LOCKHEED MARTIN
JSO5
Joint Symposium 2005 - "Engineering for Affordability"

An Attribute Proposal for Same Vendor, Version-to-Version COTS Upgrade Decisions

Jason S. Seibel
ITL Architect
Information Systems & Solutions
(703) 456-1655
jason.s.seibel@lmco.com

Abstract

Information Technology (IT) decision makers, who leverage COTS applications, are regularly faced with the challenge of upgrading to the latest version of their COTS products. Microsoft, as well as other vendors, are releasing upgrades to their products every 18-24 months, each purported to be full of new "features" and capabilities. Every new release represents another IT decision and investment for the organization.

There are advantages and disadvantages of using COTS applications. Many of these disadvantages surface when new software versions are released by vendors. As a result, IT decision makers typically employ general strategies that are not based on clear, well-defined decision attributes which, in turn, place the organization at risk. In addition, currently available COTS decision models do not specifically address same vendor, version-to-version upgrade decisions.

This research effort proposes as set of 25 specific decision attributes and their values, organized into 9 categories, which are considered key considerations for same vendor, version-to-version COTS application upgrades. These proposed attributes are based on research and literature reviews that contribute to same vendor, version-to-version upgrade decisions. Finally, this paper ends with recommendations for further research in this subject area.

This paper will build on existing research and literature to enumerate the challenges faced by IT decision makers and why current approaches are inadequate. This paper is organized into the following sections: Section 1 will describe the advantages and disadvantages of using COTS applications. Section 2 will describe the challenges current IT decision makers face. Section 3 will describe strategies and their limitations associated with same vendor, version-to-version upgrades. Section 4 will propose a set of attributes and value ranges, and Section 5 will conclude on recommendations for further research.

Keywords

Upgrades, COTS, software, decision attributes, same vendor, version

August 3, 2005

1. Introduction

Information Technology (IT) decision makers, who leverage COTS applications, are regularly faced with the challenge of upgrading to the latest version of their COTS products. Microsoft, as well as other vendors, are releasing upgrades to their products every 18-24 months, each purported to be full of new "features" and capabilities. Every new release represents another IT decision and investment for the organization.

There are advantages and disadvantages of using COTS applications; however, many of the disadvantages surface when a new version of software is released. Unfortunately, IT decision makers have leveraged baseless "strategies" in order to manage the frequency of vendor release because there are few models, if any, which consider "same vendor" software upgrades, leaving organizations at risk.

This research effort proposes nine key decision attributes and twenty-five contributing sub-attributes intended for "same vendor, version-to-version" COTS software upgrade decisions. Concurrently, this document reveals the result of a survey in which individuals were asked to rank the proposed attributes. Finally, a proposal for the use and application of said attributes concludes this research effort.

2. What is COTS Software?

Commercial Off-the-Shelf Software (COTS) is a coined phrase that has been used since the mid-to-late 1970's as an application designed for reuse. [1] The Webster dictionary defines COTS as a reference "to hardware and software systems that are manufactured commercially and then tailored for specific uses. This is in contrast to systems that are produced entirely and uniquely for the specific application." [2] Carnegie Melon University, Software Engineering Institute describes COTS products as, "sold, leased or licensed to the general public; offered by a vendor trying to profit from it; supported and evolved by the vendor, who retains the intellectual property rights; available in multiple, identical copies; and used without source code modification." [3] Salter and Buede, add that COTS software is normally purchased, "as-is" with the presumption that no modifications can be made to the core application. [4] However, "definitions found in the literature are usually very broad, covering a variety of products. As a result, researchers and practitioners use the same word with different meanings."[5] Maurizio and Torchiano "argue that COTS has to remain a term with broad coverage;" however, they do recognize that classes "of software products and a number of subclasses have to be identified" in addition to the main characteristics of COTS. [5]

3. Advantages & Disadvantages of COTS

Many organizations primarily consider Commercial Off-the-Shelf software verses custom developed software due to the cost and time savings. [6] Organizations are attracted to the ubiquity and cost savings of COTS productivity applications. However, the key to leveraging COTS software is to fully understand both their advantages and disadvantages. One author writes, "To guard against (such) grim situations, approach any COTS integration project clear-eyed and wary." [7] Below is a table describing the advantages and disadvantages of using COTS software.

