

Expanding Requirements Through Observation: An Experience Report

Nina Desnica¹, Gil Regev², Alain Wegmann²

¹desnicanina@gmail.com

²Ecole Polytechnique Fédérale de Lausanne (EPFL), School of Computer and
Communication Sciences, CH-1015 Lausanne, Switzerland
{gil.regev, alain.wegmann}@epfl.ch

Abstract. In this paper, we describe our use of ethnographic techniques for identifying requirements for a web-based system for automatic updates and data exchange in a financial investment company. During a six-month long internship, we observed stakeholders in their daily work, helped them use the system, and discussed the results of our observations with them. This helped us to define a more complete set of requirements than if we would have used only the traditional interview and workshop techniques. We describe the observations we made, analyze the results and discuss the key learning points.

1 Introduction

The work described in this paper was performed as a six-month long internship at the Multi-Asset (MA) team of a financial investment company based in Switzerland. The project goal was to improve two processes: data exchange and updates of financial reports. Apart from analyzing the initial requirements we decided to focus on the social aspects of work because we wanted to fully meet the users' needs [1]. We loosely applied contextual inquiry, an ethnography-based requirements elicitation method that was taught the previous year in the Enterprise Architecture course [3, 4]. Our approach consisted of maintaining a continuous presence with the users so that we could observe their daily work, identify the difficulties they faced and help them solve some of the technical problems. We identified new requirements by discussing the results of our observations with the users.

This approach helped us to propose requirements that were a priori unknown, and which significantly changed the scope of the project. It also brought new perspectives to the development process used in the company. The use of ethnographic techniques for defining requirements is not new. However, they are hardly ever used in practice for this purpose, mainly because they are so time consuming. Based on this work, we believe that internships can provide a good opportunity for using these techniques in companies.

This paper is organized as follows. In Section 2 we present the initial project scope. In Section 3 we describe the observation techniques we used. In Section 4 we describe

how we carried out these techniques. In Section 5 we discuss the results of our work on the product and on the requirements engineering process. Finally, in Section 6 we conclude the paper.

2 The Initial Project Scope

The MA team consists of 15 people: portfolio managers and quantitative analysts. The MA team manages equities, currencies, funds and commodities, thus structuring collected data with the main goal of providing the best investment strategy for their clients. One of the team's main challenges is to gather the needed data in order to provide customized services to clients based on the performance. The team uses a systematic investment methodology that is based on a set of economic indicators. These indicators are not only used by the MA team but also by other teams within the company, and with clients.

The intern will be responsible for the implementation of macroeconomic indicators defined by the MA investment team. These indicators will be implemented using a new database system (non-sql database coupled with a finance-specific query system) and represented in the team's intranet for access to the investment specialists of the firm and selected clients

Figure 1 Initial Project Proposal

Finally, the most important part is the creation of a web-based application which has the main purpose to enable the team members to exchange the information conveyed by the set of indicators. Furthermore, it should improve the productivity of the whole team; make the work easier and communication more efficient. The interface should be a user-friendly setting and it will need to show the latest values on the firm's intranet portal. The web portal will be a new system that should be built according to the analysis of the difficulties that employees are facing every day. The intern should start from observing, analyzing the problems of employees, localize and indicate the main issues and then explore and define ways to overcome this, and finally implement the solution and shape into a web-portal that can fulfil employee's needs and meet their expectations.

Figure 2 Excerpt of final project proposal

Seeking to improve their operations, the MA team launched a project to design a new web-based support system. The initial project proposal is shown in Figure 1. The initial project scope is defined, with a brief explanation of the main goal of the project. The project was proposed as internship to the first author of the paper who then sought the academic support of the two other authors. These two authors, being heavily in favor of the use of ethnographic techniques for requirements elicitation proposed that (a) the project summary be expanded with more details about the technical work expected from the intern; and (b) that the intern will use observation methods to identify every-day problems faced by users. The result can be seen in Figure 2.

The use of observation (ethnographic) techniques is not mainstream in the IT industry. We were fortunate that the MA team accepted to use these techniques, which are often seen as overly time consuming.

As can be expected from a project in such an early stage, when the intern began her work in the MA team, there were no specific requirements. There was only a document that explained the content that was to be displayed by the web application. The next section describes the ethnographic methods we used to elicit more requirements.

