Chapter 11
An Instructional Design
“Use Case”: Instructional Technologies for Developer Stakeholders

Shalin Hai-Jew
Kansas State University, USA

ABSTRACT

The work of an instructional designer is highly dependent on an instructional technology substrate, at every phase of the instructional design work: research, planning, communication and coordination, prototyping, design, content searches, content development, branding, alpha and beta testing, revisions, the delivery of course contents, and the archival in learning object repositories. Technologies in this substrate are built to a variety of standards. There are standards for interoperability, for machines to communicate with each other, for information to be held securely (information assurance), for from-life information to be captured and recorded (whether light, detail, sound, or motion), for communications to be exchanged among people, and for digital artifacts to be labeled and protected and delivered to users. The instructional designer “use case” then refers to the on-ground realities of instructional design work and the critical reliance on instructional technologies, and what this in vivo perspective shows about the need for (in part) user-based insights for instructional technology research, design, and development. Every technology has multiple use cases, or theoretical situations in which users use that technology. An instructional designer use case shows the many uses of technologies by an instructional designer, to shed light on how the software technologies may be better tailored to the needs of instructional designers and other digital content developer stakeholders.

DOI: 10.4018/978-1-61350-198-6.ch011
INTRODUCTION

Instructional technologies underlie every aspect of instructional design work. In the design work, multiple software programs are in use at any one time and have to work across programs. Delivery systems for the electronic learning (e-learning) vary from Web-based systems to tangible digital storage devices. Each project has different technological requirements, including the uses of proprietary software systems. There may be course management systems necessary for dedicated designs of specialized contents for particular domain fields.

In the academic environment and culture, instructional design is a service to faculty and subject matter experts (SMEs) to focus on the design and building of online learning—whether these are certificate programs and sequences, semester courses, short courses, trainings, demonstrations, stand-alone simulations, teaching case studies, or other elements.

This chapter will argue the importance of close relationships between user experiences (via use cases or theoretical scenarios of how a technology might be used in a variety of circumstances) and those designing the technologies that are critical tools for all developers. The rationale for using a live setting is the multidimensionality of development initiatives in a higher education setting. The academic context and culture involve particular requirements that need to be met. The learning, too, is designed for a wide span of learners. This then refers to the reliability and validity of product testing in the field and emphasizes the importance of user connections with the software makers and vice versa, which often occurs at professional conferences, invited-webinar presentations by various software makers (who connect with content creators for value-added knowledge sharing), one-on-one conversations, and other points-of-contact. This chapter also looks at ways that user feedback may be meaningfully for rigor and applicability to the software design in terms of clear “specs” (specifications).

LITERATURE REVIEW

Research and development work in information technology (IT) systems applications and software development has historically relied on quantitative measures. These positivist approaches involve elements such as the following: “theorem proving, mathematical modeling and simulation, controlled experiments, field experiments, quasi experiments, and testing (De Villiers, 2005, p. 112). It has only been fairly recently that more qualitative and interpretivist approaches have been used: observations, surveys and questionnaires, interviews, grounded theory (which relies on deep immersions), heuristics, applied action research, case studies, focus groups, ethnographies, document and artifact studies, and user feedback. Some types of information may only be understood through user feedback.

This includes information about the domain and context specific technical issues, and about multifaceted cultural, political, communicational, motivational, and personal issues. As there is no information systems development (ISD) method that would yield such information comprehensively, it could be achieved by user-oriented approaches, for instance by participatory design (PD) (Pekkola, Kaarilahti, & Pohjola, 2006, p. 21).

And yet, many such insights never make it into the final product:

User-centered innovations, yielding concepts that would be of great value to users, never make it to product, or they get watered down, re-engineered, and washed out on the way to becoming unusable products (Henderson, 2005, p. 25).

A use case in information technology parlance refers to a scenario in which a particular technology product is used a particular way to a particular end by a particular sort of user. A number of such cases are developed by testers and those on the