

Getting to Understand Some of the Operations and Maintenance Drivers of Cost

Jonathan W. Powell, CGFM, PMP, Security+

Abstract. The overall theme for this issue is Operations and Maintenance (O&M). This article will look at specific cost drivers, including licensing, design and documentation, replenishment planning, data collection, and impacts of the operational environment. It will then look at tools to reduce cost and improve availability. It will close with a look at the younger generation and discuss whether a tour of duty or two in O&M would help their overall career growth and effectiveness as engineers.

O&M comprises about 80 percent of system cost, yet many analyses during system acquisition ignore this phase. So if one is to look at O&M, some questions he or she might ask are:

How is the architecture influenced by license cost?

Architecture should be informed by licensing cost, in addition to other costs (such as the cost of maintenance). Over the last couple of decades, appropriate emphasis has been given to this category of cost. This was the period when Linux and the whole open-source construct came to the fore as a means to help tamp down escalating software development costs. Licensing costs pervade every aspect of architecture, from the number of tiers used for a web-based system to the makeup of the databases to transaction processes to screen displays. The rubric with respect to architecture and maintenance costs is fairly straightforward:

1. The more complicated it is, the higher the licensing costs.
2. Open source is cheaper than its alternatives.

In the federal government context, software development is most often constrained by budget. This means there's a ceiling on design costs. Since the makeup of the architecture directly impacts design costs, and licensing is an outcome of the architecture, these costs are subject to budgetary pressures. Therefore, if you're under extreme budgetary pressure, you could seek to simplify the architecture (for example, develop a single tier rather than multiple tiers).

The point is this: The right answer should always be to make the architecture as simple as possible for its intended purpose. But this is challenging and contradicts contractor self-interest, which seeks to maximize revenue by driving up complexity and, hence, costs.

Most often, when budget is an issue, the easy answer for a project manager is to default to open source components for

the architecture. Contractors invariably will argue against open source, saying it's not as robust or cannot be used for mission applications, but the government should push back. These attacks are often empty, used in an attempt to drive up revenue. They should be beaten back by the program management team.

In a perfect world, you would lay out all of the architectural options available to meet a particular set of design objectives and provide a cost-benefit assessment (CBA) for each of these options in order to apprise decision-makers of all the options and tradeoffs on the table. The CBA should be nuanced to fully account for impacts on the mission. For example, rather than using a straight tally of benefits versus costs, the benefits could be given weighted percentages of the design objectives met, ensuring all key performance parameters (KPPs) are satisfied. If you include the payback period, you can derive a matrix as shown in the following figure:

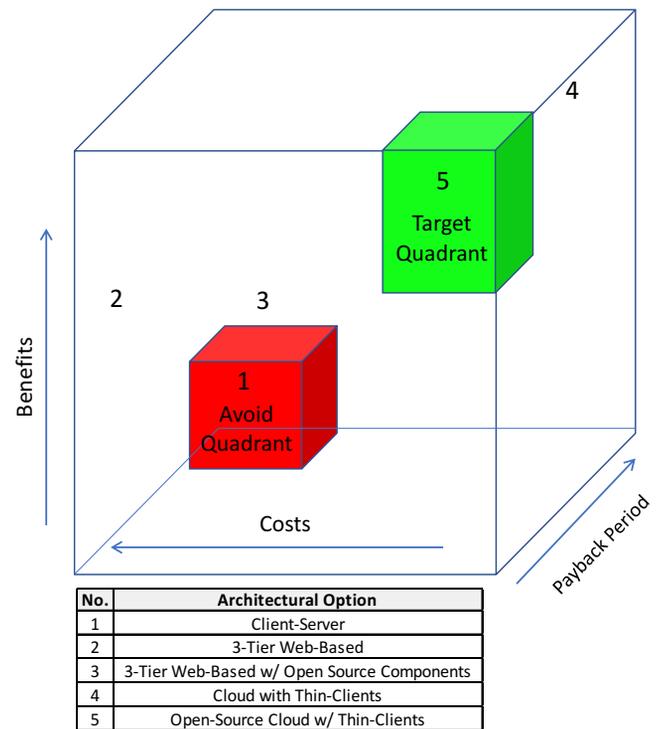


Figure 1. Example Cost-Benefit Assessment Matrix Including Payback Period Analysis

But this is rarely done, and when it is, it's often glossed over or poorly performed. Even where a CBA is required, it's often just reuse of previously submitted versions and rarely brings innovation and critical thinking to the task of sorting through the art of the possible. In the current era, when more attention is being paid to getting the requirement right the first time so the acquisition design and development are correct coming out of the chute, and when programs are positioned for success, more emphasis should be paid to this area. Or, for less complex procurements, perhaps providing a scaled-down version of the analysis for decision-makers should be considered.

How does software design and documentation influence maintenance cost?

Software design influences maintenance cost because it determines how the system will be built. A complicated design rarely leads to an easily maintainable system. In fact, overcomplicating system design is a tactic used by software developers to help drive up cost (and hence revenue) while also locking in a competitive advantage. A company can make the design of a system more complicated than it needs to be, or make it so complex that there's a learning curve built into the system such that anyone performing maintenance during O&M needs a significant period of time to learn about the system before they're productive, driving up O&M costs dramatically. I experienced this on the army software program I led for eight years. Prior to this, the program had been in existence for seven years without software being fielded. That had been enough to get the incumbent fired, but enough of a complicated design had been implemented by then. Not only were they able to lock in 40 percent work share as a subcontractor on the follow-on contract because of the inherent knowledge they had of the system's complexities, but they also forever cursed the system to a pattern of higher-than-normal maintenance costs because the byzantine architecture created could only be divined by a large number of senior and seasoned C# and PL/SQL development specialists, who were much more expensive than their more junior-level counterparts.