3 THE ETHNOGRAPHIC TECHNIQUES WE USED

The requirements elicitation period lasted two months. It involved different activities such as interviews, informal conversations, observing users work and helping them solve their problems. We worked with 8 out of the 15 members of the MA team. These eight members were five portfolio/investment managers and three quantitative analysts. We also worked with four users from other departments who collaborated with the MA team.

All the interviews were one-on-one, face-to-face, in person conversations. We asked questions related to the work the user does, situation he or she faces, feelings on the specific problems and feedback on proposed ideas. The interviews usually occurred at the user's desk. We were always taking notes during the interviews and the users were aware of that. Informal conversations were more relaxed conversations with users. These conversations occurred in different locations such as the cafeteria and common rooms. They occurred during breaks and we have never taken the notes in front of the users. The purpose of these informal conversations was to enable sharing data in a casual way. This would help the users to relax and share their opinions. The observation was unstructured. It was meant to spontaneously understand the behavior of the users at their workplace. We recorded what we saw during or after observations. Sometimes users came to us for help, which gave us the unique opportunity to gather more data related to concrete problems they were facing. After these activities, we wrote down everything we could remember.

Frequently we combined the different activities in order to obtain a more precise understanding of the everyday work of users. For example, very often an observation was followed by an interview or an informal conversation with the user. We spent a few hours every day observing the workplace and employees. The MA team works in an open-space office. It was therefore convenient for us to discretely observe colleagues unnoticed.

4 Observations

In this section we describe the situations we observed. For each situation, we explain the context, the succession of observations, the resulting requirements and some reflections on the method.

Observation 1: Using post-it notes on the computer screen

While we were watching around the workplace we spotted the following situation:

Lana, a portfolio manager, had a few post-it notes attached to her monitor. They were fastened with scotch tape

Post-it notes tend to fall-off with time. When people fasten them with scotch tape, it is often an indication that they want them to stay in place permanently. We asked Lana why she uses the post-it notes and what is written on them. She replied:

“I keep there just a couple of names of the funds that I use very often.”

This answer made us realize that this information could be added to the web portal. We asked Lana if this was a good idea and she said *“Yes, absolutely!”*

We made this into the following requirement:

Req1: The system shall display the data about the most frequently used funds.

Observation 2: Report updates

During the first few days of the observation process, we noticed the following situation:

Every day Mark, one of the MA team members, came to the office a little earlier than the others. He would update all financial reports manually and ran the calculations based on this updated data. Finally, he sent an e-mail with a notification to the whole team.

At the beginning of the project, we already knew that this process should be automated because it is tiresome, slow and may be the source of errors. We decided to ask Mark to explain what he does and how he does this every morning. We engaged in an informal talk. Here is a summary of what we understood in this discussion:

First, he was looking for fresh financial data on the special financial software. Next, he copied this data and added it to the Excel sheet. Next, he refreshed the calculated formulas and repeated this process for the six additional reports. Finally, he sent the e-mail to the investment team's mailing list to notify the team that all reports were updated. This process took Mark about 20 minutes and he felt that was very inefficient.

Using this process and the data we obtained, we decided to ask three additional investment managers for their opinion about this process and whether they find it useful.

They all agreed that this process is very important and that they could not start anything without refreshed data. One of them pointed out that not everybody needs all the reports, so each one of them named three of the reports they use.

We found out that this process is crucial for the productivity of the team. Also, we realized that one person is doing all the updates for everyone else. This made us add the following requirement:

Req2: Automated updates through the system

The users in charge of the updates pointed out the following:

Some updates took much time; it would be good if we could have some text that indicates the phase of the update process.

We responded to this request by implementing a multi-line textbox that notifies the user which phase of the update is currently on (Figure 3). After the first prototype, we decided to test it by giving it to the users to try it out. The reactions were very positive.

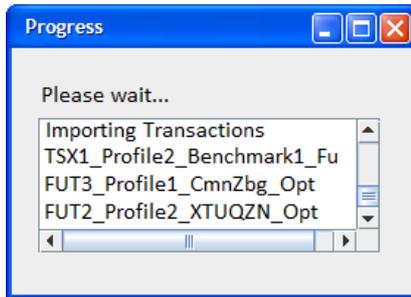


Figure 3 – Multi-line textbox showing the phase of the update process

Observation 3: Checking progress of updates

After implementing this feature, we continued to observe the users and saw an interesting pattern:

Users do not really follow the task progress closely. They minimize the application until the end of the process or occasionally take a look at the application window.

