MITIGATING THE ENTERPRISE APPLICATION BOTTLENECK WITH LOW CODE DEVELOPMENT PLATFORMS

Erik Krogh, Graziadio Business School, Pepperdine University, 6100 Center Drive, Los Angeles, CA 90045, 310-568-5528, erik.krogh@pepperdine.edu

ABSTRACT

Enterprise IS Organizations face a chronic undersupply of trained programming professionals and growing demand for application functionality, leading to an application supply-demand disequilibrium and resulting in missed business opportunities. A new breed of Low Code Development Platform (LCDP) tools are maturing and are poised to deliver value to enterprises by allowing end-users to rapidly develop and deploy applications to exploit these missed opportunities. These tools make application development by non-professional-programmers a viable possibility, albeit not without risks to the enterprise. LCDPs alleviate many of the negative aspects of Shadow IT because they provide built-in security and have been pre-tested to foster a safer development environment for business users.

Keywords: End-user development, Low Code Development Platform, End-user computing, Shadow IT

INTRODUCTION

Enterprise IS Organizations (ISOs) do not have a good track record for meeting their business constituents' demand for software applications [1]. As information technology (IT) pervades every aspect of business, this demand for rapid delivery of software is exacerbated. IS executives have long sought ways to improve their ability to deliver applications: noted initiatives include component-based approaches such as Rapid Application Delivery (RAD) and the Agile methodology. As organizations flatten and become more decentralized, another avenue, end-user development (EUD), appears poised to help ameliorate the enterprise application backlog dilemma. EUD is defined as methods and tools that allow non-professional programmers to create and/or modify software applications [2] and can consist of such activities as application customization, component configuration, and programming [3]. The "IT Productivity Paradox", long cited as evidence that investment in computer technology does not yield proportional productivity gains, may be challenged by end-users developing some of their own applications and delivering real business value. While costs for software development may rise due to non-professional programmers' learning curves, the costs for software adoption in the context of use and the costs for missed opportunities for appropriation are reduced significantly [4].

Business users have long demonstrated their desire for IT tools that they can customize to meet their needs. "End-User Computing" (EUC), an end-user programming model prevalent in the 1980s and 1990s, primarily utilized the available technologies of spreadsheets and fourth generation languages (4GLs) as its main toolset. End-User Computing has been somewhat discredited as a viable model due to the proliferation of single-use spreadsheets and databases throughout the enterprise. This fragmentation of applications and data is counterproductive to the mission of providing a "single truth" to business decision makers.

End-users know their business: they possess deep subject matter domain knowledge (e.g. business rules and algorithms). Many are very comfortable with technology, and the IT comfort level will only rise as more Digital Natives (also known as Millenials) enter the workforce [6]. The infection point of Web 2.0, where end-users became contributors as well as consumers of technology, accelerated corporate adoption of IT. IS research consultant Gartner Group calls non-professional programmers who operate outside the enterprise IS organization "citizen developers" [5].

Along with the benefits associated with EUD, there are significant challenges to its use such as security and architectural incompatibility. Companies can lose millions of dollars due to calculation errors on spreadsheets [6]. University of California Santa Barbara Computer Science Professor Tevfik Bultan notes that "everywhere applications", which are developed by non-professional programmers, are not dependable. [7] End-user developers (the "owners" of the business problem) also develop software using a different mental model than professional developers (the "owners" of the technology), potentially resulting in inattention to the many technical details required to make software safe and effective [8].

END-USER DEVELOPER PROFILE

IT competency is relatively high throughout the ranks of business users in today's enterprise. Previous research [8] based on [9] has produced a model of the varying degrees to which software users and software developers participate in the development process. Figure 1: Software development continuum" illustrates the types of applications non-IS professionals could be allowed to develop.

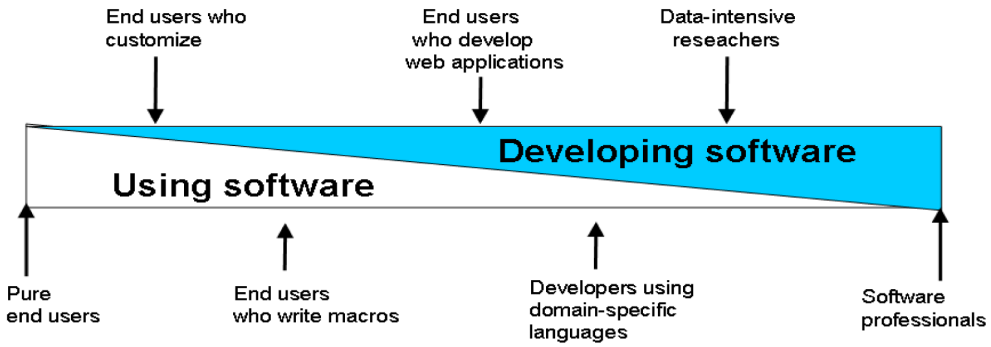

Figure 1: Software development continuum [8]

The challenges associated with developing complex enterprise software to ensure architectural compatibility, security, stability, and future maintainability would preclude part-time developers from participation: they already have full-time jobs making money for the enterprise. Many do not have the technical training or the organizational structure required for industrial-strength programming activities. However, as one business unit manager expressed, increasing numbers of business unit employees are developing IT expertise:

> *"Employees on the business side used to be considered technical if they knew Excel. Now Excel knowledge has been replaced by facility with programming tools. There are a lot of MBAs with an engineering background, who are very tech-savvy, who are excited to*

learn the business. You will find quants that know the business but also know C++, SAS, and MathLab."

