The importance of the field we know as human-computer interaction grows as information technology increasingly becomes a pervasive part of everyday life. This chapter argues that a design perspective is better suited to meet the challenge than the traditional foundations of experimental psychology and information systems development. For professional software development, main implications include addressing the whole design task rather than merely the user interface, breaking monolithic processes into smaller steps, and recognizing that problem setting can never be separated from problem solving. The main implication for higher education is the notion of a design learning studio where constructive and reflective work are intertwined.

INTRODUCTION

The role of human-computer interaction in software development is unclear. This holds for professional practice as well as for the development of knowledge concerning appropriate and interesting applications of information technology to human and social contexts. It may be argued that the main emphasis of HCI has always been placed on identifying and understanding usability problems in task-oriented computer systems. This position is perhaps not the most appropriate one when the responsibility concerns use experiences in a constructive setting, where new artifacts and new use situations are developed. It is certainly not the only conceivable one.

In this chapter, I argue that concerns with use qualities of interactive systems are better addressed by viewing the work from a design perspective. The emerging discipline is called interaction design and defined as the shaping of interactive systems with particular emphasis on their use qualities. I introduce a number of foundational concepts from contemporary design studies and discuss the impact of an interaction design view on professional software development practice and higher education.

THE LANDSCAPE IS CHANGING

It is probably fair to say that the field of human-computer interaction still has its main strengths in improving user interfaces of task-oriented applications in work contexts. There are historical reasons for this bias, of course. When the field emerged (originally known as software psychology) in the 70s, computers were more or less only used for task-oriented
applications in work contexts. Contemporary HCI carries this heritage in terms of, e.g., problem framing in research and education contexts, development methodologies with the emphasis on usability engineering and evaluation, academic prerequisites for HCI practice.

**Example:** The SIGCHI HCI curriculum. A slightly dated but still influential example is the recommended HCI curriculum developed by the ACM Special Interest Group on Computer-Human Interaction (ACM, 1992). It defines the contents of HCI quite generally as the nature of HCI, the use and context of computers, human characteristics, computer system and interface architecture, and development processes. However, the elaboration of the headings emphasizes descriptive models of human behavior and information processing and the optimization of human-machine fit for goal-oriented tasks. (End of example.)

History has shown rather conclusively that the integration of HCI in professional software development is not straightforward. One of the main issues is the HCI emphasis on evaluation of ideas, prototypes and artifacts. Empirical studies clearly illustrate the danger of antagonizing design and evaluation. (See, e.g., Näslund and Löwgren, 1999.) The surge of interest in usability labs in the early 90s, followed by a massive move to distributing HCI skills in development teams across organizations, is another example.

Other issues concerning HCI in professional software development include the “analyze without bias, then design” view of development processes that works poorly in practice; more on this below. It also favors incremental change (design based on fixing problems in the current state) over innovative leaps. One may even argue that the focus on user interfaces and measurable qualities—which is fully understandable given the academic heritage of the discipline and the industrial emphasis on requirement specifications—is actually harmful to the quality of the resulting product. These issues are discussed in some depth in (Löwgren, 1997).

The most significant change in the landscape of software development, however, is brought about by the rapid uptake of the World Wide Web. Suddenly, here is a whole new field of software design that calls for skillful crafting of use qualities. The majority reaction of the HCI community seems to be slight repurposing of existing methods and skill sets, in combination with advocacy for task-oriented perspectives on the “use” of the web. Some examples include Nielsen’s (1995–2000) influential recommendations and the empirical studies by Spool and colleagues (1999).

**Example:** An alternative perspective. It is a useful exercise to compare the mainstream HCI approaches with other alternatives, such as the design-seductive-qualities perspective put forth by Khaslavsky and Shedroff (1999). Seductive means to have alluring or tempting qualities. A promise of extraordinary experiences is involved, as well as the qualities required to make good on the promise. Tetris is clearly a seductive game. People keep playing and developing a relationship that is something other than the simple fulfillment of task-related goals. And it is quite obvious that stunning graphics or beautiful visual design are not involved (at least in the original version from 1985 for the 9-inch Mac). Dynamic queries (Ahlberg et al., 1992) is another example of a seductive, albeit less well-known approach. The concept of turning boring old database queries inside out and shaping the data with your fingertips is immediately attractive and enticing to many people.

Khaslavsky and Shedroff describe seduction as a process of enticement (grabbing attention and making an emotional promise), relationship (making progress with small fulfillments and more promises. This step can continue for a long time) and fulfillment (fulfilling the final promises and ending the experience in a memorable and positive way). The checklist for identifying seductive qualities goes as follows. Ask yourself whether the product under scrutiny: Entices you by diverting your attention? Surprises you with...
Pathways to Participatory Landscape Governance in Northern Laos: The Role of Information and Communication Technologies
www.igi-global.com/article/pathways-participatory-landscape-governance-northern/46095?camid=4v1a