requirement engineer formulates his requirements in the same programming language than the developers. The requirement engineer formulates every requirement as a unit test of an abstract function. The implementation of the function and the whole architecture is then up to the developers. This makes it possible that no translation from one to another medium has to be done. The requirement engineers and the developers are working on the same platform communicating the requirements for the building software.

The usage of unit tests to formulate requirements in the requirement elicitation phase would insure that the development by contract ("Vertrachssicherheit") is insured later on in the development. Further this could help the requirement engineers to better understand under which conditions the developers have to work out the solutions for the requirements, because they are always in contact with the software-code. This helps that the software is developed with less pressure because developers are actually able to implement the requirements.

When this technique is combined with a tablet PC the analyst has a mobile tool that supports him in formulating the requirements. He can show the customer on the display the prototypes that have already been developed and the functionalities that are already guaranteed. At the same review the mobile analyst can negotiate new requirements and write them, with his tablet pc, as new unit tests in the code.

Here is a scenario of this idea: The analyst uses a mobile device. The mobile device is in permanent communication with the subversion system (code check-in and sharing) and servers of the software development team. The development team and the requirement engineer have been working together for 2 years and

have successfully completed 3 projects together. The requirement specification was the beginning of each project. First the requirement analyst participated in a public tendering where he listened all the tasks that the customer has written out. It's the analysts task to translate the vague ideas of this "Pflichtenheft" into tasks for the developers. But the problem here is that the analyst doesn't know exactly which functionality can actually be done by the developer team. What he knows is what has been developed already in the past. But since new software usually has to grow in its functionality the analyst doesn't know how well the developers are able to grow their abilities during the period of the software project. Therefore he has to be careful that he doesn't write down requirements in the requirement specification that the team won't be able to fulfil. That's why he needs a way to communicate with the customer and the developers on the same platform. The analyst sits together with the customer and participates in the working context. Now, instead of writing down requirements of which he's not so sure whether they can be fulfilled, he checks in unit tests of the future system. With a tool that provides an interface for the analyst, that helps him to match his requirements with the code of the developers, the software engineering will be more a team work by interfaces instead of pushing pressure of the analyst down to the developers.

5 Discussion

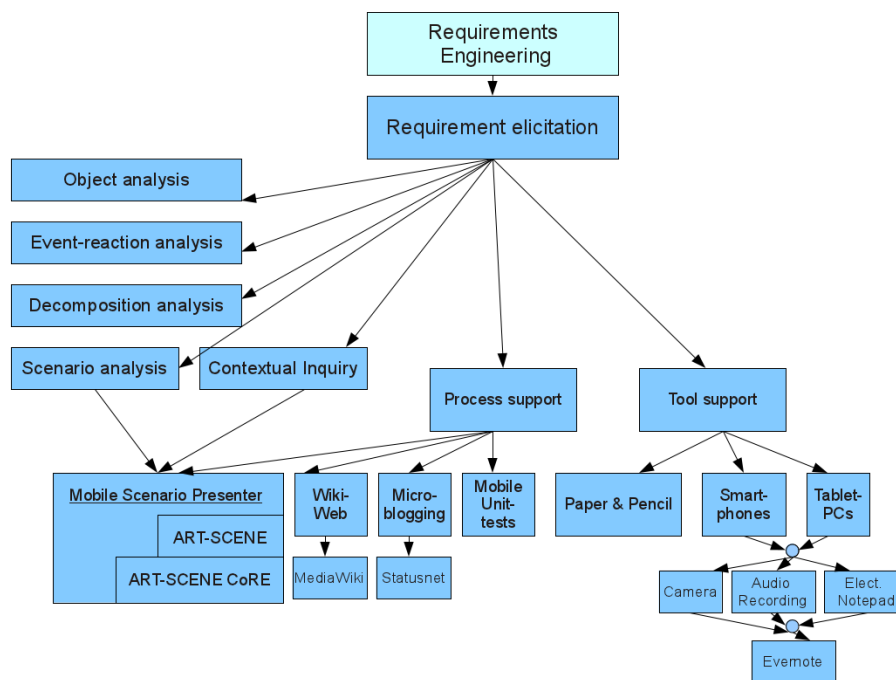


Figure 2: Overview of tool & process support for the mobile analyst.