Advantages	Disadvantages
Cost of developing software is spread across all consumers; cost reduction [8]	Future evolution of the COTS product is not under the customers control; obsolescence risk [9], [10], [11], [12]
Faster delivery of projects since product is already developed [8], [9], [13]	Source code is unavailable [9], [10], [12]
Immediately available [7]	Limitations must be dealt with or additional "glue code" must be written to compensate [2, 7, 10, 11, 12] [9], [10]
Avoids expensive development [7]	Licensing, intellectual property procurement delays [7]
Avoids expensive maintenance [7]	Up-front license fees [7]
Predictable, confirmable license fees [7]	Recurring maintenance fees [7]
Rich functionality [7], [13]	Reliability often unknown or inadequate; scale difficult to change [7], [11, 14]
Broadly used, mature technologies [7]	Too-rich functionality/Unused features [7], [15]
Frequent upgrades often anticipate organization's needs [7]	Constraints on functionality, efficiency [7]
Dedicated support organization [7]	No control over upgrades and maintenance [7], [12]
Hardware/software independence [7]	Dependence on vendor [7], [13]
Tracks technology trends [7]	Integration not always trivial; incompatibilities among vendors [3, 9] [7]
More product solutions [8]	Synchronizing multiple-vendor upgrades [7], [12], [15]
Ability to use an open system design early on [16]	Aggressive upgrade deployment [1, 12]
Partitions the testing effort [14]	Higher risk of security vulnerabilities [14], [12]
Promotes modular design [14]	Vendor behavior radically varies [11], [12], [15]
Less debugging [13]	

Table 1 - Advantages and Disadvantages of COTS Software

4. Risks & Decision Challenges

Information Technology professionals are faced with countless "recommendations" for productivity application upgrades from hungry, profit-driven software vendors. Unfortunately, "for many vendors, the competition for rushing a new version to market is more important than delivering a high level of software reliability, usability and other qualities." [17]

While all software upgrades represent some degree of risk, specific areas related to COTS productivity application upgrade stand forth. Most notably, Information Technology decision makers face two major areas of risk when making a decision: first, the Quantification of Benefits and Costing, and second, the overall management of the product, or described hereon as Product Management.

According to literature research, Information Technology decision makers identified "Quantifying relevant benefits" as a significant problem (81% or 60 out of 75) followed by "actually identifying the relevant benefits of the products (65% or 48 out of 75). [18]

.

Other, specific areas of product quantification challenges include: "assessing the maturity of products," "quantifying the actual cost of upgrading" beyond licensing cost and "identifying forecasting activities" in order to identify decision criteria. [15] [19]

These challenges of quantifying and identifying benefits, the maturity of products, costing and forecasting activities represent precursor activities to Product Management risks.

Product Management describes the concept of managing and sustaining the overall COTS application in their operating environments. Many of the risks and challenges faced during this period are analogous to the concepts described in Table 1 - Advantages and Disadvantages of COTS Software. Literature research, however, has revealed reoccurring challenges faced by decision makers. Specifically, challenges such as product synchronization, implementation, product sustainment, limited vendor support, gaps in capabilities, modification of "glue" code and training.

While the previously noted risks are not collectively exhaustive, research shows that most, if not all, are challenges faced prior to product implementation.

5. Typical Strategies

It is the opinion of this author through professional experience and academic research that organizations have relied heavily on a variety of informal "strategies" to make upgrade decisions for COTS applications. In other words, many of these "strategies" are baseless approaches, techniques and mechanisms used to decide whether a newly released version of an existing or deployed application should be upgraded. Most of these are strategies are undocumented, but "understood" among the IT community and are described hereon:

- *Upgrade to the "latest" version* – when a vendor releases a new version, an organization immediately upgrades. [20]

- *Upgrade to "every other" version* – an organization "skips" a version between upgrades, presumably, because vendors support 2-3 previous releases.[20], [21]

- *Upgrade when support is no longer available* – when support is no longer available from the vendor, the organization upgrades. [22]

- *Upgrade by influence* – when internal or external forces exert enough pressure, the organization upgrades. [23]

Employing the aforementioned upgrade strategies have resulted in catastrophic deployment and operational failures in businesses *and* home users (they're not immune either). Trade magazine articles site days on the phone with technical support [21] and entire networks down for weeks at a time as the result of software bugs, poorly planned upgrades or unexpected incompatibilities. "Upgrade strategies must take into account the impact of new and upgraded products on training, installation, and support." [24]

6. Current Decision Models

System Engineers and Information Technology professionals have attempted to mitigate the aforementioned challenges and risks through the use of generic and custom methodologies. While there are very few, if any, specific "same vendor, version-to-version" COTS software upgrade decision models, there are several, attributable models that consider this context.

