Chapter 18

Scientific End-User Developers and Barriers to User/Customer Engagement

Judith Segal
The Open University, UK

Chris Morris
STFC Daresbury Laboratory, UK

ABSTRACT

When software supports the complex and poorly understood application domain of cutting-edge science, effective engagement between its users/customers and developers is crucial. Drawing on recent literature, the authors examine barriers to such engagement. Significant among these barriers is the effects of the experience that many research scientists have of local scientific end-user development. Through a case study, the authors demonstrate that involving such scientists in a team developing software for a widely distributed group of scientists can have a positive impact on establishing requirements and promoting adoption of the software. However, barriers to effective engagement exist, which scientific end-user developers can do little to address. Such barriers stem from the essential nature of scientific practice.

INTRODUCTION

The aim of this paper is to shed some light onto the problems of user/customer engagement in scientific software development. We define scientific software to be software specifically designed either to advance science directly, for example, by providing models and simulations to investigate problems where the science is too fast or too slow or too large or too small or too complex or too dangerous to investigate in vivo (Wilson, 2006), or to support the practice of science, for example, by providing means by which a community of scientists might share remote instruments or data. There is (at least) one common factor underlying

DOI: 10.4018/978-1-4666-2059-9.ch018

every scientific software development and that is the necessity of effectively involving the users, the scientists who are going to be using the software in their workaday lives, in the development in some way. This statement is probably true of all software developments to a greater or lesser extent, but is especially relevant to scientific software where the application domain is both complex and only partially understood, even by experts in the domain. It is equally important to involve the customers effectively, that is, those scientists who control the purse-strings of the development and have overall high-level responsibility for the direction in which the science is going and hence for how the software might best support this direction. As we shall discuss later, the users and the customers may or may not be the same people.

In this paper, we suggest that the deployment of scientific end-user developers on the development team might go some way towards addressing the problem of how to optimize user/customer engagement with developers. We base our suggestion on evidence from a case study carried out by the first author over a period of about three years, tracking the development of a Laboratory Information Management System (LIMS). This LIMS is intended to support a community of biologists, that is, biologists who are not necessarily co-located but who are working on different problems within the same sub-discipline. The LIMS development team consisted of software engineers with no prior knowledge of biology, software engineers with some biological background and scientific end-user developers. The term ‘scientific end-user developer’ used in the context of this development team is somewhat ambiguous. The people referred to *are* scientific end-user developers in the sense that they think of themselves primarily as scientists, have little or no formal education in software development, and have considerable experience of developing software for their own use and for the use of other scientists working closely with them. They are *not*, however, representative of the potential user group of the LIMS, though they do work alongside representatives of that group.

We begin this paper by discussing barriers to user engagement in general, and then in the particular context of software development for a distributed community of research scientists. We focus specifically on those barriers arising from the experience of many research scientists of developing their own software in a very local context, that is, in order to address a particular scientific problem at a particular point in time at a particular laboratory.

We then describe our case study and its findings. The overall finding is that including scientists with scientific end-user development experience on the development team of a significant piece of community software can have a very positive effect on the development. We are aware that some readers may baulk at the idea of taking seriously findings based on a single case study. But sometimes a case study grounded in real-world practice with all its richness of context is all that is needed to convince a reader that ‘yes, this must be true’. And sometimes the findings of a case study are strengthened by their being consistent with findings in other disciplines, as is the case here with an aspect of diffusion theory (Rogers, 2003).

We continue with a discussion on the scope of our findings, both as regards the extent to which scientific end-user developers can improve user/developer relations and the extent to which our results can be generalized. As is traditional, we conclude with a summary and conclusions.

BARRIERS TO USER ENGAGEMENT IN COMMUNITY SOFTWARE DEVELOPMENT PROJECTS

We begin by discussing our use of two terms: ‘community software’ and ‘user’. By the term ‘community software’, we mean software which, like the LIMS of our case study, is intended to support the activities of a distributed group of