

A Balanced Approach to Systems Engineering

6/13/2011

Abstract: A balanced approach to systems engineering project management will help navigate difficult and seemingly intractable decisions. A project manager's focus should remain on business needs, with technology and tools as supporting elements, having a limited and identified lifespan. "Agile" principles should be considered for technology selection and product maintenance, as well as for software development.

Key Words: Systems Engineering, Agile, Portfolio Management, Trade-off Analysis.

Systems engineering management can be a complex endeavor, especially for large programs, as are often found in the Washington, DC Metropolitan area. Certain programs, for example building a space shuttle, will require a high degree of precision, while other projects, such as prototyping a new internal Sharepoint site, may be less precise, even if the information the site will contain will be just as important to the nation's interests. Many technical initiatives also may have no clear beginning or end; a project may occur in the middle, with a goal to, for example, reverse engineer technical requirements from an existing product, in order to assess the cost of producing a more modern replacement. Also, if a modernization effort is undertaken, the legacy product may well remain in operation, running in parallel to the new product, as a "hot-backup," with no plans for retirement. So, with all of the variation and complexity in systems engineering, how do you know, as a systems engineering project manager, how to approach decisions in process tailoring, technology and tool selection?

The PMBOK leads us to **start by identifying the deliverables**. If the project team's work is in support of building a space shuttle, but the task is to design a new logo, then deep technical rigor will probably not be necessary. The list of final deliverables, which have been directed and approved by the customer, should drive systems engineering management decisions.

Avoid the trap of "gold-plating" technology selections. For example, some systems engineering tools provide a feature to "automatically generate" technical documentation. However, this feature may have low utility. In fact, a case can be made that there is an inverse relationship between the amount of automated technical documentation and a high quality and low risk delivery. The heart of the issue lies in "what" vs. "how." Business requirements should describe *what* is needed by the users and business organization, and technical documentation should describe *how* a solution satisfies what is needed. Automated technical documentation

typically produces a long and monotone list of *what* is included in the technical solution.

Technical teams often make a mistake in assuming that this list obviates the need to describe *how* the technical product was developed. As a result, automated technical documentation may *increase* cost and risk. In time, there may no longer be any personnel available to explain *how* the solution was produced and, even if a limited number of people are available, when the information is needed, it may be a challenge to remember the details of a project from well in the past.

It can be to the customer's advantage to generate the technical *what* information, if it is needed and when it is needed. The information will exactly match what was delivered, which is an advantage in technical precision. This information may then be discarded, as point-in-time technical data, in support of an analysis or defect fix, as opposed to an additional and lengthy artifact, which may require unnecessary and expensive maintenance.

While "round-trip" systems engineering suites may provide great capabilities in visual modeling and automated testing, they tend to include very limited and insufficient support for project management schedule and cost analysis. MS Project is an excellent project management tool, as is MS Visio for architecture and design, so long as the information is properly version and access controlled, using tools like CVS. Most CMMI Level 3 certified programs use MS Office. A fully integrated and single brand set of systems engineering software is not necessary for process maturity or a high quality and low risk delivery.

The risks in "gold plating" process, technology and tools are the same as with business scope. Additional technical scope comes with a cost: acquiring and supporting licenses from an initial purchase through maintenance, personnel training costs, and additional and often hard to find specialized resources, to install and configure the technical products and tools. It's important

to look towards the future in the field of technology, but given the industry project failure rate, coupled with often changing customer needs, it can be unwise, and even dangerous, to buy more than is needed, and to “engineer for the future,” without proper controls. The best solution is typically to **use only as much process, technology, and tooling as is necessary to deliver a solution to meet the customer’s needs.**

There is also a risk in striving for a single, monolithic, and point-in-time technology or suite, which will inevitably come with technical limitations, may quickly become dated, and possibly lock the customer into one vendor. **Avoid the temptation to search for a systems engineering “holy-grail.”** An “Agile” approach to technology and tool trade-off analysis may be more successful.

Even software maintenance can be “Agile.” In very tightly controlled environments, such as financial institutions operating under Sarbanes Oxley requirements, after a product has been delivered into production, a tool like CVS may be successfully used to store and access both technical and business artifacts, such as “sign-off” documentation. If a systems engineering suite has been in place, it generally isn’t necessary to continue to provide licenses for all team members. One member of the team can be appointed to act as a secretary, to keep the information contained in the suite up to date, until its contents are exported and archived, and the suite retired.

The focus of a systems engineering project manager should remain on delivering to the business needs of the customer, with a good ROI, and against known and reasonable points of time in the future. **Make practical cost management decisions.** For example, software licenses should only be purchased for users who need them, and they should be turned in as soon as possible. Additionally, technical teams should take advantage of any available floating licenses,

when possible, and where it provides a cost advantage over fixed licenses to particular individuals or machines. The systems engineering project manager should keep in mind that there is a trade-off between acquiring licenses and hiring personnel. A software license, when not in use, will not contribute to a delivery, so a deliberate effort should be made to consistently drive down licensing costs, in support of a lower cost delivery, and a greater opportunity to hire additional personnel as needed (See Figure 1).

