ABSTRACT

Managing software and its various components such as maintenance, upgrades, and patches, has become a requirement to secure one’s place in a competitive market. The rapid change in engineering software, especially for the Aerospace industry, highlights the importance of software management. Negligence of any maintenance component can put the organization into risk. Therefore, standardizing processes are necessary to avoid not only these risks, but also minimize costs and ensure employee productivity. Currently, an Aerospace company upgrades most of their engineering software on an ad-hoc basis. This means that an upgrade is only done when an end-user requests it. The study will therefore examine the current operations and offer a strategic reengineered process. The objective of such changes is to ultimately streamline the distribution of engineering software upgrades proactively throughout the organization.
INTRODUCTION

Upgrading sophisticated engineering software is not as easy as clicking “Yes Install” after a pop-up message indicates that there is a new version of Internet Explorer. Managing software applications is significantly more complex particularly in the Aerospace industry. As a result, an entire department, the Information Technology (IT) unit, is dedicated to coordinating and supporting the various system solutions. For instance, the IT department must ensure that all legal matters related to the software are sufficiently up to date, including the annual maintenance payment, license usage rights, etc. Next, one must consider the number of users in need of a particular application, and consequently weigh its costs and benefits. If the needs justify the expenditure, then the IT department will proceed with its implementation, otherwise they will contemplate alternatives. Such analysis is carefully examined as certain sophisticated software applications could cost up to $20,000 annually for just one license. While the cost is considerable, aerospace manufacturing successes are due in part to the cutting edge technology in use. Indeed, it is through these very same enhanced engineering tools and software solutions that engineer end-users are capable of designing and creating 3D prototypes, as well as generating simulations with which to test their ideas prior to physical production.

Currently, an Aerospace company upgrades most of their engineering software on an ad-hoc basis. This means that an upgrade is only done when an end-user requests it. Consequently the software upgrade distribution has no release frequency plan, nor clearly defined sequence of operations. This results in lengthy delivery cycle time and user dissatisfaction. According to Stackpole and Harrison (2008), “There is more than enough technology available to secure and manage systems. The problem is process, or more to the point, the lack of process” (p. 2). This study will therefore examine the current operations and offer a strategic reengineered process. The objective of such changes is to ultimately streamline the distribution of engineering software upgrades proactively throughout the organization.

In the initial section, the difference between a software update and an upgrade will be explained, as well as some definitions of ad-hoc and standard processes. The following section will reflect the data obtained from various stakeholder interviews and their respective experiences. Lastly, the study will develop and propose a strategic reengineered process based upon the conclusions determined from these observed assessments.

The final proposed process ascertained from this research consists of six overarching individual operational activities. While all six occur at least once annually, three of these processes reoccur with greater frequency according to corresponding vendor release. This concluded process is by no means absolute; rather it is to be considered as an initial step towards a larger more permanent goal.

The objective of the reengineered process is as follows:

A. Standardize, and Streamline the Current Upgrade Process
B. Reduce Cycle Time
C. Create Strategy for Frequent Scheduled Update
D. Develop User Testing and Integration Phase

BACKGROUND

There are three types of software evolutions, patches, updates and upgrades. While the first two are often considered pretty much the same and thus interchangeable (Stackpole and Hanrion, 2008, p. 13), upgrades differ significantly. An upgrade is a major change that occurs on the software with less frequency, perhaps once a year, if not more. For instance, a move from Adobe Acrobat 9 to Adobe Acrobat 10 is considered an upgrade. On the other hand, the minor changes that occur within