

# End–User Computing Success Measurement

**Conrad Shayo**

*California State Univeristy, San Bernardino, USA*

**Ruth A. Guthrie**

*California State Polytechnic University, Pomona, USA*

## INTRODUCTION

End-user computing (EUC) is the optional development of computer applications and models by personnel (individuals or groups) outside the MIS department. The emergence of EUC in the 80s and early 90s can be traced to the proliferation of computers, increased organizational computing needs, more sophisticated user application development tools coupled with higher computer and information literacy among staff and professional workers. Prior to the arrival of personal computers and graphical user interfaces, end users relied on data processing (now information technology) personnel to assist in meeting their information needs (Inman, 1986). Programming a mainframe was beyond the skills of most workers. Problems identified during this era of computing include:

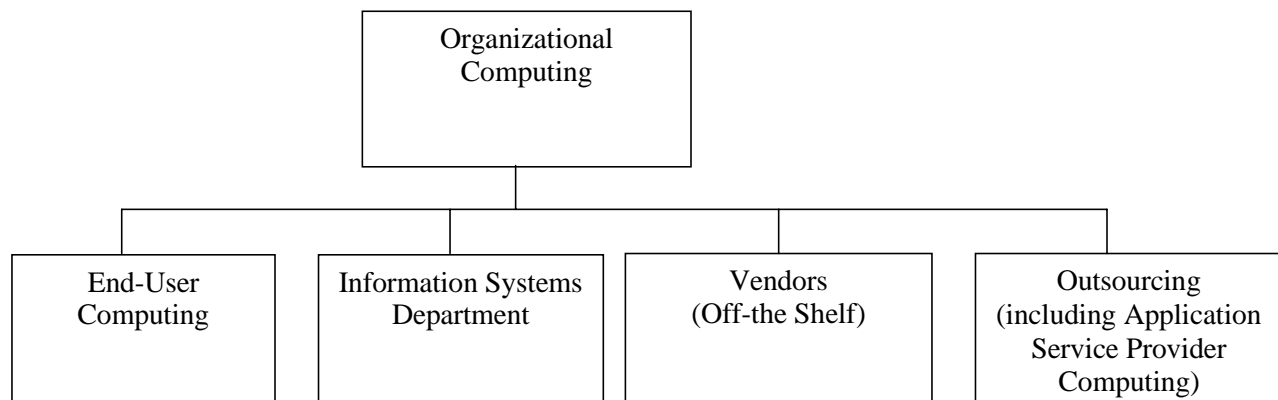
- Failure to meet end-user needs.
- Cost of developing end-user applications was high.
- Large backlog of end-user applications made the development very slow.

As users required more information for decision making and highly user-friendly applications became available, end users began developing customized solutions

to the needs that the data processing departments could not (Ahituv, Neumann & Riley, 1994). Today, EUC has become commonplace. Small, customized applications with spreadsheets and databases are commonplace in end-user departments. At present, EUC is just one contributor to overall organizational computing. As shown in Figure 1, the other sources of computing include applications developed by the information systems department (ISD), applications developed by vendors (off-the-shelf), and outsourcing—including application service providers (ASPs). EUC success is therefore just one contribution to overall organizational computing success. For our purposes, we define EUC success as the degree to which the organizational EUC strategy contributes to individual, group, and organizational computing success in an environment that includes applications developed by the information system department (ISD), application service providers, outsourcing parties, and of-the-shelf vendors. This means EUC complements the other components of organizational computing.

The type of applications developed by end users include transaction processing systems, manufacturing systems, expert systems, executive information systems, decision support systems, and online application processing systems (McLean, Kappelman & Thompson,

*Figure 1. Components of organizational computing success*



1993). There are problems, however: although end-user developed applications are low risk, localized and quickly meet user needs, unlike applications developed by the ISD or vendors, they lack integration, standardization, documentation and quality control. They also lack security, data consistency, and may result in duplication of effort. Table 1 compares the characteristics one would find in the different types of organizational computing.