Using this observation, we realized that, it would be more useful if the users had a simple progress bar so they can quickly check how much time is left until the update is completed. This idea was converted to a requirement (Figure 4):

Req3: The system shall display progress bar while update processes are running

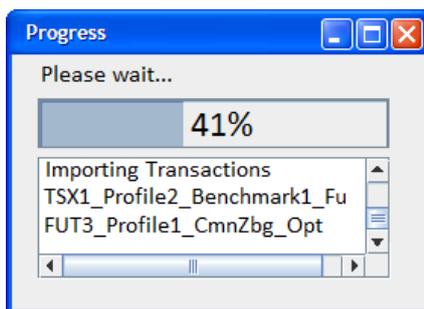


Figure 4 – Progress bar added to the multi-line textbox

Having in mind that users usually minimize this window, we also implemented a notification in the form of a message box that pops up indicating that the update

process has completed. This is linked with requirement 5 explained in Observation 4.

Observation 4: Difficult search of right documents

Sometimes, users asked us for help with some IT problems. While helping them we noticed that some users were having difficulties finding the documents they need:

Jade, one of the managers, asked if we know where she can find the reports about data analysis for the following day, because they were not where they used to be.

Jade showed us the folder that usually contains daily reports about data analysis:

Jade found the folder mentioned above by opening around five folders, one after another. When she opened the folder that contained the reports she needed, she discovered that the latest reports were four days old.

We realized that Jade was facing two problems. One was that the location of the document she was looking for has changed, or the documents were not updated. Another problem was the complicated structure of files and folders, which was not an obvious problem for Jade. We hoped that if we ask Jade about this problem it will become more obvious. We asked her whether she would like to have a better structure of the folders. She replied:

“Well, I guess, I don’t know, I just want to be sure that the latest document is on specific place where I can find it.”

We noticed that she was obviously worried only about the first problem. In order to solve the first issue, we had to ensure that all documents will be up-to-date and on specific location. We already ensured that the documents will be automatically updated instead of manually which guaranteed that documents will be always updated. However, the problem was that the reports were updated but placed in a location where Jade could not find them. We asked Jade if she would like to have a possibility to choose a personal folder for storing the reports:

The feedback was really positive and Jade was excited about the idea.

The problem led to two new requirements:

Req4: Users shall be able to choose a folder where they want to store the report for the following day.

Req5: System shall open the file location upon the end of the updates

However, we decided to elicit the following problem further.

Observation 5: Searching files through nested folders

Even though Jade did not point this out as a problem, a better folder structure would help her to locate the reports much faster. For Jade the process of going through five levels of folder hierarchy was normal and usual. She was not able to see this as a problem. Consequently she could never mention this as a requirement. This led to the

following requirement:

Req6: The system shall display all the documents at one place and enable users to search files they need by using the date and the name of the file.

Even though we enabled users to choose a folder for the updated reports (Req4), it is still not easy to find the correct file because a folder usually contains several hundred documents. Compared to Req5, it is important to mention that users update the reports once per day, whereupon the system opens the file location, whereas they access reports several times per day. Hence the search by date and name of file in Req6.

Observation 6: The use of SQL code

This situation occurred when the one of the quantitative analysts was on vacation. Therefore, one of the investment managers asked us for help with a database.

George, the portfolio manager, asked us to put data about a financial transaction into the database.

We found a tutorial written by the quantitative analyst, which explains step-by-step how this task should be done. The steps included the construction of SQL statements. We showed these instructions to George, who said:

“I saw that tutorial and I read it, but I cannot understand it. It must be something basic and easy but I do not have any knowledge about it as I have never been trained in SQL.”

Based on this conversation, we solved George’s immediate problem by adding the data he needed into the database, but we needed to solve the more general problem we identified, that the system forced non-IT users to write SQL statements. Based on this discovery, we concluded that this process should be simplified so that users are not required to have SQL knowledge. Thus, we made this into the following requirement:

Req7: Provide forms that query the database without requiring users to use SQL code

Using this requirement, we developed a new feature. On the web portal, there is a section where every user can add any data into the database, simply by inserting details in a form and clicking the submit button.

