

# Use of Sense-Making Methodology in a Requirement Process

**Georg Strom**  
Gormsvej 1  
DK-4320 Lejre, Denmark  
+45 40 18 12 72  
strom@georg.dk

## ABSTRACT

This paper describes a field study where Sense-Making Methodology was used in an actual IT-project to elicit user needs that were transformed into requirements. It shows how it was possible to create a complete set of requirements, where the background and need of each requirement clearly could be traced, and how it in particular was possible to transform the elicited user needs into requirements in a rapid and reliable manner.

## Keywords

Interviews, Sense-Making Methodology, Requirements

## INTRODUCTION

The most important activity in a project is to define the outcome that is required of it. Only then can it be decided whether the project is necessary and feasible and ensured that the results of the project are beneficial or creates value.

Koen [5] presents the following heuristic: Allocate resources as long as the costs of not knowing exceed the costs of finding out. This means that the definition of the required outcomes of a project shall continue, until the cost of continuing is higher than the expected value of the additional insight into the required outcome.

However, it requires some knowledge of the required outcomes to identify any lack of information and the possible value of acquiring it, so it may be difficult to argue for requirement work before at least part of it is done.

The first part of the definition of the required outcomes consists of collecting information about the needs of the organisation and future users. If the results of that part are not valid and reliable, the resulting requirements may be misleading. One particular problem is that users may forget to mention existing valuable functions or routines that they have become accustomed to.

The second part consists of the transformation of the collected information into formats that may be used in the

project. This is more complex than it may seem, because the information may be needed for different purposes:

- It may be needed to guide a further investigation, as it becomes clear which additional information that is required.
- The required outcomes may be used as basis for a business case or of other evaluations of the possible benefits of the project.
- The required outcomes may be used to evaluate whether the project is feasible, for instance whether any available software can support them.
- Part of the required outcomes may be included in a contract between a supplier and the customer and used as a legal document.
- A supplier may use part of the collected information as input to design a system that fits the precise needs and situations of use.
- The collected information may be used to evaluate the value of different outcomes of the project, if it is necessary to prioritize the requirements.

## SENSE-MAKING METHODOLOGY

In this study, I have investigated how Sense-Making Methodology (SMM) developed by Dervin and other [3] could be used to collect and process information about the required outcomes of a project.

I have used SMM in earlier projects [7, 8, 9], and found that it offers several advantages. One of them is, that it is based on a number of explicit assumptions so it is easier to discuss its suitability in a particular project.

The following description of the assumptions is based on a paper that for now is the most complete description of the methodology [2].

SMM [2] is based on the assumption that humans may work inside, outside or against the structures surrounding them. A person may for instance use a system as it is supposed to, decide not to use it, or use it in an unplanned and unforeseen manner, depending on his or her actual goals and ideas. This makes it possible to analyse conflicts between how a user is supposed to do a task and how he or

*LEAVE BLANK THE LAST 2.5 cm (1") OF THE LEFT  
COLUMN ON THE FIRST PAGE FOR THE  
COPYRIGHT NOTICE.*

she does it, or between the organisation's and the user's personal goals for a task.

Another of the assumptions is that humans in general move step by step through time and actively try to solve the problems that block their progress [2]. SMM focuses on the thinking and problem solving that are part of an activity, and makes it possible to describe the large range of tasks where users adapt their use of an IT-system to a specific task and situation. This encourages a better understanding of how the user's work can be supported, than when using contextual enquiry [1] which aims at documenting fixed procedures that are done repeatedly.

A core part of SMM is a theory of the interview with its own set of assumptions [2]. This makes it possible to get valid and reliable information about how specific users think and act when solving specific problems. It is a major advantage compared to Weick [10], who has a similar focus on mental activities in his Sense-Making. However, he does not have a way to study the mental activities, so his descriptions appear often to be post-hoc rationalisations similar to "The person did this, so he had probably thought that".

During the interview, the problems and how they are solved are seen from the perspective of the participant, without trying to restrict them to the perspectives of the interviewer or the planners of the study [2]. Even though the interviewer and the planners need information for designing a new IT-system, the participant is encouraged to tell about the whole work situation, and not only about his or her experiences using existing IT-systems. I have seen in my previous studies [9] how that inspired new creative solutions, because it made it easy to identify unfulfilled needs. I also found [9] that it tended to make the interviews highly effective. In most cases the participants told first what they felt were the most serious problems, so I got an almost prioritised list of problems from the interviews.

In addition, it is in accordance with the democratic and equalitarian ideals of many Danish workplaces, to see the problems from the perspective of the participant, and not the interviewer.

SMM [2] is based on the assumption that all communication is designed, no matter whether the design is done spontaneously or consciously. The interviews are therefore explicitly structured in a manner that helps the participant to articulate an insight in the problems he or she experiences. I have experienced during my previous use of SMM, that such a structure is similar to the sort of active listening we expect in a normal conversation, so it feels natural and encourages the participant to talk.

