

Managing Risks Related to Unsupported Software

By Warren Axelrod

Introduction

Commercial and homegrown computer software products age and invariably, at some point in time, are no longer supported and maintained by their builders or service agents. Because of the competitive and dynamic nature of the IT product marketplace, manufacturers are constantly bringing out new versions of existing products and entirely new substitute products. At the same time, vendors make their earlier products obsolete by announcing a date, usually in the not-too-distant future, after which they will no longer support earlier versions of the software. Vendors may also make earlier versions less attractive by increasing the costs of maintenance and support, for example, in order to encourage purchase of the new version and generate increased revenues.

Internally developed custom software is often more difficult to discontinue than commercial products and is more likely to become unsupported because of attrition of internal staff or consultants familiar with the system.

Whatever the reason for a product losing support, the fact that there is no one to make functional changes, maintain compatibility with other software or hardware, fix bugs or create and apply patches, particularly security-related patches, represents a significant and increasing risk to customers and end-users. If support for the product can be found or created, the cost is much greater than that of supporting an equivalent current product.

Objective

We wish to identify and quantify, where feasible, the risks and costs that derive from unsupported software and to illustrate how one might decide whether to continue using the unsupported product or replace it. We will:

- ▲ Define the scope of software types and situations to be addressed
- ▲ Look at reasons why the products might become unsupported
- ▲ Identify risks, costs and benefits related to retaining or replacing unsupported products, and
- ▲ Recommend an approach to determining which products should be retained or replaced.

Scope

We shall consider the following categories of software, which may be subject to loss of support:

- ▲ Commercial Off-The-Shelf (COTS) software
 - ▼ Supported by manufacturer
 - ▼ Supported by another third party (e.g., service organization)
- ▲ Externally-developed custom software
 - ▼ Supported by developer company
 - ▼ Supported by another third party
 - ▼ Supported internally
- ▲ Hybrid software (developed by both internal and external staff)
 - ▼ Supported by internal staff
 - ▼ Supported by a third party
 - ▼ Supported by combination of internal and external staff
- ▲ Internally-developed software
 - ▼ Supported by internal staff
 - ▼ Supported by third party

Reasons for Obsolescence

There are a number of reasons why software may no longer be supported, including:

- ▲ The vendor abandons the software altogether in line with its business strategy
- ▲ The vendor goes out of business
- ▲ The vendor is acquired by another company with competing software in the same space and decides to discontinue one of the products
- ▲ The vendor replaces the software with a new version or with entirely new software
- ▲ For internally supported software, the company changes its business requirements and no longer needs or wants to use the software
- ▲ For internally supported software, the subject matter experts needed to support the software are no longer available, due to attrition, dismissal, etc.

In general, a vendor will upgrade or replace software because of market pressure to do so, or because another company with a competitive product acquired the vendor, or if the vendor feels that new features and capabilities will provide a competitive advantage.

The reasons for discontinuing internally developed or hybrid software are similar to those of COTS software except that it is usually the business unit, rather than IT, which provides the impetus to change or replace internally-developed software. It is less usual for unavailability or inadequacy of staff to force an upgrade, but it can and does happen.

In addition, the obsolescence of, and resulting lack of support for, software upon which another specific piece of software depends, may force the hand of a customer to replace current COTS products or revise or replace custom software. A good example is when an operating system (OS) is discontinued and no longer supported. This means that the company might have to replace the applications running on that OS or significantly modify them in order to run on the new, supported OS version.

If a company goes out of the business of providing the software, then the software will clearly become unsupported, unless a third party is able to support it. Alternatively, the customer can directly hire personnel who formerly worked for the defunct company and who are familiar with the software. In the latter case, the company must arrange for escrowing the latest version of the source code and must ensure there is enough documentation for someone conversant in the technology to maintain and support the software. Otherwise, the strategy will be unproductive.

For internally developed or hybrid software, where the support is provided in-house or via a third party, the customer generally has greater control as to when the software will be retired.

Once the vendor announces withdrawal of support for software and the whole marketplace moves to the next generation, there will be attrition in the expertise available to support former versions of the software. This is because staffers either learn about software that is more current or change jobs, and new staff does not want to be trained in older technology. Even if the vendor or customer wanted to train new staff, there would be a question as to whether suitable, knowledgeable trainers were still available, as there is a diminution over time of trainers for obsolete technologies.

Risks and Costs Related to Loss of Support

Whatever the reason for software becoming unsupported, the very fact that adequate support no longer exists presents the software customer and end-user with a number of considerable risks. These include the following:

Operational Failure—If software ceases to operate due to an intrinsic bug, the lack of support might mean that the software is not repairable and it becomes unusable. In addition, it is not usable in a new situation, or it is only fixable at high cost over a protracted period.

Loss of Flexibility—Not being able to make changes to unsupported software might preclude taking advantage of new functionality that would be normally provided via product support. This reduces the flexibility and usefulness of the software.

Legal and Regulatory Requirements—The inability to change unsupported software can create real problems when changes in laws and regulations mandate that the software be brought into compliance.

Business Liability—Not changing the software to comply with laws and regulations increases the risk to the business.

Opportunity Costs—Reduced flexibility will sometimes result in lost business opportunities, either through the failure to attract new customers or the loss of existing customers to the competition.