Observation 7: Graphical data representation

One of the initial requirements was a graphical representation of financial indicators. We observed that every day investment managers were plotting graphs from the updated data in Excel files. In order to automate this process, we made it into two customer requirements:

Req8: Interactive graphical representation of indicators

Req9: Users shall be able to download graphs directly from the web portal

We asked one of the investment managers if she can show us how she does it.

Sarah opens the Excel file and updates the data, then plot the graphs. In the Excel

files there were three graphs for the same indicator. When we asked why, she replied: "Well, sometimes I need to see a graph based on a three-year dataset, but sometimes I would prefer it based on an annual one."

We decided to check whether other users had the same needs. We asked two other investment managers the same questions.

Observation 8: Choosing dataset for the graphs

The investment managers told us they used three different datasets: one, three and five years. This observation was converted to a requirement for a dynamic graphical representation of indicators:

Req10: Web portal shall enable user to choose dataset for generating graphs depending on the timeframe

This was implemented in the web portal as a new feature. Users are able to switch datasets for one, three and five years by pressing a button.

Observation 9: Missing necessary information

Communication among the investment team was done mostly through e-mails, direct discussion and meetings. When it comes to the reports discussion it is important to mention that comments by the investment managers are very valuable. The discussions are usually done in the office where everybody takes notes about what other people said. During the observation period we witnessed the following situations many times:

Somebody is asking for a clarification of the comment on the report from the last month, but the person who has the information is not in the office.

It was very important but hard to keep track of all the comments. We realized that information exchange about reports could be improved in many ways. Based on the observations we proposed that comments on reports should be one of the features of the web portal:

Req11: Enable posting comments on every uploaded file

Req12: Automatic emails

Table 1 shows the summary our observations, our interpretation of the problems faced by the users, the resulting requirements, and the follow-up observation that led to further requirements if any.

Observation	Interpretation of the problem	Requirements	Further Elicitation
<p>1. Using post-it notes on the computer screen We spotted that manager glued post-it notes by scotch tape to make them permanent</p>	Lack of visibility of needed info	Req1: The system shall display the data about the most frequently used funds.	
<p>2. Report updates Watching the manager doing updates by hand we realized how they should be automated</p>	Manually updated reports sometimes cause errors Slow, tiresome process	Req2: Automated updates through the system	Observation 3.
<p>3. Checking progress of the updates Users do not watch the update progress continually. They continue working and occasionally check the progress.</p>	Lack of the progress indicator for the report updates	Req3: The system shall display progress bar while update processes are running	
<p>4. Difficult search of right documents We were asked by a manager to help her to find a document</p>	Difficulties finding documents Unknown location of a needed document	Req4: Users shall be able to choose a folder where they want to store the report for the following day Req5: System shall open the file location upon the end of the updates	Observation 5
<p>5. Searching files through nested folders The manager was going through several levels of nested folders in order to find the document</p>	Data is not easily accessible	Req6: The system shall display all the documents at one place and enable users to search files they need by using the date and the name of the file	
<p>6. The use of SQL Users could not use the tutorial that was provided to them in order to enter data in the database</p>	Updates of the database are done by writing SQL queries	Req.7 Provide forms that query the database without requiring users to use SQL code	
<p>7. Graphical data representation Observing the managers plotting the graphs for the financial indicators</p>	Difficulty obtaining three views of the data depending on the time frame	Req.8 Interactive graphical representations of the indicators Req.9 Download graphs directly from the portal	Observation 8
<p>8. Choosing dataset for the graphs We noticed that managers were plotting graphs for three different periods of time</p>	Lack of possibility to quickly change the dataset and views	Req.10 Buttons for changing dataset depending on the time frame	
<p>9. Missing necessary information Many people complained that they cannot get the right comment on some report because it's hard to keep track of all of the comments during discussion</p>	Lack of repository linking reports with comments.	Req.11 Posting comments on every uploaded file Req.12 Automatic e-mails	

Table 1 Summary of observations and requirements

5 Discussion

We believe that if we had not used ethnographic techniques we would have been unable to capture the details of the everyday work and challenges faced by users. For example, during Observation 1. If we had not very carefully observed the working environment, we would not have been able to see the details and significance of the post-it notes. Some observation results led to further observations where we discovered issues that were even less apparent on the surface. For example, Observation 4. The investment manager complained about the location of the file he was searching for. If we would not have asked him to show us how he searched for the document, we would never have been able to spot the other problem that he was facing. It is similar with Observation 2. We discovered the additional need of the users after watching how they interacted with the application. This is a very good example that proves that if we had not used an ethnographic approach and watched over the shoulder of the users, we probably would never have found out about this.