Generic models have been applied to upgrade decisions for years. These models are labeled as "generic" because they have been derived from other disciplines or have been applied across a wide variety of decision applications. Quantifiable models such as "Return on Investment (ROI)," "Cost/Revenue Analysis," "Cost/Benefit Analysis" and "Return on Management (ROM)" have assisted decision makers to evaluate upgrade decisions based on mathematical principles. Other qualifiable methods such as "multi-objective, multi-attribute," "Weighted Scoring Method (WSM)" and "Value Analysis" have provided methods for evaluating more abstract or difficult to quantify attributes. Unfortunately however, each of these models lack the COTS software depth necessary to safely and effectively execute upgrade decisions.

Extending beyond basic methods, are those models tailored to Commercial Off-the-Shelf software elicitation, evaluation, comparison and selection. Models such as "Checklist Driven Evaluation Methodology (CDSEM)," "Off-the-Shelf Option (OTSO)," and "COTS-based on Requirements Engineering Model (CRE)," are all geared toward "*different vendor*, same product" application selection. While these models weren't specifically designed for "same vendor, version-to-version" upgrades, they can be applicable if modified appropriately.

Another model, the Rapid Response Technology Trade (R2T2) study, unique to his own model, is based on a patented mathematical optimization routine developed by Tom Herald of Lockheed Martin. The model is designed to identify "the ideal point for a Technology Refreshment" through the use of four primary input attributes: the "technology life cycle," the "current technology maturity," the "technology change frequency" and the "technology double period." [22] R2T2, while originally designed for military hardware systems, has been successfully applied to enterprise information technology systems.

7. Attribute Proposal

The goal of this research effort was to identify key attributes when considering "same vendor, version-to-version" upgrade decisions. Extensive literature research failed to identify a single, comprehensive list of related decision attributes. In addition, a majority of the COTS decision models, specifically in the requirements phase of systems engineering, only spoke for the need to identify said attributes.

Therefore, through extensive literature research, a list of key decision attributes and contributing sub-attributes were identified and are proposed hereon. Each attribute and sub-attribute have been referenced to specific sources and described with a brief explanation, as noted in the following tables.

7.1 *Cost Attribute*

Cost Attribute		
	Contributing Attribute	**Explanation**
1.	Software License Acquisition	Cost of acquisition [25], [9], [26]
2.	Integration Cost	Cost of integrating new product [9], [27] [28]
3.	Training Cost	Training end users and support personnel on the new version [25], [27], [28]
4.	Hardware Upgrades	Cost of upgrading equipment [25]
5.	Glue Code Cost	Cost of custom code used to integrate product. [9], [26]
6.	Infrastructure Upgrade Cost	Cost to upgrade dependent software [28]
7.	Testing Cost	Cost of testing product (regression, integration) and/or identifying product bugs [28], [26]
8.	Modification Cost	Cost of modifying (applying settings within) COTS software [28]

Table 2 - Cost Attribute

The cost of upgrading any product is always a consideration when deciding to "upgrade or not." Researchers and decision makers have attempted to model and estimate cost points of Information Technology projects for years using generic methods such as "Cost-Benefit Analysis" and "Return on Investment." In 2004, the U.S. Treasury department is expected to spend ~1% ($2,300,000) of its $2.3 billion budget on software upgrades alone. [29]

7.2 *Knowledge Attribute*

Knowledge Attribute		
	Contributing Attribute	**Explanation**
1.	In-house Expertise	In-house staff has knowledge to engineer and deploy upgrade [30], [31]
2.	End-user Expertise	End users have the knowledge to leverage new capabilities [32]

Table 3 - Knowledge Attribute

The knowledge of support staff and end users is required in order to successfully evaluate, implement and support COTS software upgrades. Leveraging in-house expertise to engineer automated software deployments reduces or eliminates the need for vendor consultants. In addition, end users who have the knowledge to leverage new software capabilities reduces training costs by requiring an "update" training session rather than training on the entire COTS product.