Programming applications such as customer-facing web-based marketing apps, departmental reports, develop workflows, or mashups might be considered good candidates for end-user efforts. ERP systems, long lamented by enterprises for their inflexibility (e.g. either the existing business process must be modified to conform to that built into the software or the ERP software must be modified to fit the process) presents opportunities for end-users to build their own functionality to extend the ERP capabilities to suit their own purposes. Considerable EUD research efforts vis-à-vis ERP have been delivered by European researchers [2] and [10].

Not only must end-users learn development practices, but IS developers must learn the nuances of business. A matrix of application categories and developers based on Figure 1 is proposed. The columns specify developer capabilities (e.g. *End-user with no development capabilities* to *IS professional developer*), and the rows list the types of applications for which each user category is qualified. Using the continuum of using/developing software by employees possessing varying IT development capabilities can serve as a guide to producing such a matrix (see Figure 2: Application Development Assignment Matrix).

	End-user with no dev capabilities	User with minimal dev capabilities	User with mid-level dev capabilities	User with high dev capabilities	IS professional developer
Single-use spreadsheet	X				
Marketing blog		X			
Departmental report			X		
Departmental management system				X	
ERP integration					X

Figure 2: Application Developer Assignment Matrix

LOW CODE DEVELOPMENT PLATFORMS: AN ALTERNATIVE TO SHADOW IT

While EUD can deliver benefits to the enterprise, "Shadow IT", defined by Gartner as "IT devices, software and services outside the ownership or control of IT organizations" [11] has long been a problem for ISOs. Issues associated with Shadow IT include applications not coded to standards, data fragmentation resulting from stand-alone apps, and lack of application integration. Gartner predicts that "by 2020, one third of successful attacks experienced by enterprises will be on their shadow IT resources" [12]. Acknowledgement by ISO leaders of the positive business value of EUD can facilitate open and honest discussions with business users that can ultimately deliver better outcomes for the enterprise. For example, by supporting the safe use of EUD through dialogue with business users, the ISO can also raise awareness of its potential negative consequences. It is often difficult to categorically define an instance of EUD as either beneficial or harmful, and having business users consider the ISO a trusted consultant rather than an obstacle to be avoided can help to avert disaster as more scrutiny is directed to digital safety.

Modern IT affordances such as Cloud Computing have overcome what were once barriers to citizen developers creating their own applications. A relatively new class of software development tools suitable for EUD is Low Code Development Platforms (LCDP). LCDPs are "platforms that enable rapid delivery of business applications with a minimum of hand-coding and minimal

upfront investment in setup, training, and deployment" [13] . Maturing LCDP tools are now available to harness the potential of end-users writing their own software. These LCDP tools boast such features as graphical user interfaces, wizards, and property sheets that require little 3GL programming ability. The vendors of these products claim that they are easy to use, and many allude to the fact that end-users can rapidly build and deploy their own applications. LCDPs allow end-users to drag-and-drop components to easily build workflows and implement business logic. They also support scalability, security, and reduce unit testing as many LCDPs have already tested components to be usable out-of-the-box.

As businesses roll with the waves of discontinuous change, time-to-market is a desired attribute in most enterprises, and agility in developing and deploying software is key [14]. Most of the LCDP tool vendors prominently advertise their products as "agile" or "RAD" tools, an acknowledgement that their products need to be easy to use in order to reach the deployment stage as early as possible.

End-users do not have significant work time to invest in learning new technologies. The propensity to accept LCDP tools "will be inversely proportional to product complexity and variability in the user population" [15]. Developing highly accessible and effective interfaces has long been studied by the IS academic discipline, and EUD-specific studies focused on producing tools for domain-expert users (i.e. business users who work in a specific business functional area) have been published since the early 2000s [16].

THEORETICAL UNDERPINNINGS

Supporting the acceptance of LCDPs by end-users are relevant theories including Role Theory, the Technology Acceptance Model, and Transaction Cost Theory.

Role Theory should be considered when making decisions regarding development assignments. As it would appear, Role Theory is concerned with "the necessary division of labor" in an organization and the roles assumed by the actors within it [17]. This theory is useful in understanding how business users (i.e. non-professional programmers) will adapt to a different role in the enterprise, that of a part-time application developer.