The possibilities and functions of electronic devices are fascinating and cunning. However new technology does not necessarily perform given tasks better. As soon as new technology is invented it is vital to research the reasoning for applying them. Matching new technologies with appropriate scientific methods can help. It has to be discussed under which conditions and for which purposes the tools should be applied. To investigate this, we take a look at advantages and disadvantages of the techniques and methods.

5.1 Electronic devices contra paper

Advantages of electronic devices Electronic devices have the advantage that requirements can be typed down on machine-letters and thus are easier to read. Every requirement can be processed automatically. Videos, audio files and notes can be linked together to a user-story. Further every requirement has an automatic time stamp which makes it possible that entries can be sorted. Due to interoperability the requirements can be transformed from one medium to the other which makes it possible that they can be observed from different viewpoints and terminals.

Advantages of paper One of the biggest advantages of eliciting requirements with paper is its simplicity and speed. When working with a customer its much faster to take paper and pencil and write down what he says, than to turn on an electronic device and type in the requirements. It also has to be taken into account that the development of an electronic tool, its distribution, its installation, the workshop of its usage and the actual usage at the context rise their opportunity-costs. The fact that it works without electricity contributes positively to the ecological environment.

Short summary of paper advantages:

- Has lower opportunity-costs
- Higher simplicity of its usage
- Has a higher ecological sustainability

5.2 Enlightening in requirement Elicitation with the Mobile Scenario Presenter

Glinz defines: "Requirement Engineering is always also enlightening of creativity at all participants" [3]. All efforts in gathering requirements will be waste if they don't inspire developers, analyst and customers to work and use the developed software. The more the requirement engineers inspire the developers in the software generating process the more it will inspire the customer to use the software in the end.

The Mobile Scenario Presenter has a great ability to inspire analysts to find the right requirements. In comparison with the paper version, it appears much more professional to go through a list on a mobile device with the customer. This

will enlighten the customer and he might trust more in the developing process. When the customer trusts more in the process of the requirement engineer, he has more motivation to provide time, resources and the right information to the analyst. As well, the communication basis and professional relationship among the business partners and all working participants might be more settled. Showing up with a mobile scenario presenter sets out as well another signal: The scenarios have been prepared and put already in a software. The more the analyst can present a good and professional usage of software driven devices, the more he might understand the problems other software has to solve.

5.3 Comparisons & drawbacks of mobile unit tests

The great problem of the idea with the mobile unit test is that there exists no implementation proposal. There is also almost no discussion about it in the academic field. There must be great drawbacks of this idea. The greatest drawback of it is most probably that it is very difficult to present unit tests to the customer in an understandable way.

Comparing this approach with our requirement engineering methods, it can be said, that the check in of unit-tests is probably a method of the decomposition analyse. Because a problem is splitted into tests that are written before the code is actually generated. This means a problem is recognized and will be decomposed into partial problems by the developers. Coming back to our old example: The user of the ALP- touring- navigation- system has to be able switching restaurants during the trip and calculate a new route. The unit-test approach will generate a test case before the developers start implementing and thus before the actual programming- problems of that method occur.

5.4 Lessons learned from mobile devices

The experience [9] has shown that the total number of requirements can be increased with ART-SCENE CoRE. Total number requirements documented, number of requirements on normal course, number of requirements on alternative course and average number of requirements per normal course event were measured. From these metrics other knowledge can be concluded. The first lesson learned was that the number of requirements is dependent on how well the analysts are familiar with the environment. This was a problem in the field because since the team was testing the requirements for a requirement eliciting method they probably didn't have the knowledge about the context of ski tours. For practitioners this can be a big issue. Using a requirement technique such as the Mobile Scenario Presenter practitioners need to have the contextual knowledge of the working process. Then battery problems most probably would not occur. Nevertheless these experience have shown that it always stay possible to elicit requirements with the classical methods and therefore are a good alternative. The biggest lesson learned, as it appears, was that audio recordings speeded up the process. This method can be compared with the old dictation machine method where each occurrence was logged. The disadvantage surely is

that for the workshop the requirements are not there in written form and that they can not be validated as fast, as if that amount of work was already done when eliciting the requirements.