7.3 *Stability Attribute*

Stability Attribute		
	Contributing Attribute	**Explanation**

1.	Future of the Product	Direction of the product; expected life of product [30], [33], [31]
2.	Robustness of the Product	Historical reliability; have previous version upgrades proved reliable? [30], [31], [17], [21]
3.	Stability of the Vendor	Financial stability of vendor; vendor support; number of employees; industry certifications; overall product offering maturity [33], [31], [7], [34], [35]
4.	Stability of the Industry	Stability of vertical market; level of IT development related to product line; pundit forecasts [35]

Table 4 - Stability Attribute

The dictionary defines stability as, "the state or quality of being stable: especially, reliability; dependability." [2] Stability in the context of *same vendor, version-to-version* upgrades consists of four specific, but equally important attributes. These attributes consider the future and robustness of the product, the financial viability and stability of the vendor and the overall stability of the industry related to the product's vertical market. This attribute is important because decision makers want to ensure their investment can be supported throughout the intended life of the product.

7.4 *Licensing Structure Attribute*

Licensing Structure Attributes		
	Contributing Attribute	**Explanation**
1.	Licensing structure	Licensing fee structure (i.e., per seat, per server, per processor, lease, version-to-version upgrade requirements) [30], [17], [31], [36], [15]

Table 5 - Licensing Structure Attribute

Licensing, in the context of Commercial Off-the-Shelf software, refers to a purchase for the *right* to use a software package from the developer. A Google search reveals several similar definitions such as, "agreement that allows an individual or group to legally use a software program" [http://mts.admin.wsfcs.k12.nc.us/admin/techplan/techpl16.html] and "A contract between a license producer (for example, Digital) and a license receiver (customer) that grants permission to use a specific software product as described by the applicable Software Product Description (SPD), and the terms and conditions of the license contract." (Hewlett-Packard)

The licensing category contains a single attribute: licensing structure. The structure is important because it can impact the actual cost of acquiring, using and maintaining the software. There are two fundamental licensing types: license and subscription (lease). Licensing allows an organization to make a one-time expenditure to use the software indefinitely. Subscribing, however, allows an organization to make small, time-based payments while it uses the software. However, upon termination of the subscription, the organization can no longer use the product; however, any upgrades during the subscription period are free.

7.5 *Productivity / Performance Attribute*

Productivity / Performance Attribute

	Contributing Attribute	Explanation
1.	Productivity Increase (performance)	Improves user productivity (directly/indirectly) [30], [37], [38]
2.	Reduced Cost	Reducing overhead or saving money [30], [37], [35]

Table 6 - Productivity / Performance Attribute

Table 6, Productivity / Performance attribute considers whether the software upgrade will impact productivity or performance. One of the reasons organizations invest in technology is with the expectation they can reduce costs, generate revenue or increase productivity. Vendors attempt to meet business's needs by providing software aimed at those expectations.

7.6 *Meets Business Goals Attribute*

Meets Business Goals Attribute		
	Attribute	**Explanation**
1.	Capabilities	Gaining new IT capabilities; significant new capability (i.e., major or minor release) [30], [17], [20]
2.	Supports Business Objectives	Implicit, explicit, general, supports "x-year plan," solves current business problems today, number of business functions supported [18], [39], [33], [34], [37], [31], [20]
3.	Meets End User Needs	Maps directly to end user needs; end users requesting software capabilities [37], [20]

Table 7 - Meets Business Goals Attribute

"A critical organizational objective is the alignment of technology with organizational objectives." [40] The software upgrade, at a minimum, should provide the organization with new capabilities, support the overall business objectives and meet end users needs. The attribute, "Meets Business Goals" considers whether the proposed upgrade contributes to the objectives of the business in three specific areas.

7.7 *Compatibility Attribute*

Compatibility Attribute		
	Contributing Attribute	**Explanation**
1.	Operates on Existing Platform	Product does not require system upgrades (i.e., hardware, dependent software, glue code) [17, 20], [31]

2.	Backward Compatibility	Product is backward compatible with previous versions (i.e., file formats, related products) [20]

Table 8 - Compatibility Attribute

Compatibility in the context of same vendor, version-to-version upgrades, refers to the productivity application's ability to cooperate within a particular environment and with previous versions of itself without failure to itself or other cohabitating applications. In other words, the decision to upgrade should consider the application's overall compatibility with its intended environment.

7.8 *Product Obsolescence Attribute*

Product Obsolescence Attribute		
	Contributing Attribute	**Explanation**
1.	Current Product is Obsolesced	Produce is no longer available or has reached end-of-life [20], [15]

Table 9 – Product Obsolescence Attribute

The attribute Product Obsolescence, evaluates whether the current, deployed software version has reached "end-of-life" and the vendor has discontinued product support. This is important because organizations can no longer expect product patches (i.e., bug, security or compatibility patches) or technical support from the vendor. While some may believe product obsolescence is an automatic upgrade, other attributes described within this research effort may counter *same vendor, version-to-version* upgrades and therefore should be considered carefully.