	Year 1	Year 2	Year 3	Year 4	Totals
Unsupported Product					
Conversion	\$ -	\$ -	\$ -	\$ -	\$ -
Maintenance	\$ 100,000	\$ 150,000	\$ 250,000	\$ 400,000	\$ 900,000
Losses from Breaches	\$ -	\$ 300,000	\$ 500,000	\$ 800,000	\$ 1,600,000
Regulatory fines	\$ -	\$ -	\$ 200,000	\$ 200,000	\$ 400,000
SLA Penalties	\$ -	\$ 100,000	\$ 200,000	\$ 400,000	\$ 700,000
Loss of Certification	\$ -	\$ 300,000	\$ 500,000	\$ 800,000	\$ 1,600,000
Totals	\$ 100,000	\$ 850,000	\$ 1,650,000	\$ 2,600,000	\$ 5,200,000
Supported Product					
Conversion	\$ 2,000,000	\$ -	\$ -	\$ -	\$ 2,000,000
Maintenance	\$ -	\$ 100,000	\$ 110,000	\$ 125,000	\$ 335,000
Losses from Breaches	\$ -	\$ 100,000	\$ 150,000	\$ 200,000	\$ 450,000
Regulatory fines	\$ -	\$ -	\$ -	\$ -	\$ -
SLA Penalties	\$ -	\$ 50,000	\$ 50,000	\$ 50,000	\$ 150,000
Loss of Certification	\$ -	\$ -	\$ -	\$ -	\$ -
Totals	\$ 2,000,000	\$ 250,000	\$ 310,000	\$ 375,000	\$ 2,935,000
Supported v Unsupported	\$ (1,900,000)	\$ 600,000	\$ 1,340,000	\$ 2,225,000	\$ 2,265,000
Net Present Value					\$ 1,722,749

Figure 1: Comparison of Costs and Benefits of Unsupported vs. Supported Software

Security Vulnerabilities and Incidents—If a security vulnerability is discovered, or if an attack is directed at unsupported software, customers will not be able to get patches or may have to try to somehow get them at a high cost. Otherwise, a customer might have to implement other costly mitigation measures to reduce this risk.

Integration with Other Systems—If software, which is still supported, interfaces with, or depends upon, other software, which is not supported, then the company is subject to all the risks affecting the unsupported software because of this dependency.

Rapidly Rising Costs of Support—As software ages, and available support diminishes, the cost of remaining support tends to increase very rapidly over time such that it will likely far exceed the cost of conversion to a supported product.

Additional Load on Help Desk and Technical Support—The likelihood that tagging, unsupported software increases the number of intrinsic problems results in more Help Desk and Technical Support calls by end-users having problems with the software. This is often accompanied by a decreasing ability of support personnel to resolve issues.

Breach of Contract—In some agreements between customers and providers, there is often a requirement to upgrade software to the latest supported version within a specified period of time.

Service Level Agreements—Even if there is not a contractual requirement to maintain software current, problems specific to unsupported software can increase the probability that service levels, as stated in Service Level Agreements (SLAs), will not be achieved. This might lead to financial penalties, loss of reputation, degradation in customer relations, lower morale, etc.

Third Party Security Certifications—There is a chance that a third party, such as an auditing firm or security consultant, will discover obsolete software in the course of the security assessment. Consequently, the assessor will not grant the desired certification or provide a positive report. The lack or loss of certification may result in a breach of contract or a loss of existing and potential business. It is also likely to result in undesirable consequences for internal staff. With executive management having to sign off on such matters as the protection of customer data, the loss of such certification could have severely negative legal and regulatory implications.

Decision to Retain or Discontinue Product

There is clearly a tradeoff between keeping obsolete software in place and sustaining all the risks and costs related to running unsupported products. This tradeoff includes the risk of incurring the potentially high cost of replacing the software with currently supported products.

It is necessary to conduct a risk analysis to determine whether to continue using or disband unsupported software. The factors to be included in the risk analysis are:

Criticality of the successful operation of the software to the business—If the software supports a critical business process or affects the ability to comply with SLAs, laws, and regulations, it makes a difference. This concept is opposite of software that is “nice to have.”

Relative cost of maintaining current software versus upgrading—Some software is intrinsically simple to upgrade and has little impact on other systems. Other software is integrated into the infrastructure tightly and requires major rework to change to a new version or different product.

Analysis

Figure 1 provides a representative analysis of two alternatives, namely, retaining unsupported software for a further four years, versus replacing the software with a new version. The replacement effort requires various hardware upgrades and application changes. It should be noted that the costs of replacing and maintaining software can usually be determined fairly accurately. However, other costs, such as the cost of a successful attack on the software, are only estimates based on broad probabilities and order-of-magnitude numbers that have been reported in the press. The time value of money is assumed to be five percent throughout the four-year horizon and it is assumed that the costs are all incurred at the end of each year.


As can be seen from figure 1, under the given assumptions, the \$2 million up-front investment for moving to the new software saves almost an estimated \$2.265 million over four years. Even when discounted, the net present value (NPV) is \$1.723 million.

The figure illustrates that the loss of use of critical software can lead to considerable fines, penalties and loss of business.

Recommendations

The recommendation is to perform an analysis similar to that shown in figure 1 and upgrade to current supported software if the NPV is greater than zero. If the NPV is negative or close to zero, then it is reasonable to stick with the older software, particularly if the resources, which would be applied to the conversion, can be applied to projects with a better return.

If it is decided to retain the unsupported software, it is important to review the situation and redo the analysis at least annually or when any major change occurs.

For supported software, it is also important to reevaluate whether to retain or replace the software every time there is a change in status, such as the introduction of more fully featured and/or less-costly solutions. 

C. Warren Axelrod, PhD, CISSP, CISM is director, global information security, for Pershing LLC, a BNY Securities Group Co. He is the author of the book "Outsourcing Information Security," which is to be published by Artech House this month. He is chair of the GAISP Information Security Policy Principles Working Group and a member of Editorial Advisory Board of "The ISSA Journal." He can be reached at caxelrod@pershing.com.