Int. J. Production Economics 168 (2015) 211–223

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Real options, learning cost and timing software upgrades: Towards an integrative model for enterprise software upgrade decision analysis

Horatio M. Morgan [a,1], Ojelanki Ngwenyama [b,c,*,2]

[a] Ted Rogers School of Management, Ryerson University, Canada
[b] Institute of Innovation and Technology Management Ryerson University, Canada
[c] Faculty of Commerce, University of Cape Town, South Africa

ARTICLE INFO

Article history:
Received 7 November 2014
Accepted 28 June 2015
Available online 6 July 2015

Keywords:
IT decision model
Learning-based approach
Real options analysis
Software upgrades

ABSTRACT

A key challenge facing information technology (IT) managers is how to carefully analyze the decision options available to them when considering enterprise software infrastructure upgrades. We present an illustrative case that not only captures the trade-offs involved in retaining an existing software infrastructure as opposed to adopting a new one at an appropriate time, but also demonstrates how the combined application of various analytical tools including real options analysis may provide richer information than a single approach. Our model offers IT managers the potential to arrive at a deeper understanding of software upgrade timing decisions while generating important information relevant to practical decision situations.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Enterprise software infrastructure is a general purpose information technology which supports not only the productive capabilities of an organization but also its strategic core capabilities as well (Bresnahan and Greenstein, 2001; Brynjolfsson and Hitt, 1995; Yen and Sheu, 2004). As such, enterprise software infrastructure requires continuous upgrading to improve productivity and sustain the organization's competitiveness (Voulgaris et al., 2015; Hawking and Sellitto, 2015). Deciding when to upgrade critical enterprise software infrastructure is a challenging problem for senior management (Choi et al., 2013; Ngwenyama et al., 2007; Olson and Zhao, 2007; Khoo and Robey, 2007; Ng, 2001). Upgrading enterprise software infrastructure is a significant investment that may impact the efficiency and competitiveness of the enterprise; furthermore, it is costly in terms of downtime, implementation and learning (Gebauer and Schober, 2006; Ngwenyama et al., 2007; Ashurst et al., 2008; Demsey et al., 2013). In addition, vendor charges for software infrastructure upgrades and maintenance are relatively high, and their software upgrade release life cycle is often excessive (Gable et al., 2001; Irani et al., 2006; Sahin and Zahedi, 2001a; Ellison and Fudenberg, 2000; Jansen and Brinkkemper, 2006). But in our hyper-competitive global economy, enterprises are dependent on software infrastructure for providing timely service to clients, continually improving their operating efficiency and effectiveness, and managing supply networks that extend across geography and time zones. Consequently, billions of dollars are spent each year on new software infrastructure with the expectation of high returns on productivity and competitiveness (Osei-Bryson and Ko, 2004; Jurison, 1996a, 1996b; Demsey et al., 2013).

In the United States (U.S.) alone, spending on information technology (IT) for 2014 has reached US$1.03 trillion (Bartels et al., 2014). Much of this spending is on upgrading existing software infrastructure, such as enterprise resource planning (ERP) system upgrades, which the Gartner group forecasted to be close to US$4 billion in 2014 (Gartner, 2013). However, studies have shown that these IT investments do not necessarily meet the expectations of senior managers (Doherty et al., 2012; Cecez-Kecmanovic et al., 2014), and those IT projects that ultimately fail can be very costly for firms in terms of competitiveness and lost market value (Bharadwaj et al., 2009). A key challenge that IT managers face is how to carefully analyze the decision options available to them when considering software upgrades geared toward improving their cost-performance. Some researchers have raised concerns that IT upgrade decisions are not well researched (Khoo and Robey, 2007; Light, 2001), and others have raised concerns about the limitations of existing decision models to support a comprehensive analysis of the range of IT decision-making problems that managers face (Ngwenyama et al., 2007;

* Corresponding author.
E-mail addresses: horatio.morgan@ryerson.ca (H.M. Morgan), ojelanki@ryerson.ca (O. Ngwenyama).
[1] Tel.: +416 979 5000x2526.
[2] Tel.: +416 979 5000x4203.

http://dx.doi.org/10.1016/j.ijpe.2015.06.028
0925-5273/© 2015 Elsevier B.V. All rights reserved.

Osei-Bryson and Ngwenyama, 2008; Plaza and Rohlf, 2008). Other researchers have repeatedly pointed out that decision models developed by academics are often inaccessible to practicing managers, and advise a focus on accessibility and applicability (Ball, 1985; Little, 1986, 2004; Kasanen et al., 2000; Čibej, 2002; Lilien, 2010; Jarzabkowski and Kaplan, 2015).

In this paper, we respond to the call for decision support models which are accessible to practicing managers and applicable to real-world IT management problems. We contribute to the literature by presenting an illustrative case that sheds new light on the software-upgrade timing problem. A major challenge that arises in developing accessible decision analysis tools is how to strike a balance between simplicity to ensure tractability, and realism to accurately convey the complexity and difficulty of real-world software upgrade decisions. We focus on an analytically manageable, but non-trivial choice between the following two alternatives: (1) the retention of an existing ERP system under which maintenance costs increase annually after the vendor discontinues user support, and (2) timing the adoption of one of two versions (*technical* versus *functional*) of a newly released ERP system that contain maintenance costs, but require substantial upfront expenditures at a time when estimates of the benefits to be had are not reliable.