The participant [2] is invited to go over the same experience several times, so he or she gradually reveals more aspects of it. During my previous use of SMM I found this offered an additional advantage: As interviewer I could get an early overview of a number of different topics

that the participant wanted to tell me about, so I later could bring them up and avoid that the interview got stuck in the details of one of them.

## **METHOD**

The study was done in the municipality of Copenhagen. It is the largest municipality in Denmark, with more than 14,000 estimated computer users. The municipality has established a Koncernservice with about 200 employees. It is clearly a professional and capable organisation and responsible for acquiring and operating most of the municipality's IT-systems.

During the study I used SMM to collect information about the needs of the intended users of a future IT-system and transformed the information into formats that could be used for evaluating and preparing acquisition of the system. I acted throughout as a participant-observer [4] without having any special influence on whether or how the results of my work were going to be used.

I wrote equal to 21 typewritten pages of continuous notes during the study. They captured the progress of my work and my experiences during the study, and I have used them as a basis when reporting the results in this paper.

## **RESULTS**

I had earlier been in contact with Koncernservice. They had become interested in what I told about SMM, and I was invited to discuss possible consultancy work.

They were evaluating the value of and preparing to invite proposals for a system to simplify users' access to the range of IT-systems used in the municipality. The background was anecdotal evidence that some users had to handle different user-ids and passwords to more than twenty IT-systems, and that they spend a considerable part of their workday logging in and out of them.

A major part of the evaluation and preparations was a proof-of-concept test where two competing access systems should be used for a period in actual work situations. The project group had planned to do workshops or focus groups to collect requirements, but I understood that it might be difficult to find time for the workshops when all required participants would be available. So one argument for my work was that individual interviews were easier to arrange than work-shops.

We decided I should interview a number of technicians in the IT-support, to ensure that the access system would be easy to maintain for them. We also decided that I should interview users from three groups: The so-called citizen service centres, where the employees had to log into and use a larger number of systems than any other user group, and where they had to provide direct service to citizens while using the different systems; The city-planning department whose employees had highly varied tasks that required processing power and a number of special applications; Finally a number of nursing homes that had a

large number of intermittent users with limited computer skills.

I decided to use so-called micro-moment time-line interviews from SMM [2], where each participant is guided to tell about a situation where a problem has occurred, the actual problem, how it was overcome, and what the outcome was. During the first interviews I found that I had to divide the last point into two: What the participant found might make it easier to solve the problem – that was often valuable to know when defining a requirement to the access system – and the consequences of the problem, for instance whether the problem had delayed the work or made it more cumbersome. The consequences would often be crucial for deciding how serious the problem was for the organisation.

I prepared checklists for the interviews, where I first asked general questions about problems in the work and then with the IT-systems, and after that asked specifically about for instance problems when logging on to systems or when handling passwords. That was done in order to ensure that all relevant topics were covered with each participant.

During the interviews I often found that most topics were covered when I asked the more general questions. However, the more specific questions made it possible to ensure that any remaining problems were included. I also found that one of the specific questions did not generate any response from the first three participants, and then an extremely interesting one from the following. If in doubt, it seems better to keep than to omit a question.

In addition to exploring the problems the access system should solve, it was also necessary to explore the functions it should support. I asked about a number of these and used micro-moment time-line interviews to explore the problems to be solved by each function, and the situations when it was needed.

I had prepared separate checklists for each of the four groups to be interviewed, to take into account the specific problems of each group. Even then I found, in particular in the nursing homes, that it was essential to be open and see the world from the perspective of the participants. If I had been more focused on getting useful results or on discussing IT-systems, I would have missed some serious and unexpected problems they experienced.

In total I interviewed 22 persons at seven different locations, with each interview lasting about 30 minutes. I made notes during the interviews and typed them up with some additional explanations and expansions of abbreviations shortly after I had completed each day of interviews. That was done in order to make the results accessible when the details of the interviews were no longer present in my memory.

I had discussed with other members of the project, which formats of the results that would be most useful, and we decided I should make two documents. The first document was for use when writing the actual requirements, and it

was highly structured. I sorted the results of the interviews according to the reported problems, so it was possible to see the different situations when each problem had occurred, the different ways it had been solved and what the consequences had been in each case. In total the document gave information on about 90 different problems.

The micro-moment time-line interviews had produced results that all were structured in a similar manner, so it was easier to process the results than results from other unstructured or semi-structured interviews.

The second document was written for decision makers. It was much shorter and started by describing the seven most essential requirements, followed by a brief background of the study, brief details about the requirements and four brief human-centered stories [7] or vivid scenarios [6] that described two current situations of use and two possible usage situations with the access system.

This first part of the work took in total 90 hours in June 2009. After my vacation and the proof-of concept test, I was invited back in late August and September to write the functional requirements for the system. I found that one additional and fairly important function had been identified during the proof-of-concept, when users had complained that it was not available in the system they tested.