7.9 *External Drivers Attribute*

External Drivers Attribute		
	Contributing Attribute	**Explanation**
1.	Regulations/Policies	Government regulations or policies [30]
2.	External Customers	Required by external "customers" [23]

Table 10 - External Drivers Attribute

The decision attribute, External Drivers, is used to describe whether external pressures, outside of the organization, are influencing or "driving" an upgrade. External drivers may be encouragement from customers to support business-to-business capabilities or recommendations from the United States government.

8. Survey Results: Attribute Prioritization

In a survey conducted by this author in January of 2005 entitled, "A Version-to-Version Upgrade Decision Survey for Commercial Off-the-Shelf Productivity Applications," participants were asked to rank the previously proposed upgrade decision attributes based on the overall contribution to the

decision process. Participants were directly elicited based on academic achievements, professional experience or executive position within an organization. Forty-two responded as:

Decision Attribute	Percentage
Meets Business Goals	39.47%
Application's Impact on Productivity	23.68%
License Cost	15.79%
New Version Compatibility	13.16%
External Drivers	5.26%
Licensing Structure	2.64%
Technical Knowledge	0.00%
Vendor Stability	0.00%
Obsolescence	0.00%

Table 11 – Results of Survey: Decision Attribute Ranked as Number One.

The table is interpreted as, "Of the forty-two respondents, when asked to rank each of the nine attributes, 39.47% ranked 'Meets Business Goals' as the most important decision attribute." Interestingly, "Technical Knowledge," "Vendor Stability," and product "Obsolescence" were never ranked with a "number one" and therefore may be construed as less significant in the overall decision process. Further analysis of the data is beyond the scope of this effort.

9. Mitigation Proposal

When considering "same vendor, version-to-version" application upgrades, decision makers should consider each of the described nine attributes and contributing sub-attributes throughout the decision making process. Whether the proposed attributes are leveraged independently or concurrently within a process, extensive research re-iterates the need for careful consideration.

It is proposed that for models (e.g., CDSEM, CRE) that require the elicitation of decision attributes for the identification and selection of COTS applications, should be modified to utilize the proposed nine decision attributes as foundation for which to work.

In addition, literature research and analysis of the survey data re-iterate that despite popular belief, licensing cost is not the most significant decision attribute. Identifying new version capabilities and contribution to the objectives of the organization or program remain high on executive decision maker's priorities. Therefore, focusing system engineering activities beyond cost analysis is recommended.

Bibliography

1. Hennessy, E., *COTS Supportability Across the Program Lifecycle—A Vendor's View of COTS' Evolution in the DoD Market*. 2000, TechOnLine.com.
2. Merriam-Webster. *Webster Dictionary (online)*. [cited 2004-2005; Available from: http://www.webster.com.
3. Brownsword, L., T. Oberndorf, and C. Sledge, *Developing New Processes for COTS-Based Systems.* IEEE Software, 2000(July/August 2000): p. 48-55.

