Chapter 3

Requirements and Design

Ian Sommerville, University of St Andrews

Summary

Much of our work in social analysis of complex systems has been concerned with how we can use such analyses in systems specification and design. In this chapter, I discuss how fieldwork can be used to gather date that informs the requirements engineering process. I discuss the ways in which fieldwork can be used in conjunction with system prototyping and how analyses can be used for the 'sanity checking' of requirements. I conclude by discussing the limitations of fieldwork in informing complex system requirements but suggest, nevertheless, that short observational studies should be an inherent part of requirements engineering processes.

The chapter on Coherence discusses a method to support the use of fieldwork in requirements engineering and the chapter on Patterns captures common features of work settings that can be used to help understand system requirements.

Background

Requirements engineering (RE) was one of the starting points for our work in socio-technical systems and interactions with our colleagues from Sociology. We were interested in the general problems of understanding the requirements for complex IT systems and in developing new viewpoint-oriented approaches to RE. But it became clear to us that many of the practical problems that people encountered with systems were a consequence of the ways in which they worked and that if we were to understand their 'real' requirements, then we really needed a better understanding of work, the actual rather than the formal business processes and the relationships between these processes and organisational factors that influenced the ways in which work was done. Consequently, we started to explore how ethnographic approaches could be used to understand work and to investigate how fieldwork could be used to inform the requirements engineering process. If we understood work as it really was done, then we were convinced that we would produce higher quality software system requirements.

Of course, we know of some of the views of the agile community who suggest that requirements should emerge incrementally during development. While this is perhaps true for some kind of requirements, the reality of current systems engineering is that some kind of requirements document is always needed for large systems before development begins (and sometimes before the development contract is issued).

Understanding requirements

There are two problems in requirements engineering where ethnography may help:

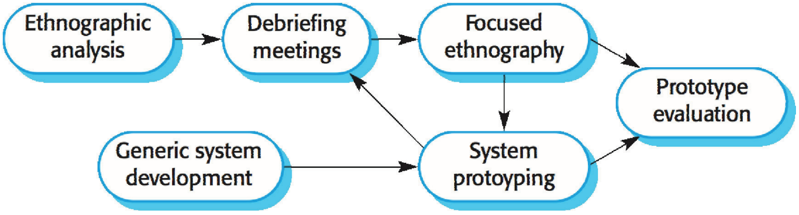
1. **Businesses do not understand their own processes**. Requirements for software are often based on an understanding of a 'formal' process which bears little relationship to the real business processes that are used. Ethnography can reveal the differences between formal and actual processes and so allow more appropriate requirements to be proposed.

2. **People cannot articulate their requirements**. Ethnography can provide information about what people actually do and so can serve as a source of requirements. This is particularly useful when combined with a prototyping approach as discussed below.

However, fieldwork simply develops a rich picture of how people work and the business processes involved. It does not, in itself, generate system requirements so we need to have one or more mechanisms to help translate the understanding of work into practical requirements for system support.

Fieldwork and prototyping

Prototyping is an established technique for supporting the requirements engineering process. A system prototype is developed and is used as a basis for experiment. People find it much easier to articulate what they need when they have a real system in front of them. However, a major problem with prototyping is getting user involvement and so we explored how fieldwork can be used to inform prototype development. The following diagram illustrates the process that we used.



We identified 4 key questions that we should ask fieldworkers who have been engaged in studies of work:

1. What **characteristics** of the existing system are **unimportant** and need not be supported in an automated system?
2. What are important **current activities** which need not be supported in an automated system because the activities are a consequence of the fact that no or limited automated support is available?
3. What **characteristics** of the existing system must be **replicated** without change in a new system?
4. What **activities** from the existing system may be **supported** in a way which is different from that used in the current system?

We developed these questions during our initial studies of air traffic control but we think that they are still the key issues in translating an understanding of work to system requirements.

Sanity checking

The reality of systems development is that requirements are not necessarily going to be informed by fieldwork so they may be based on an inadequate understanding of way in which people really do their work. An approach to requirements engineering based on social analysis can be helpful here as it can highlight pitfalls and things that should not be in the requirements.