We not only recognize that prior research has sought to enhance ERP cost estimation through the development of the COnstructive COst MOdel (COCOMO), among others (Kotb et al., 2011), but also that the application of real options analysis (RoA) to IT-related investment problems is not new to the production economics and information system (IS) literatures (Brynjolfsson and Hitt, 1995; Dimakopoulou et al., 2014; Fichman, 2004; Kyläheiko et al., 2002; Yen and Sheu, 2004). Our primary focus is on providing a decision model accessible to IT managers, and illustrating its usefulness to a strategic analysis of the software upgrade problem. Our research objective differs from other studies that emphasize more conceptually complicated approaches to real options analysis which may limit the discourse to academics (Datar and Mathews, 2004). Our research program seeks to advance production economics and IS/IT management research by opening up a wider discussion and research on the conceptualization, measurement and tracking of uncertainty in the context of real options applications (Collan et al., 2003, 2009).

The remainder of this paper is organized as follows. In Section 2, we briefly discuss the relevant literature on the software upgrade problem. In Section 3, we present an options-based framework for the analysis of the option to wait. Drawing on this framework, we present and analyze an illustrative case example in Section 4. Section 5 concludes with a discussion of the main practical implications, and directions for future research.

2. The software upgrade problem

Deciding when to upgrade an enterprise's software infrastructure is a dilemma that most IT managers face (Demsey et al., 2013; Ellison and Fudenberg, 2000; Khoo and Robey, 2007; Ngwenyama et al., 2007). The timing of software infrastructure upgrades is often driven by the software vendor's product life cycle and profit maximization goals, and not by the implementing enterprise's strategic objectives (Light, 2001; Khoo and Robey, 2007; Demsey et al., 2013). Increasingly, IT managers are pressured into upgrading technologies before their organizations have achieved the expected return on investment from their last upgrade/new implementation (Demsey et al., 2013; Luftman et al., 2013; Ngwenyama et al., 2007). Upgrades of enterprise infrastructure software are frequent, costly and largely dictated by the software vendors (Ellison and Fudenberg, 2000; Law et al., 2010; Sahin and Zahedi, 2001a). At regular intervals vendors announce new releases of their software infrastructures and set a certain date for ending maintenance on their current versions. The releases can be a technical upgrade, which replaces the existing ERP software in the form of a system with a better technical platform and superior performance; or a functional upgrade in the form of key business improvements and superior functionality, but a higher implementation cost (Demsey et al., 2013; Law et al., 2010; McGinnis and Huang, 2007; Nah and Delgado, 2006). While enterprises may have no business imperative to implement the latest version of the software, they might feel that failing to do so may result in a loss of competitiveness, as software vendors often claim that the upgrade will result in productivity improvements (Khoo and Robey, 2007; Ng, 2001; Ononiwu, 2013). Furthermore, enterprise software infrastructure users feel pressure to undertake the upgrade to mitigate the risk of losing support from the vendor (Demsey et al., 2013; Sahin and Zahedi, 2001b).

The cost-benefit challenges of upgrading enterprise software infrastructure are well documented (Light, 2001; Khoo and Robey, 2007; Demsey et al., 2013; Law et al., 2010; Lewandowski et al., 2013). Many organizations use a range of software having different product life cycles that will eventually need upgrading. For example, a medium-size enterprise could have 3000–5000 computer workstations running various software infrastructures. In addition, there could be a number of servers, and a variety of enterprise software applications. If we assume for simplicity, that all the computers are running the identical operating system, a single decision to upgrade the enterprise software infrastructure would result also in upgrading the operating systems of every workstation. Further, upgrading the operating systems might result in the need to upgrade other software applications, as they might be incompatible with the newer version of the operating system. Therefore, when contemplating a software upgrade, IT managers have to consider a variety of costs including: (a) direct software costs; (b) implementation and training costs; (c) the cost of learning new tasks and coordination routines, and (d) lost output or revenue due to a decline in productivity (Argote et al., 1990). The costs and disruption of recurrent software upgrades can be quite burdensome. Implementation and upgrading of enterprise software infrastructure are often the largest investment that IT managers must make, and this investment can have a significant impact on the productivity and competitiveness of the enterprise (Demsey et al., 2013; McGinnis and Huang, 2007; Rettig, 2013; Shang and Seddon, 2002).

2.1. Existing software-upgrade decision models

An increasingly prominent approach to capital budgeting decisions is real options analysis (RoA). Based on prior studies that apply the RoA approach to IT investment decisions (Balasubramanian et al., 2000; Dimakopoulou et al., 2014; Fichman, 2004; Kyläheiko et al., 2002), we now have a better understanding of the value of operational flexibility with respect to the timing and other aspects of IT projects. Furthermore, these studies generally lend support to a later adoption of new technology than IT vendors might suggest (Benaroch and Kauffman, 2000; Taudes et al., 2000). At the same time, it remains unclear how IT managers should proceed when they have to decide between retaining an existing software infrastructure and timing the migration to one of several versions of a newly released software infrastructure.

Another distinct approach that analyzes the timing of IT investments is the learning-based (LcA) approach (Ngwenyama et al., 2007; Guergachi and Ngwenyama, 2011; Plaza et al., 2010). When assessing the costs of adopting a new software infrastructure, organizations commonly account for the one-time acquisition costs with limited

Download English Version:

https://daneshyari.com/en/article/5079525

Download Persian Version:

https://daneshyari.com/article/5079525

Daneshyari.com