I made a note about the additional function, and started to write the requirements based on my first document. A number of reported problems were outside the scope of the present project, so they were omitted from the further work. (They are still preserved in the structured description of the problems, and may for instance be used as a basis for improving work routines.)

Based on the remaining problems I wrote 77 requirements in less than 32 hours, in most cases by transforming one or a few problems into one or a few requirement to be fulfilled by the IT-system. This systematic approach meant that the risk of errors was reduced, and that it was easy to trace each requirement back to specific problems and their consequences. To most of the requirements I added a description of the problem to be solved and its consequences to explain the need of the requirement.

The writing of requirements generated a few additional questions that had to be answered, in particular about the handling of software updates and licenses. It was expected to save money on software updates and licenses, and I realized during the writing, that there were no requirements for tools to handle them.

The participants in the project appeared to be satisfied – maybe even impressed - by the speed and results of my work. One indication was that both documents describing the problems experienced by the users were included in the information used to brief the decision makers.

## **DISCUSSION AND CONCLUSION**

The circumstances of the study make it possible to give a valid evaluation of the general value of using SMM in a

requirement process. The only aspect that can be argued is whether my own skills are so outstanding, that other practitioners cannot produce a similar result. I do not believe that is the case. (But for personal reasons I will not argue strongly against it.)

In earlier software projects where I have used SMM [7, 9] the goal was only to suggest improvements to the software, whereas the goal in the present project was to supply a complete set of functional requirements. It appears that this goal could be fulfilled. At the end it was possible to identify the additional information that was needed to complete the requirements.

However, the identification of existing valuable functions and features was not done in an optimal manner. One such function was discovered almost by accident during the proof-of-concept test. It may be advantageous to use the micro-moment time-line interview to explore, which of the capabilities of the existing system users find particularly valuable, so they are not left out by accident.

It may be advantageous to reverse engineer an existing system to create more complete lists of existing functions and features, and to interview users about situations when each of them are used. In particular by using micro-moment time-line interviews or another type of interviews, that encourage participants to provide background information that makes it possible to evaluate the need of each function and feature. Otherwise, all capabilities of an existing system may be included in a new system, just to be on the safe side.

Similar to my previous studies, I found that the interviews felt natural and encouraged the participants to talk freely; that the interviews were very time-effective because the participants first told about the problem they considered most important; that the users' perspective on the work resulted in new valuable and unexpected information; and that the background information made it possible to evaluate the validity of the results and the importance of each of the needs expressed by the participants.

This study demonstrated the value of continuously discussing the work with other members of the project, for instance which users to be interviewed, topics to be covered and the contents and formats of the results. That was in particular necessary in a project as the present, with several ongoing parallel activities, where it is worthwhile to make continuous adjustments.

It is necessary to take into account that most users do not know or focus on the organisational goals of the project. These have to be clarified through the organisation's

management. If I had involved myself more in these - including discussions of the business case - it is possible that the need for tools for handling software updates and licenses had been identified earlier.

One particular positive experience was, that the results of the interviews rapidly and reliably could be transformed into formats that could be used in the project, and in a manner where it was easy to trace arguments for each requirement back to specific parts of the interviews.

#### LITERATURE

1. Beyer, H. & K. Holzblatt. *Contextual design*. Morgan Kaufmann Publishers, USA 1998
2. Dervin, B.: Interviewing as dialectical practice: Sense-Making Methodology as exemplar. Presented to audience section International Association for Media and Communication Research (IAMCR), IAMCR 2008 Annual Meeting
3. Dervin, B. From, the minds eye of the user, The Sense-Making Qualitative-Quantitative methodology (1992), in *Sense-making Methodology Reader* ed. by Brenda Dervin and Lois foreman-Wernet, Hampton Press, USA, 2003
4. Kristiansen, S. and H. K. Krogstrup. *Deltagende observation, introduktion til en forskningsmetodik*, Hans Reitzels Forlag, Denmark 1999
5. Koen, B. V.: *Discussion of the method*. Oxford University Press 2003
6. Lauesen, S.: *Software requirements, Styles and Techniques*. Addison-Wesley 2002
7. Strom, G.: Stories with Emotions and Conflicts Drive Development of Better Interactions in Industrial Software Projects. Proc. Ozchi 07, Australasian Computer-Human Interaction Conference 2007
8. Strom, G.: Power Distance and User-Centred Design in a Traditional Culture. Proc. DHRS 07 – Seventh Danish HCI Research Symposium ed. by Anker Helms Jørgensen, Morten Borup Harning. IT-U kompendier, 2007
9. Strom, Georg: Sense-Making Methodology: Learn What Users Understand is Important. Proc. Sixth Danish HCI Research Symposium, Aarhus, Denmark November 15, 2006, ed. by O.W. Bertelsen, M. Brynskov, P. Dalsgaard, O. S. Iversen, M. G. Petersen, M. Wetterstrand, University of Aarhus 2006
10. Weick, K.E. *Sensemaking in Organizations*, Thousand Oaks: Sage, USA 1995