The maturity of a technology should also be taken into account, during a technology trade-off analysis. A new technology typically starts with a low ROI, which steadily increases over time, based on the technology's success, and which eventually lowers again, as the technology shifts out of favor, and becomes obsolete (See Figure 2). For example, COBOL was initially a new and sophisticated programming language, which became commonplace and very practical to use in the 1980s, but which now requires hard to find and often expensive resources. **Consider the time horizon of the customer's business operations**, and correlate lines of business to technology selection decisions, based on technology maturity. Keep an eye on maintenance, retirement, and disposal, as these phases typically consume 80% of the total cost of a technical solution.

A periodic review of technical inventory and software licensing, across the customer's organization, will aid in making these decisions. The importance of this activity cannot be overstated. Even a large and well funded organization can face a diminishing opportunity to invest in new and necessary systems, in order to adapt to business changes and new business opportunities, as legacy systems' maintenance costs increase, technologies are retired by vendors, and maintenance personnels' business and technical skills diverge over time, while their

knowledge fades. Maintenance can overwhelm an organization's technical budget. **Consider an "Agile" approach to IT portfolio management.**

A balanced approach to systems engineering project management will help navigate difficult and seemingly intractable decisions. A project manager's focus should remain on business needs, with technology and tools as supporting elements, having a limited and identified lifespan. "Agile" principles should be considered for technology selection and product maintenance, as well as for software development.

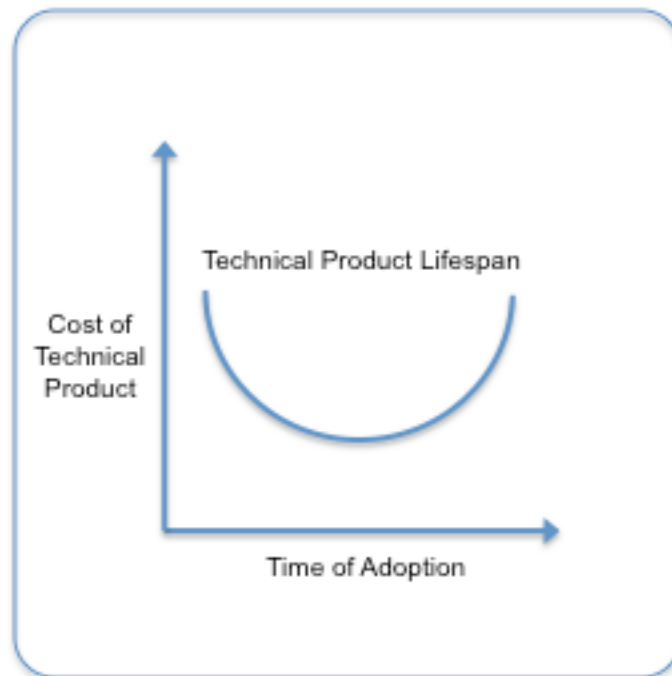


Figure 1- Cost of a Technical Product at Time Adopted Within the Product's Lifespan

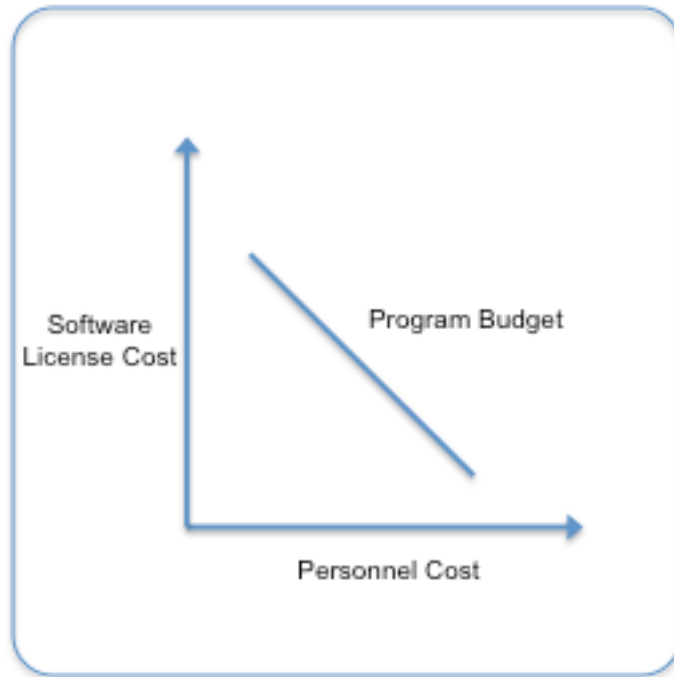


Figure 2- Budget Trade-off Between Personnel and Software Licenses

Ryan L. Maxwell, PMP

MTS

PO Box 9410

Washington, DC 20016

ryanmaxwell@iS3cure.com



About the Author: Mr. Maxwell is a Senior Consultant with MTS, a Virginia based technical consulting firm. He earned a Bachelor of Arts from Johns Hopkins University, has 13 years of systems engineering experience, is a PMI certified project manager (PMP) and member of the Washington, DC Chapter, a Sun Certified Enterprise Architect (SCEA), and an IBM certified process specialist and requirements manager (RMUC). Mr. Maxwell is a former

employee of IBM, where he held several positions including the Rational Technical Excellence Program (TEP) Enablement Leader. He is the author of “Improving Project Agility,” enjoys speaking about systems engineering, working on technical projects, and lives with his wife in California.