Most of the cost in typical systems is O&M, and therefore there is a need for documentation to help ensure the O&M is performed as effectively and efficiently as possible. This is especially true if the customer seeks to enhance their flexibility and enable vendors to compete for the maintenance work. A solid technical documentation package can help with this and help a new vendor to come up to speed quicker on the O&M of a system rather than coming in cold, thus helping reduce transition costs. The problem is that software development invariably falls behind schedule — refer to the Standish reports for data on this. And when development falls behind, documentation and training are usually the first items to go in an effort to recover schedule. The corollary to this is when documentation is overemphasized and too much investment is made in documentation. No documentation package is perfect and can fully inform staff members of all the ins and outs of performing system O&M, especially on complicated systems. In most cases, if the documentation can act as a guide and a useful reference, that will suffice. Investing in documentation beyond this is pointless as it will end up sitting on the shelf unused. The other aspect of documentation that should be carefully watched is documentation as part of O&M. How time-consuming is the paperwork regime for configuration control? Implementing bug fixes and patches? An onerous paperwork regime can have a significant adverse impact on the cost of maintaining a system, especially if every little change needs reams of paperwork to get approved and authorized. Let program managers beware.

How does the replenishment plan influence hardware cost?

Again, it depends on the mission and budgets, but in general, a replenishment plan with frequent updates will lead to increased hardware costs. For example, if the application is mission-critical and data processing intensive, the replenishment plan might call for frequent 'tech refreshes' to take advantage of increasing price performance and gains in processing power, as well as software updates designed to take advantage of these gains. However, the converse can be true as well — legacy hardware can be left in place for years without a replenishment plan and can function just fine if it meets the needs of the mission. In these cases, when the time comes to perform a major refresh of this system, developers often choose to start fresh and replace the system outright rather than attempt to upgrade the legacy infrastructure with new hardware components.

How does data collected during operations influence future service level agreements (SLAs), and sparing?

In theory, it should enhance future SLAs because the underlying detail should be available to objectively evaluate performance and implement both "carrots" and "sticks" as appropriate. Also, increased data should improve the effectiveness and efficiency of sparing operations. Assuming the right sort of data is collected, it could be analyzed to yield information that reduces duplication and helps get the right part to the right place faster, all of which leads to reduced cost. However, the underappreciated part of this equation is the cost of collecting data. The program team should assess this carefully. Collecting data for data's sake is not only costly but of minimal value. To ascertain what data needs to be collected and how this information will be used is key. A cost-benefit analysis should be done to show how the data will be transformed into useful, actionable information such that the relative benefits of the increased data collection outweigh the costs.

How do we account for the operational environment in estimating the O&M cost?

This is the whole area dealing with unknowns. You can plan all you want with respect to operations, but at the end of the day, stuff happens, and it's often stuff that hasn't been encountered before. So, there are two parts to accounting for the operational environment in estimating O&M costs. First, plan, plan and plan again. In the submarine force, we use lessons learned. In the Army, it's AARs (after action reviews). Whatever the methodology, take the collective body of evidence and build out planning based on situations that have occurred. These should be prioritized based on probability and impact, and response plans should be built and budgeted for so that the O&M team can rely on them to execute when the time comes. The impact to O&M cost for these plans includes the cost of recurring training on the plans so personnel are familiar with them and they remain fresh and can be executed effectively when needed, as well as any surge costs the plans provide for — for example, emergency personnel and equipment brought in to deal with certain scenarios. As far as the unknowns go, these should be budgeted for as well. This fund is often referred to as "management

reserve,” and the leadership should put into place some buffer they can call upon if unexpected situations are encountered in the operational environment the O&M team must respond to.

What tools should be used during operations to reduce cost, improve availability, or both?

The most important tool is a corporate culture of free-flowing communications. It's not the higher-ups who are going to know about the issues; it's the operators on the ground who are encountering them. In order to address these issues, appropriate channels need to be in place so information can be readily captured. A system must be in place so issues can be quickly adjudicated and addressed. Emphasis should be placed on ensuring the communications channels are not overly burdensome (such as having to fill out long forms to document the points). These communications channels can have the added benefit of capturing staff members' good ideas and innovations. The best way to improve availability and reduce cost are finding ways to most effectively and efficiently capture the good ideas and suggestions of those performing line operations on a daily basis. The challenge for leadership is establishing an open environment where these innovations can be shared and brought forward. The hardware or software tools for capturing these ideas are not important, as long as they make it easy for the operator to convey the information. For example, it could be as simple as a Sharepoint portal where operators can enter the information — either publicly or anonymously — and begin a workflow to get the ideas known and visible to leadership. Remember, the most important thing here is the corporate commitment to capturing good ideas and then acting upon them.

Should bright young engineers be assigned to O&M for educational purposes?

Absolutely. It's vital that young engineers understand the full life cycle of products and software. It's especially important because they'll get to see how decisions made “way back in design and development” impact the lives of maintainers during O&M. This will give them an appreciation for the effects of these decisions. It will also encourage them to be proactive in addressing downstream maintenance impacts when they lead and participate in design and development later.

ABOUT THE AUTHOR



Jonathan W. Powell, CGFM, PMP, Security+, is an adjunct professor at UMBC. A former submarine officer, Mr. Powell has led complex engagements for military, federal and intelligence agencies. His articles have also been published in “PM Network” and “Contract Management.” Mr. Powell serves on the board of the Montgomery County Revenue Authority and is a member of the State of Maryland's Cybersecurity Council. A graduate of the FBI's Citizens Academy, he holds a B.S. from USNA and an MBA from the University of Maryland.

**13205 Dodie Dr.
Gaithersburg, MD 20878
240-418-8588
jopowell@umbc.edu
201-248-4332**

