Chapter 12
Collaboration in Open Source Domains
A Perspective on Usability

Görkem Çetin
Gebze Institute of Technology, Turkey

Mehmet Göktürk
Gebze Institute of Technology, Turkey

ABSTRACT
Free and open source software (F/OSS) developers have a tendency to build feature-centric projects rather than following a user-centered design, ignoring the necessity of usability in the resulting product. While there are many reasons behind this, the main cause can be stated as the lack of awareness of usability from developers’ point of view and little interaction of project stakeholders with Human-Computer Interaction (HCI) studies. This chapter examines different types of collaboration methods of usability experts and developers focusing particularly on open source projects, together with potential issues envisaged during the communication phases. The chapter also focuses on the collaboration trends and patterns of HCI experts, developers and users with an emphasis on concerns related to inefficient exploitation of current tools and technologies and provides an open usability engineering method which could be exploited in distributed projects.

INTRODUCTION
Free/Open Source Software (F/OSS) is a generic term that is used for software developed under a license, which allows the use, modification and distribution of the software without claiming a fee for it. There are other terms, like “open source”, which describe a slightly different version of the F/OSS term; however the basic idea is the same in terms of usage, distribution and modification. A surprising fact with F/OSS development is that while this model does not pose an in-line methodology with traditional software development models, there’s a growing interest in F/OSS products showing an effective, rapid and reliable software development process. As of this writing, there are more than 1.5 million registered users in Sourceforge with more than 150,000 projects and Apache is the clear market leader with 53% market share according to Netcraft statistics.
Over the last 20 years, we have seen an abundance of literature on how usability can be used to leverage the return of investment, increase end user satisfaction and lower the training budget. Setting interface standards, providing usability requirements before a software project begins, heuristic evaluations and usability tests have been conducted in order to evaluate and improve an application or a series of web pages in order to increase the performance of the end user.

On the other hand, little attention has been paid to the usability of end user applications in F/OSS projects, although it is widely acknowledged as a success factor of desktop applications developed on the Internet. Lack of common usability design guidelines and methods of communication between usability experts and F/OSS developers resulted in F/OSS software with relatively low level of usability. While there are counterexamples of this, it’s worth investigating the main factors of lack of F/OSS application usability and concentrate on how to improve the user centered design experience of voluntary projects, since usability can be integrated into current open source development processes.

As such, F/OSS usability is an important phenomenon deserving a study in itself. User centric design has not been the first priority of open source projects developed by people geographically distributed in all parts of the world. Thus, it can be argued that usability awareness and representation has been neglected for a long time. Traditionally, F/OSS has been successful in server products (Apache, MySQL, etc), libraries and compilers (GNU C compiler, Glibc, Qt, GTK) and console-based applications (bash shell, pine, etc). These applications were used by experienced people in specific organizations like universities and research institutes and those who have used them demanded less training and usability criteria. Most of the users of these applications are relatively technically sophisticated and the average desktop user is using standard commercial proprietary software (Lerner and Tirole, 2002).

Lack of clear usability requirements, awareness of user-centered design and a social collaborative tool to discuss usability issues usually result in a poor evaluation for F/OSS products. This is problem in ergonomic design and interface psychology, and hackers have historically been poor at it (Raymond, 1999). As F/OSS matures and enterprises start using F/OSS, it entered the mainstream area and enterprises started to demand not only reliability, security and efficiency, but usability, ergonomics and ease of use. As a result, in order to answer the demand from customers, some open source projects have tried to adopt techniques from previous proprietary work, such as explicit user interface guidelines for application developers (Benson, Adam, Nickell & Robertson, 2002).

On the other hand, there’s a scarcity of published usability studies and test reports of F/OSS from academics. We are aware of studies on GNOME (Smith, et al., 2001), Greenstone (Nichols, Thomson, Kirsten & Yeates, 2001), file browser screening test (Reitmayr, 2007), Linux desktop out of box experience (Göktürk & Çetin, 2007) and OpenOffice.org usability test (Çetin, Verzulli & Frings, 2006), to name a few. While there is little formal research on F/OSS usability, developers are well aware that F/OSS usability is a problem that should be solved and usability is a significant issue of unsuccessful F/OSS projects. According to Müehling (2004), “it becomes clear that F/OSS projects have a fundamental problem: they lack usability resources that help achieve better usable software for non-geek users”.

Issues stemming from development paths affect the usability process of open source projects. Lack of developers also indirectly means lack of usability experts. Hierarchical models in which main decisions are taken by lead (core) developers with a lack of user interface design background affect the usability centeredness of F/OSS developers, hence the usability of the product. A developer working in a F/OSS project should have explicit rights and title, together with a strong technical