Chapter 2.5
Design Methods for Experience Design

Marie Jefsioutine
University of Central England–Birmingham, UK

John Knight
University of Central England–Birmingham, UK

ABSTRACT

The following chapter describes an approach to Web design and evaluation where the user experience is central. It outlines the historical context in which experience design has evolved and describes the authors' experience design framework (EDF). This is based on the principles of user-centred design (UCD) and draws on a variety of research methods and tools to facilitate the design, development, and evaluation of user experiences. It proposes that to design usable, accessible, engaging, and beneficial Web sites, effort needs to focus on visceral, behavioural, reflective, and social factors, while considering contexts such as the who and why; what and how; when and where; and with what of Web site use. Research methods from a variety of disciplines are used to support exploration, communication, empathy, and speculation. Examples of the application of the EDF, to various stages of the Web design process, are described.

INTRODUCTION

Although change is afoot, designers (including design engineers)—still worry that corporate bosses and clients see them as merely 'making things pretty.'

— Molotch, 2003, p. 28

Producing Web sites is a process that involves a range of skills and disciplines. Design is not an add-on to make the screens look good. Design impacts on what people do, how they do it, and with whom they do it. Products, whether they are Web sites, toasters, or services embody a range of values and our interaction with them is a form of communication. When interacting with a Web site, for example, we may be communicating with other people, real or imagined. As we type with the keyboard, we hear and feel the keys and may vent our frustration through the force of our tapping. While reading the words on a screen we are also taking in all kinds of nonverbal messages

from the layout—images, colours, fonts, icons, language, and style of language. Our experience is also affected by our surroundings, our memories of past actions, our current knowledge, and our expectations of the future.

Interaction with a Web site is mediated by a network of people, machines, and systems. The Internet works because of the design of protocols, browsers, laws, technical standards, security mechanisms, machines, communication technologies, and physical infrastructure as much as it does because of the design of the Web pages. Our experience of a Web site is therefore a product of our personal understanding of its context. Givechi and Velázquez (2004) describe the "positive space" of a product, which becomes more significant than the product itself, "the aura of a product, the sum of its physical attributes plus its intangible essence—or the meaning it hosts for each of its users" (p. 43). The meanings we attach to a product change with time, in part due to our changing experiences, the wider socioeconomic and political context of a product, and our changing expectations. To build a successful product, a designer needs to be aware of as many of these factors as possible, and this is why a consideration of the whole user experience is important.

FROM SOFTWARE ENGINEERING TO EXPERIENCE DESIGN

Concepts like "user experience" and "design of experience" are common in the design and business communities now (Fulton-Suri, 2004, p.14), but this has not always been the case, and in some software development circles *designer* still has negative connotations. It is important to recognise the context in which Web design has emerged, to understand why these concerns are so topical.

The development of computer software and interfaces in the 1960s emerged from fields traditionally associated with engineering and science. This era was typified by optimism for technology, and even in the traditional design disciplines there was a movement towards rationalising design methods. Hailed as the "design science decade" (Fuller, 1969, p. 305) the emphasis was on objectivity, rationalism, and technology.

The term *software engineering* can be traced back to an international conference convened by NATO in 1968, to solve the "software crisis" (Campbell-Kelly & Aspray, 1996, p. 200). The crisis emerged from large scale (often military) software projects that encountered problems in management and quality. In response to these failings a number of risk management strategies were advocated including the *waterfall development lifecycle*. Here design and development flowed through a predetermined course of phases. The formalised nature of the process was predicated by each phase having specified inputs and outputs, which could be checked along the way.

A design approach based purely on linear, logical problem solving did not work for the new technologies (e.g., computing, software, and solid state electronics), where problems were nebulous and constantly evolving through technological development and changing requirements. Brooks' (1975) experience on IBM's System/360 project was typical. He describes how software defied traditional logical engineering approaches. Even increasing resources did not improve the success of the approach.

In design disciplines such as product design and architecture, there was a growing realisation that some problems were unsolvable by logical deduction (Cross, 2001). Rittel and Webber (1973), for example, contrasted the "wicked problems" of design with the "tame" ones of science and engineering. In software design attention was turning to human factors. Whereas traditional engineering and industrial design focussed largely on external and measurable ergonomic factors, computer interaction required an understanding of internal, cognitive aspects. In 1969 the *International Journal of Man Machine Studies* was launched