Without direct observation it would have been difficult to identify the problem in observation 6, where the non-IT users needed to use SQL code in order to add data in the database. We might have found it by studying the tutorial, but often these documents are not stored in a central repository where they are readily available. In these cases, even finding the document requires observation and time. In this specific case, it is quite probable that if the quantitative analyst were present, he would have solved the data entry issue easily and we would have never have been able to observe this problem.

This case showed another important aspect. The user asked us to help him enter the data. This provided us with a magnificent opportunity to: a) solve his immediate problem; b) find the tutorial and understand that he couldn't use it; c) establish a relationship of trust with the user because we were able to help him. This last point is probably the most important because a good relationship with users is very valuable.

Helping users to do their work goes beyond the apprenticeship model proposed by Beyer and Holtzblatt [1] where the user is the master and the designer is the apprentice. But designers, in this case the intern, may be experts in using the system. They can then help the users to use the system, therefore building a two-way relationship rather than the one-way master-apprentice.

In observation 8 the key point is that we spotted the problem because we asked the user to show us the process of plotting the graphs. The obvious requirement was the need for automated graphical representation of the data, but again we spotted another problem that was not that obvious. If we would not have observed very closely the update process, we would have never spotted the second problem.

About half of the observations and resulting requirements are strongly related with the ergonomics of the system. This is not surprising because the use of ethnography in IT has mostly been in this field, e.g. [1, 2]. Most other observations and requirements are related to content needed by the users. One observation and requirement has to do with the need to shield users from the technical details of the system.

One of the advantages we had in this work is that the intern was a developer, therefore able to understand the system on the one hand and users and their needs on the other. By spending time in the open space and observing users she was able to spot many aspects of work that users were not able to express. The observation

sessions were less structured than contextual interviews as defined by [1, 2], but they did enable the two-way relationship we discuss above. Whereas Beyer and Holtzblatt recommend that the designer moves away as quickly as possible from the inverse relationship, we found it useful to be able to help users and therefore foster the very partnership that Beyer and Holtzblatt seek [1, 2].

This work bears resemblance to the early work on ethnography and requirements engineering [5], where lightweight ethnography techniques for engineers were developed. As envisioned in [5], it is the intern, a would-be engineer, who did the ethnography work without an ethnographer as proxy.

6 Conclusion

In this paper we described a project where an intern was given the opportunity to observe the work of users in order to improve the requirements of an IT system. During formal interviews, many users indicated needs that were difficult to understand. The observation helped us to better understand these needs and to identify additional requirements.

In most organizations it is impossible to have an engineer observe users in the way that is described in this paper. By just being there. Engineers are costly resources. It is difficult to invest their time in such seemingly wasteful activities. That said, many organizations would happily use interns for this activity, their time being seen as less costly. Universities should make encourage industry relationships to help their students find such internships and then coach them accordingly. For many engineers it is the first and last time they could really use ethnography skills. For companies it can be a very valuable use of the intern's time.

7 References

1. Beyer, H.R. and Holtzblatt, K.: Apprenticing with the customer. *Communications of the ACM*, 38(5), pp.45-52 (1995)
2. Beyer, H. and Holtzblatt, K.: *Contextual design: defining customer-centered systems*. Elsevier (1998)
3. Regev, G., Regev, L., Na'im, Y., Lang, J. and Wegmann, A.: Teaching an ethnographic approach to requirements elicitation in an enterprise architecture course, *CEUR Workshop Proc.*, vol. 1374, pp. 5–19, (2015)
4. Regev, G., Gause, D.C. and Wegmann, A.: Experiential learning approach for requirements engineering education. *Requirements Engineering*. vol. 14. pp. 269–,287. Springer (2009)
5. Viller, S. and Sommerville, I.: Social analysis in the requirements engineering process: from ethnography to method. In: *Proc. IEEE International Symposium on Requirements Engineering*. IEEE (1999)