Introducing Usability

Natalia Juristo, Universidad Politécnica de Madrid
Helmut Windl, Siemens AG
Larry Constantine, Constantine & Lockwood

Certainly many of you have had enough frustrating experiences using software to acknowledge that usability strategies, models, and methods are often not applied adequately during software construction. Usability is not a luxury but a basic ingredient in software systems: People’s productivity and comfort relate directly to the usability of the software they use.

One definition of usability is quality in use.¹ In simple terms, it reflects how easy the software is to learn and use, how productively users will be able to work, and how much support users will need. A system’s usability does not only deal with the user interface; it also relates closely to the software’s overall structure and to the concept on which the system is based.

Usability is a difficult attribute to embed in any system—not only software—and it requires specific knowledge and a lot of awareness about the user’s likings, requirements, and limitations. However, many software developers would rather work with machines than with people; they show little interest in issues such as how much data should appear on the screen at one time. Additionally, many designers do not realize that their perception of their creation does not provide much information about how others will react to it. That is why we get all those “perfectly obvious to the designer” creations.

We should regard usability as one more quality attribute for consideration during software construction. Of course, we shouldn’t concentrate on just a single qual-
In This Issue

Our aim in this special issue is to encourage software developers to listen more carefully to usability engineers and to give usability a more meaningful place in the overall software process.

Usability is not an abstract idea; applications already exist that demonstrate it well. This issue tries to promote the application of that knowledge to a wider range of companies and systems by emphasizing existing techniques rather than research on new methods or models. Most of the articles following deal with real experiences.

The seven articles we have selected fall into four categories. First, because of the low level of usability awareness and skills among software engineers, we include a tutorial, “Usability Basics for Software Developers” by Xavier Ferre, Natalia Juristo, Helmut Windl, and Larry Constantine. This will help some readers better understand the rest of the articles.

Second, the links between usability and business competition or market domains is an especially interesting topic. In “Usability and the Bottom Line,” George M. Donahue discusses usability cost-effectiveness and describes how to perform a cost–benefit analysis. A company will often make this the first step toward integrating usability into its processes.

Two articles discuss industrial experiences with usability and software development. In “Usability in Practice: Three Case Studies,” Karla Radle and Sarah Young present three real cases of introducing usability engineering into an organization. Their article points out common obstacles and describes some lessons learned. In “Integrating Usability Techniques into the Software Development Process,” Kathi Garrity, Francie Fleek, Jean Anderson, and Fred Drake describe how two groups in their company, the Software Engineering Group and the User Performance Group, came to understand each other’s processes, vocabulary, and approaches. The article discusses the challenges they faced and the development process that resulted.

Because of the Web’s rapidly increasing significance in software development, the fourth group of articles addresses the role of usability engineering for Web applications. The usability of Web sites and applications continues to be worse than that of more traditional software. However, to be competitive in e-business, usability is a must. In “A Global Perspective on Web Site Usability,” Shirley Becker and Florence Mottay discuss how to assess Web application usability.

Beyond usability, we have other issues to consider. As an example, the one research article in this issue, “Engineering Joy” by Marc Hassenzahl, Andreas Beu, and Michael Burmester, talks about the joy of use—the task-unrelated aspects of quality called hedonic quality. After a brief overview of research on the relationship between enjoyment and software, this article shows that traditional usability engineering methods are not suited to analyzing and evaluating hedonic quality. These authors present promising new approaches.

References


About the Authors