In this article, we review the major research studies on EUC success measurement focusing on what has been accomplished and what remains to be done. We conclude that the measurement of EUC success seems to be an intractable problem. For example, we identified among others that there is shortage of longitudinal EUC measurement studies. There is lack of studies that have controlled for task, technology and work context. Also, there is lack of research about the relationship between EUC and other forms of organizational computing.

## BACKGROUND

### Measuring the Elements of EUC Success

Figure 1 indicates that EUC success should be measured as an embedded unit of organizational computing success. One problem is that the specific objectives of EUC are often invisible to the end user and to the company. The extent to which the objectives are attained is also unknown because end users often develop applications without knowledge of how their actions impact the other embedded units of organizational computing. End-user developed applications are rarely tracked by organizations. At the same time, it is not difficult to find organizations where an end-user developed application (e.g., DSS) is critical to daily operations. Furthermore, end users may be unwilling to allow objective measurement of the

Table 1. Characteristics of types of organizational computing

	<b>End-User Computing</b>	<b>Information Systems Department</b>	<b>Vendors (Off-the-Shelf)</b>	<b>Outsourcing</b>
<b>Cost</b>	Low/None	Moderate	Moderate	High
<b>Schedule</b>	Immediate	Slow	Fast	Slow
<b>Size</b>	Small	Moderate	Large	Moderate/Large
<b>Control</b>	Low	High	High	High
<b>Risk</b>	Low	Moderate	Low	High
<b>Influence</b>	Local	Local to Organizational Levels	Local to Organizational Levels	Local to Organizational Levels
<b>Fit to User Task</b>	High	Moderate	Low	Moderate
<b>Support (help desk, documentation, training)</b>	Low/None	Moderate	Moderate	Moderate
<b>Integration</b>	Low	Moderate	Low	High
<b>Security</b>	Low	High	High	High
<b>Data Quality and Integrity</b>	Questionable, Sometimes higher quality than corporate systems	High	Moderate	High
<b>Duplication of Effort</b>	High	Low/None	Low/None	Low/None
<b>Acceptance</b>	High	Low	Moderate	Moderate

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/end-user-computing-success-measurement/14384](http://www.igi-global.com/chapter/end-user-computing-success-measurement/14384)

## Related Content

---

### On the Convergence of Analysis and Design Methods for Multi-Agent, Component-Based and Object-Oriented Systems

Bernard Moulin (2001). *Information Modeling in the New Millennium* (pp. 189-198).

[www.irma-international.org/chapter/convergence-analysis-design-methods-multi/22988](http://www.irma-international.org/chapter/convergence-analysis-design-methods-multi/22988)

### Words Matter: How Prompt Framing Shapes Strategic Behaviour in Large Language Models

Milan Koínek and Kamila Štekerová (2026). *Journal of Cases on Information Technology* (pp. 1-28).

[www.irma-international.org/article/words-matter/398628](http://www.irma-international.org/article/words-matter/398628)

### Applying Constructivist Self-Regulating Learning Approach for ICT Students

Yuk Kuen Wong and Donald Vance Kerr (2009). *Encyclopedia of Information Communication Technology* (pp. 40-54).

[www.irma-international.org/chapter/applying-constructivist-self-regulating-learning/13338](http://www.irma-international.org/chapter/applying-constructivist-self-regulating-learning/13338)

### A New Approach for Schematics for Public Transport Spider Maps

João Tiago Ribeiro, Rui Riço and António Leal (2014). *Information Resources Management Journal* (pp. 67-87).

[www.irma-international.org/article/a-new-approach-for-schematics-for-public-transport-spider-maps/109533](http://www.irma-international.org/article/a-new-approach-for-schematics-for-public-transport-spider-maps/109533)

### Alignment with Sound Relationships and SLA Support

AC Leonard (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 130-135).

[www.irma-international.org/chapter/alignment-sound-relationships-sla-support/13561](http://www.irma-international.org/chapter/alignment-sound-relationships-sla-support/13561)