4. Salter, C.P. and D.M. Buede, *Lifecycle Based Method for the Acquisition of Commerical-off-the-shelf (COTS) Technology to Support Organizational Processes.* Systems Engineering, 2001. **4**(4).
5. Torchiano, M., et al. *COTS Products Characterization*. in *14th International Conference on Software Engineering and Knowledge Engineering*. 2002.
6. Long, J.E. *COTS: What You Get (In Addition to the Potential Development Savings)*. in *INCOSE 2000 Mid Atlantic Regional Conference*. 2000. Minneapolis, Minnesota.
7. Boehm, B. and C. Abts, *COTS Integration: Plug & Pray?* IEEE, 1999.
8. Tran, V. and D.-B. Liu. *A Risk-mitigating Model for the Development of Reliable and Maintainable Large-scale Commercial-off-the-Shelf Integrated Software Systems*. in *Reliability and Maintainability Symposium*. 1997.
9. Boehm, D.B., *COCOTS (COnstructive COTS) Software Integration Cost Model: An Overview*. 1998.
10. Voldase, I.S., *Satellite Ground System Cost Benefit Trade-Offs of Commercial Off-The-Shelf Items Using Parametric Cost Models*. 1999: El Segundo, California. p. 24.
11. Medin, H. and A. Forsgren, *Outsourcing, COTS for the new millennium*: Västerås, Sweden.
12. Shaffer, G. and G. McPherson, *FAA COTS Risk Mitigation Guide: Practical Methods For Effective COTS Acquisition and Life Cycle Support*. 2002, FAA. p. 113.
13. Seaman, C., *COTS-based Development Process in the FDD, Study Brief #6*. 1997, NASA Software Engineering Laboratory.
14. Jha, S., B. Miller, and T. Reps, *Vulnerability and Information Flow Analysis of COTS*. 2002, University of Wisconsin: Madison, WI.
15. Owens, K., S. Eslinger, and G. Chaudhri, *Managing COTS Integration for High Integrity Systems: How far have we come? Problems and solutions in 2003*. 2003, Aerospace Corporation.
16. Halpin, D., *Procurement Strategy: Dragged or Driven?* COTS Journal, 2004. **6**(March).
17. Clapp, J.A. and A.E. Taub, *A Management Guide to Software Maintenance in COTS-based Systems*. 1998, Mitre, Center for Air Force C2 Systems: Bedford, MA. p. 35.
18. Seddon, P.B., V. Graeser, and L.P. Willcocks, *Measuring Organizational IS Effectiveness: An Overview and Update of Senior Management Perspectives.* The DATA BASE for Advances in Information Systems, 2002. **33**(2): p. 11-28.
19. Boehm, D.B., *COCOTS (COnstructive COnstructive COTS) COTS Software Integration Software Integration Cost Model: An Overview.* 1998.
20. Anonymous, *Think Twice Before You Upgrade.* InfoWorld, 1993. **15**(52): p. 1.
21. Gregory, K., *Rx for Upgrade Headaches: Here's how to avoid the frustrations and kinks that accompany new hardware and software purchases.* On Wall Street, 2004: p. 1.
22. Herald, T. and J. Seibel. *Technology Management Forecasting and Predictions for System of Interest Analysis*. in *INCOSE MARC 2004*. 2004. Arlington, Virginia: INCOSE.
23. Nirgudkar, R. *COTS Software/Hardware*. [cited 2000; Available from: file:///C:/DISS/Catalogued/COTS%20Software%26Hardware.pdf.
24. Clapp, J., *Understanding Risks Allieviates COTS-based Systems Woes.* The Edge Perspectives, 2001. **2**(1): p. 3.
25. Florida, S.o., *Modernization of State Government Financial Management Business Practices Study*, S.o. Florida, Editor. 2000.
26. Minkiewicz, A.F., *The Real Cost of COTS*. 2001, PRICE Systems, L.L.C. p. 5.
27. Martín-Albo, J., et al., *CQM: A Software Component Metric Classification Model.* IEEE.
28. Kontio, J., et al., *A COTS Selection Method and Experiences of Its Use.* IEEE Software, 1995(16).
29. McVay, W.H., *Report on IT Spending for FY2002-FY2004*. 2004, United States Government, Office of Management and Budget (OMB/OIRA/IPT). p. 2.
30. Sledge, C. and D. Carney, *Case Study: Evaluting COTS Products for DoD Information Systems*. 1998, Software Engineering Institute: Pittsburg, PA. p. 13.
31. Government, U.S., *Assessing the Risks of Commercial Off-the-Shelf Software*. 1999, Information Technology Resource Board.
32. Maybury, D.M.T., *What rots about COTS*. 1999, Mitre.
33. Keller, E., *It's Easy to Pick Applications.NOT!* Manufacturing Systems, 1994. **12**(1): p. 50.
34. Miller, S.M., *Assets & Liabilities: Accounting Software that's right for your Business.* Office Solutions, 2000. **17**(3): p. 39-41.
35. Farbey, B., F. Land, and D. Target, *How to Assess your IT Investment*. 1993: Butterworth-Heinemann, Ltd. 144.
36. Taub, A., *COTS Software Licensing Accompanied by Trepidation.* The Edge Perspectives, 2001. **2**(1): p. 6-7.

37. Baum, D., *New Technology Burnout? Here's how to cope.* Datamation, 1995. **41**(21): p. 45-48.
38. Furger, R., *The Treacherous Upgrade Path: Upgrading to the latest version of a software package is a costly and time-consuming process with no guarantee of better performance, increased productivity or user compliance.* InfoWorld, 1988. **10**(9): p. 43.
39. Kirk, D., *Software Upgrades are Driving Managers Mad.* InfoWorld, 1993. **15**(29): p. 58.
40. Salter, C.P., *A Decision Framework for the Selection of COTS Technology for Organizational Processes*, in *Department of Systems Engineering and Operations Research*. 2001, George Mason University: Fairfax.