Is end user productivity an oxymoron?

Barbara A. Schuldt
University of Wisconsin, Oshkosh

Recommended Citation

Is end user productivity
an oxymoron?

Barbara A. Schuldt
University of Wisconsin, Oshkosh

The primary goal of end user computing is to increase the productivity of the end user. By using information technology the end users have changed the scope and depth of their job tasks, making productivity changes hard to measure. The end user decision maker can have data at his/her finger tips through a few key strokes. This data can then be processed to provide additional information to improve the decision. Decision improvement is rarely quantifiable or even measurable as an efficient and effective use of the end user's time or the computer. This makes it difficult to measure end user productivity with these types of job tasks.

Traditionally, end user computing productivity has been measured by assessing the computer utilization, satisfaction, and quantifiable output. None of these measures has effectively evaluated the changes in the end users' productivity and job performance by using the computer to improve a decision. Software developers have focused on the utilization of the computer and satisfaction with the software as keys to strengthen sales.

Thousands of dollars and uncountable person hours have been expended in the development of software tools to enhance these aspects of end user computing. It has been generally assumed that increasing the end user's satisfaction and use of the computer will equate to increased productivity. To improve end user computing satisfaction and usability, software developers added features (tools) to help end users.

Today these tools, such as The Wizard (Microcoft), Coaches (Adaptiv), and Design Tips (Harvard Graphics for Windows), etc. are being sold as on-line productivity modules with end user software. These tools work great, as long as the data and/or the task conforms to the predefined parameters of the software. These tools have built-in assumptions to provide user friendly help about the software and the problem the end user is trying to solve. Although, most of the tasks performed by a decision maker are not predefined or they would already be done by the computer.

The end user is sheltered from verifying model assumptions that have been built into these tools or even challenging the results derived from the computer. From personal observations I have seen end users rely on these tools to the detriment of the job task and their own intellectual abilities. Hubert Dreyfus (1993) believes that "Our risk is not the advent of superintelligent computers, but of subintelligent human beings." I would agree. These tools are shaping what the end user will do with data in constricting models. It is driving the kinds of decisions that the end user will make in the future.

The objective behind the development of these tools was to make the software easier to use, thereby selling more software. Unfortunately, the software developers forgot that human decision makers are the ones that see inferences, patterns, or opportunities between objects, which the computer cannot understand by just processing zeros and ones.

It is up to the information system community to challenge this approach to "user friendly" software development, as only a money making gimmick that doesn't in reality enhance end user productivity, unless productivity is narrowly defined by computer utilization. The software development approach we need to pursue is to incorporate cognitive science - how the human mind works - into future projects. This will provide software for the end user that is easy to use, without blindly leading them to conclusions based on an unseen model, with assumptions not defined by the end user, but by a software developer whose only interest is selling software. Measuring end user productivity should evaluate the effectiveness of the decision along with the efficiency of the decision. This lack of productivity effectiveness measures makes the evaluation of end user computing an oxymoron and sends the wrong message to software developers.

REFERENCE

Dreyfus, H. (1993). *What computers STILL can't do: A critique of artificial reason.* Cambridge, MA: The MIT Press.