

Chapter 7.4

Evaluating Computer–Supported Learning Initiatives

John B. Nash

Stanford University, USA

Christoph Richter

University of Hannover, Germany

Heidrun Allert

University of Hannover, Germany

INTRODUCTION

The call for the integration of program evaluation into the development of computer-supported learning environments is ever increasing. Pushed not only by demands from policy makers and grant givers for more accountability within lean times, this trend is due also to the fact that outcomes of computer-supported learning environment projects often fall short of the expectations held by the project teams. The discrepancy between the targets set by the project staff and the outcomes achieved suggests there is a need for formative evaluation approaches (versus summative approaches) that facilitate the elicitation of information that can be used to improve a program while it is in its

development stage (c.p., Worthen, Sanders & Fitzpatrick, 1997). While the call for formative evaluation as an integral part of projects that aim to develop complex socio-technical systems is widely accepted, we note a lack of theoretical frameworks that reflect the particularities of these kind of systems and the ways they evolve (c.p., Keil-Slawik, 1999). This is of crucial importance, as formative evaluation will only be an accepted and effective part of a project if it provides information useful for the project staff. Below we outline the obstacles evaluation faces with regard to projects that design computer-supported learning environments, and discuss two promising approaches that can be used in complimentary fashion.

BACKGROUND

According to Worthen et al. (1997), evaluation is “the identification, clarification, and application of defensible criteria to determine an evaluation object’s value (worth or merit), quality, utility, effectiveness, or significance in relation to those criteria.” In this regard evaluation can serve different purposes. Patton (1997) distinguishes between judgment-, knowledge- and improvement-oriented evaluations. We focus on improvement-oriented evaluation approaches. We stress that evaluation can facilitate decision making and reveal information that can be used to improve not only the project itself, but also outcomes within the project’s target population. The conceptualization of evaluation as an improvement-oriented and formative activity reveals its proximity to design activities. In fact this kind of evaluative activity is an integral part of any design process, whether it is explicitly mentioned or not. Accordingly it is not the question if one should evaluate, but which evaluation methods generate the most useful information in order to improve the program. This question can only be answered by facing the characteristics and obstacles of designing computer-supported learning environments.

Keil-Slawik (1999) points out that one of the main challenges in evaluating computer-supported learning environments is that some goals and opportunities can spontaneously arise in the course of the development process and are thus not specified in advance. We believe that this is due to the fact that design, in this context, addresses ill-structured and situated problems. The design and implementation of computer-supported learning environments, which can be viewed as a response to a perceived problem, also generates new problems as it is designed. Furthermore every computer-supported learning experience takes place in a unique social context that contributes to the success of an intervention or prevents it. Therefore evaluation requires that designers pay attention to evolutionary and cyclic processes and

situational factors. As Weiss notes, “Much evaluation is done by investigating outcomes without much attention to the paths by which they were produced” (1998, p. 55).

For developers designing projects at the intersection of information and communication technology (ICT) and the learning sciences, evaluation is difficult. Evaluation efforts are often subverted by a myriad of confounding variables, leading to a “garbage in, garbage out” effect; the evaluation cannot be better than the parameters that were built in the project from the start (Nash, Plugge & Eurlings, 2001). Leaving key parameters of evaluative thinking out of computer-supported learning projects is exacerbated by the fact that many investigators lack the tools and expertise necessary to cope with the complexity they face in addressing the field of learning.

We strongly advocate leveraging the innate ability of members of the computer science and engineering communities to engage in “design thinking” and turn this ability into a set of practices that naturally becomes program evaluation, thereby making an assessment of the usefulness of ICT tools for learning a natural occurrence (and a manifest activity) in any computer-supported learning project.

Design-Oriented Evaluation for Computer-Supported Learning Environments

There are two approaches that inherently relate themselves to design as well as to evaluation. Therefore they are useful tools for designers of computer-supported learning initiatives. These two perspectives, discussed below, are scenario-based design and program theory evaluation. Both approaches assume that the ultimate goal of a project should be at the center of the design and evaluation discussion, ensuring a project is not about only developing a usable tool or system, but is about developing a useful tool or system that improves outcomes for the user. Beyond this

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/evaluating-computer-supported-learning-initiatives/27612

Related Content

Information Retrieval in Virtual Universities

Juha Puustjarvi and Päivi Poyry (2006). *International Journal of Distance Education Technologies* (pp. 36-47). www.irma-international.org/article/information-retrieval-virtual-universities/1682

A FAQ-Based e-Learning Environment to Support Japanese Language Learning

Yuqin Liu, Chengjiu Yin, Hiroaki Ogata, Guojun Qiao and Yoneo Yano (2011). *International Journal of Distance Education Technologies* (pp. 45-55). www.irma-international.org/article/faq-based-learning-environment-support/55798

Quality Function Deployment in Training Design

Arthur B. Jeffery and Mary F. Bratton-Jeffery (2009). *Encyclopedia of Distance Learning, Second Edition* (pp. 1714-1722). www.irma-international.org/chapter/quality-function-deployment-training-design/11978

I Scratch and Sense But Can I Program?: An Investigation of Learning with a Block Based Programming Language

N. K. Simpkins (2014). *International Journal of Information and Communication Technology Education* (pp. 87-116). www.irma-international.org/article/i-scratch-and-sense-but-can-i-program/117279

Instructor Presence in Online Distance Classes

Janet Lear (2009). *Encyclopedia of Distance Learning, Second Edition* (pp. 1216-1219). www.irma-international.org/chapter/instructor-presence-online-distance-classes/11901