This can be a cost effective way to use fieldwork in the requirements engineering process as the requirements focus the fieldwork – rather than building a general picture of the work being done, the fieldworker can focus on the activities that are reflected in the requirements and can identify requirements which could cause problems in practice.

A situation where we used this approach was in a financial institution that wanted to introduce a new counter system. We discovered that the requirements were such that they required the teller to enter all of the customer information without interruption – something that is completely impractical in a busy high-street branch. After our studies, the requirements were changed to allow information to be added incrementally.

Viewpoints

We have always adopted the position that, while the ideal fieldworker is a trained ethnographer, we will only be able to introduce fieldwork into requirements engineering processes when we provide support for non-specialists to do this work. The Coherence method and the work on Patterns of Interaction all reflect this view.

Our starting point for developing these approaches was earlier work done on ‘quick and dirty’ ethnography where we departed from the conventional notion that ethnography should be a prolonged process and developed an approach to ethnography that was intended to generate useful information quickly. This could be requirements sanity checking as discussed above or could be a short period of fieldwork that focused on understanding the system in general and the types of problems and issues that arose. We discovered that even a short interaction was very valuable in illustrating the complexity of some processes.

To help with this short period of fieldwork, we came up with the notion of social viewpoints as a way of organising the fieldwork and its documentation. These social viewpoints are, essentially, perspectives on a system and we recommend three such viewpoints:

1. The **work setting**, which describes the environment where the work takes place and the interactions with this environment. This is often reflected in the way that work is physically organised to allow, e.g. to facilitate communications between the people involved in the processes. It may also take into account the use of shared electronic resources such as shared folders on a server.

2. The **work flow**, which describes the sequences of work activities, infor-

mation flows etc. The important thing here is to look at how the flow of work is used to coordinate the work of the people involved and to look for how people handle exceptions that arise.

3. **Social and organisational perspectives**, which show how the work of individuals in the process relates to other people's work and to broader organisational issues. Therefore, you might look at the effects of the need to comply with regulations on how the work is done, how people become aware of what other people are doing, etc.

Documentation

Fieldwork is usually recorded as a set of notes in narrative that is perhaps supplemented by photographs of the work setting, documents about the work, video recordings, etc. This body of work is quite personal to the fieldworker so it needs them to be available to interpret it. This is obviously problematic as they cannot always be available for consultation so we looked at alternative means of documenting the work.

We developed a tool called the Designer's Notepad which was, essentially, a simple tool that allowed a user to cut and paste information from the fieldwork record into multimedia notes and to link these notes together. These could be set up by the fieldworker and then consulted by others involved in the RE process.

This was a one-off tool and is no longer supported but you can replicate much of its functionality by using mind mapping or brainstorming software.

Retrospective

An accepted problem with fieldwork is that it tells you about work as it is done and doesn't really give you any clues about ways of doing things differently. Of course, a more informed picture of the work being done means that, hopefully, you will come up with better system requirements. Consequently, we now think that fieldwork is not an activity that should precede the development of system requirements but should not be started until an outline set of requirements is available. It can then be used for sanity checking as discussed above but also for adding details to high-level requirements.

Much of our early work on ethnography and requirements involved work in control rooms or other settings where everyone worked together in the same place. The reality of modern work is that it is often distributed both in time and place so situated fieldwork is much more difficult. These difficulties are exacerbated by the fact that distributed work is facilitated primarily by electronic rather than paper documents and the ways in which electronic documents are used is harder to observe.

The other problem with fieldwork, which is shared with user centred approaches to requirements such as those used in agile methods, is that it is mostly concerned with work as done by users. It is therefore less useful for understanding broader organisational requirements or what are sometimes called 'non-functional' requirements – dependability, security, compliance, etc.

Nevertheless, in spite of these problems, we are convinced that a short period of fieldwork can be immensely valuable in the requirements engineering process. By observing how the work to be supported by a software system is actually done, you can identify key activities that must be supported and can discover problems with proposed requirements before these are implemented.