The Technology Acceptance Model (TAM) is concerned with whether users will embrace a new technology tool. Users will evaluate the perceived usefulness of the tool as well as its perceived ease-of-use [18]. Based on TAM, business users should accept LCDPs if both usefulness and ease-of-use are perceived positively. TAM is important as the focus of this study is how easy LCDP tools are to use.

Transaction Cost Theory (TCT) posits that end users will self-source IT if they perceive that the cost in time and money is less that the cost would be to source through their ISO, thus using an EUD solution such as an LCDP instead of the ISO-supplied solution would be advantageous.

CONCLUSION

LCDPs are acknowledged as a development model that is gaining traction in the practitioner literature. The easy-to-use development tools of LCDPs combined with their built-in security capabilities and ability to scale should ease the ISO's Shadow IT fears and encourage end-user developers to work within an approved framework to deliver digital capabilities for which the ISO

does not have the capacity. Software vendors are investing considerable capital in producing new LCDP tools as well as refining their existing ones. While LCDP tools have matured, the question regarding the ability of non-professional programmers to intuitively use these tools continues to be at issue for those enterprises considering an EUD strategy. However, there is some theoretical support to suggest that EUD tools in the form of LCDPs should be accepted and used by business users intent on developing applications and workflows. The lack of academic research in this area presents an opportunity for scholars to study how maturing EUD technologies such as LCDPs can impact the expediated delivery of business capabilities in the Digital Economy.

REFERENCES

[1] Symons, C., "The Balanced Scorecard for IT: User Metrics", Forrester Research , March 15, 2004

[2] M. Spahn, V. Wulf, "End-User Development of Enterprise Widgets", IS-EUD '09**:** Proceedings of the 2nd International Symposium on End-User Development, Publisher: Springer-Verlag

[3] G. Fischer, E. Giaccardi E, Y. Ye, A.G. Sutcliffe, N. Mehandjiev, "Meta-Design: A Manifesto for End-User Development", Communications of the ACM,Volume 47 Issue 9, September 2004

[4] V. Wulf, M. Jarke, "The Economics of End-User Development", COMMUNICATIONS OF THE ACM, September 2004/Vol. 47, No. 9

[5] Garter IT Glossary, Retrieved September 20, 2018 from https://www.gartner.com/it-glossary/citizen-developer/

[6] M. Burnett, "End-User Software Engineering and Why it Matters", Journal of Organizational and End User Computing (JOEUC), Editor(s)-in-Chief: Mo Adam Mahmood (Ed.) (University of Texas at El Paso, USA), Volume 22, Issue 1, 2010

[7] T. Bultan, "Software for Everyone by Everyone", Proceedings of the FSE/SDP workshop on Future of software engineering research ACM New York, NY, USA 2010

[8] M.F. Costabile, P. Mussio, L. P. Provenza, A. Piccinno, "End Users as Unwitting Software Developers", ACM Proceedings of the 4th International Workshop on End-user Software Engineering, May 2008

[9] Ye, Y. and Fischer, G. 2007. Designing for Participation in Socio-Technical Software Systems. Proc. HCII 2007 (Beijing, China, Jul. 22-27, 2007). LNCS,Springer, 312-321.

[10] C. Dörner, J. Heβ, V. Pipek, "Improving Information Systems by End User Development: A Case Study", Proceedings of the 4th international workshop on End-user software engineering, New York, NY, USA 2008

[11] Garter IT Glossary, Retrieved September 20, 2018 from https://www.gartner.com/it-glossary/shadow

[12] Gartner, "Gartner's Top 10 Security Predictions 2016," Retrieved September 20, 2018 from https://www.gartner.com/smarterwithgartner/top-10-security-predictions-2016/?cm_mmc=social-_-rm-_-gart-_-swg

[13] Richardson, C., Rymer, J.R., "The Forrester Wave: Low-Code Development Platforms, Q2 2016

[14] X. Liang, I. Marmaridis, A. Ginige, "Facilitating Agile Model Driven Development and End-User Development for Evolving Web-based Workflow Applications", IEEE Proceedings ICEBE '07 Proceedings of the IEEE International Conference on e-Business Engineering

[15] A. Sutcliffe, "Evaluating the Costs and Benefits of End-User Development", ACM Proceedings of the first workshop on End-user software engineering, New York, NY, USA 2005

[16] M.F. Costabile, D. Fogli, G. Fresta, P. Mussio, A. Piccinno, “Building environments for end-user development and tailoring”, Proceedings HCC '03 Proceedings of the 2003 IEEE Symposium on Human Centric Computing Languages and Environments IEEE Computer Society Washington, DC, USA 2003
[17] D. Galletta, R. L. Heckman, Jr, “A Role Theory Perspective on End-User Development”, Information Systems Research, Vol. 1, No. 2, June 1990, pp. 168-187
[18] Davis, F. D., “Perceived usefulness, perceived ease of use, and user acceptance of information technology”, MIS Quarterly, September 1989