Talking about establishing stakeholder needs with mobile technologies there are 6 lessons learned in research. They were revealed by experiencing mobile RE tools such as the Mobile Scenario Presenter and are shown in Table 1.

#	Lesson	Quote
1	Usage	"Use mobile technologies in the workplace to discover requirements the future end users needs."
2	Range	"We need a range of mobile requirements tools."
3	Different needs	"Consider the needs of different users such as analysts and future system users."
4	Usability	"Usability is essential for mobile requirements tools."
5	Dependence	"Carefully plan your use of mobile RE tools in advance to lessen your dependence."
6	Complete specification	"Capture enough information about a requirement to enable its complete specification."

Table 1. Reported lessons for determining stakeholder needs with mobile technologies by Maiden et al [6].

Maiden et al conclude these lessons with: "You can use the reported lessons learned as an initial guide to develop and use mobile RE tools successfully. We believe that mobile RE tools will complement rather than replace traditional approaches and the combination of context-aware and conventional elicitation and negotiation approaches has the potential to improve the quality of requirements" [6].

6 Conclusion

This seminar paper has shown several techniques and processes that support the mobile analyst. It has elaborated and compared paper & pencil, smart-phones, tablet-PCs as mobile tools and the mobile scenario presenter, wiki-web, micro-blogging and mobile unit tests as mobile processes. Paper will be an original, long lasting technique with three big advantages: Low opportunity-costs, high simplicity of usage and high ecological sustainability. For each new electronic technique methodical reasoning should be undertaken to evaluate their fit for purpose. For mobile unit tests there exists no implementation so far and it has great disadvantages. Further research could try to falsify this. The Mobile Scenario Presenter is the strongest of all techniques because it has the greatest power of enlightening the mobile analysts and customer.

References

1. <http://www.evernote.com>
2. Ferreira, D., Rodrigues da Silva A.: Wiki Supported Collaborative Requirements Engineering. In: Wikis4SE08 Workshop September, Porto. Portugal (2008)
3. Glinz, M.: Requirements Engineering I. In: Script University of Zurich. Zuerich (2009)
4. Glinz, M.: Informatics IIa Modelling, Chapter 10.3, Interaction modelling. In: Script University of Zurich. Zurich (2010)
5. Jiang et al.: A methodology for the selection of requirements engineering techniques. *Softw Syst Model* 7:303328. (2007)
6. Maiden, N., Seyff, N., Gruenbacher, P., Otojare, O., Mitteregger, K.: Determining Stakeholder Needs in the Workplace: How Mobile Technologies Can Help. In *IEEE Computer Society* vol.24, No.2. (2007)
7. Maiden, N., Seyff, N., Gruenbacher, P.: The Mobile Scenario Presenter: Integrating Contextual Inquiry and Structured Walkthroughs. In: 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'04), pp.115-120. (2004)
8. Reinhard W.: Communication is the key Support Durable Knowledge Sharing in Software Engineering by Microblogging. Proc. of the SENSE Workshop, Software Engineering. University of Paderborn, Institute for Computer Science. (2009)
9. Seyff, N., Graf, F., Maiden, N., Gruenbacher, P.: Scenarios in the Wild: Experiences with a Contextual Requirements Discovery Method. City University London.
10. Seyff, N., Gruenbacher, P., Maiden, N., Tosar, A.: Requirements Engineering Tools Go Mobile. In: Proceedings of the 26th International Conference on Software Engineering (ICSE04). (2004)
11. <http://status.net/>
12. Svoboda, T.: Tailoring Scenario Walkthrough Tools Towards Current Software Engineering Needs. Johannes Kepler Universitaet Linz. (2009)
13. Syed, A. F., Ahmad, I., Ho-Jin, C. "Enhancing Requirements Engineering Activities through the Use of Mobile Technology Devices and Tools." *FgcN*, vol. 2, pp.578-581, Future Generation Communication and Networking (FGCN 2007) - Volume 1. (2007)
14. Hughes, J., O'Brian, J., Rodden, T., Rouncefield, M., Sommerville, I.: Presenting Ethnography in the Requirements Process. In: Proceedings of the Second IEEE International Symposium on Requirements Engineering (RE 95). (1995)
15. Zhao D., Rosson M.B.: How and why people twitter: The role that Micro-blogging plays in informal communication at work, Pennsylvania State University (2008)