

Custom vs. Off-the-shelf

Today's businesses can't afford to waste time or money buying and implementing the same solutions that their competitors are using. The resources committed to a project should move the company forward and create solutions that put them at the head of the pack. Software must optimize business processes, facilitate workflow, provide needed tools, and ensure security. It needs to be cost effective, and require minimal effort and expense to maintain. It has to support innovation that will drive the growth of the company.

What's the best way to get that software? Is buying a commercial product off the shelf (COTS) the best bet, or is custom crafting software to exact specifications going to provide the best value? The answer lies in establishing a complete picture of what a business requires from the software, as well as the total cost of ownership (TCO) associated with all options.

Understand the problem space

The first step is a requirements analysis, resulting in a document that formalizes the scope and priorities of the project. It should include:

- The needs and "wish lists" of all stakeholders
- Guidelines for balancing IT needs and business needs
- Prioritization of cost, security, efficiency, usability and interoperability
- Identification of business processes that must be supported by the software
- Identification of business processes that may be modified, or should be updated
- Evaluation of the current software environment
- Identification of possible changes in the business environment due to changes in the market, and the company's strategic plan.

The resulting requirements document provides a picture of the current situation and the desired outcome, as well as the information needed to evaluate the fit of any COTS products under consideration. If a custom product is chosen, the requirements document will guide the design and management of the project and help compute the TCO of the project.

Magenic

Custom solutions that fit. Guaranteed.

Custom vs. Off-the-shelf

Know the TCO of all options

The TCO of COTS products will often overpower the ROI. Although it is often the case that COTS products have a smaller up-front investment when compared to custom software, the purchase price is only the beginning.

The fit of a COTS product is critical. If the product does not fit business needs, it isn't a solution. A product that does not fit the IT environment will result in additional costs and problems. If the product and vendor are not reliable, the resulting inefficiencies and costs will add up quickly.

The larger up-front costs in custom software are invested in the design, quality assurance, testing and development of the product. Properly done, these activities dramatically reduce the TCO of custom software, while ensuring that it fits both the business and IT requirements of the owner.

One of the biggest hidden expenses of COTS software is the cost of customization. If a COTS product isn't a perfect fit, the choice is between modifying the business processes and requirements, modifying a COTS product, or building a custom product. All of these solutions have attendant costs.

If business processes provide a competitive advantage, it is not worthwhile to change them.

Modifying business processes or requirements can result in additional expenses which should be computed and added to the TCO of the COTS product:

- Training costs
- Costs of internal communications to promote understanding and acceptance of the changes
- Acquiring additional software to meet dependency requirements of the new product
- Upgrading hardware
- Time and money spent designing and implementing changes

Modifying COTS products is a very expensive option. Cindy Shelton, in her paper "Business Process Reengineering with Commercial Off-the Shelf Software", states: "If we are to achieve the expected gains from purchasing software versus building it ourselves, then for the entire life cycle of the products, we cannot allow any modification." (Shelton, 1)

If any modifications to a COTS product are needed, a custom solution will be better. The general rule of thumb is that as the

number of changes increases, the Total Cost of Ownership (TCO) increases on an exponential curve, as illustrated in Figure 1. In addition to the cost of the initial customization effort, each change increases the chance that future updates will require re-customization and that updates will introduce flaws into the system. The complexity of identifying and fixing the flaws increases with each change. These issues very quickly absorb the ROI in COTS software.

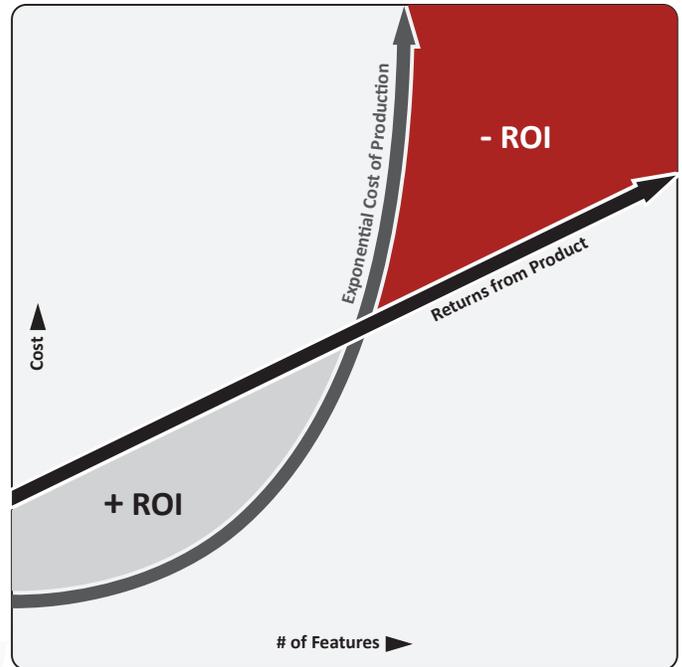


Figure 1

There are additional costs to consider when looking at COTS solutions. These can include:

- Hardware upgrades required by the product
- Supporting software required by the product
- Certified personnel to properly configure product
- Licensing/maintenance costs
- User training requirements
- Additional services/support

An evaluation of any product under consideration should include complete information about these costs, and they should be added to the TCO of the product (Couts, 42-43).

Magenic
Custom solutions that fit. Guaranteed.

Custom vs. Off-the-shelf

Know the product and the vendor

Another consideration is the dependency on the COTS product and its vendor. While it may be attractive to think of COTS products as risk-free, there are both technical and business issues that can affect the long-term stability and security of the product.

A custom solution includes access to and control over the source code. This means the owner can prioritize upgrades and create solutions on a chosen time-line. With a COTS product, the vendor sets the rate of change for upgrades, patches, and security updates. A key priority is obtaining a stable product line from a robust company that can be counted on to maintain the product.

When a vendor goes out of business or stops supporting the product, they leave their customers with a lot of problems. The questions for deciding on a software vendor are much the same as those for deciding on a capital investment; is the vendor company stock performing well? Do bond rating services give it a good rating? Do they have enough customers to indicate long-term stability and high customer satisfaction? How much does the product contribute to the vendor's bottom line? (Couts, 44)

If a customer's rate of change is significantly different from the COTS vendor's, the customer may either find themselves required to upgrade related software more often than they would like, or trapped on an old version that is no longer adequate for their business needs. It's important to know how often upgrades and updates come out for a COTS product, especially if it will be modified or if that software has interdependencies with other systems and products in the operating environment.

Know the security needs

The more widely used a software package is, the more potential pay-off there is in hacking it. COTS software presents unique security risks due to wide use and availability. It is a very attractive target because hacking software represents a substantial investment in time and expertise. The hacker wants to receive the highest pay-off possible for his efforts.

"Again, more tangibly, the major COTS packages typically manage important information and connect to more systems. They are central in both the business sense and the technical sense. Further, the information and experience obtained in one attack can be used again on the same package elsewhere." – Craig Miller, "Security Considerations in Managing COTS Software" (Miller, 2)

COTS products are attractive for hackers because they are accessible. It is easy to download a demo version, or a free limited-time trial version of many products. Once security holes are found and exploited, that information is shared within the hacker community; along with the code needed to perform the hack.

In contrast, a custom system is only as widely distributed as the owner allows. The owner controls access to copies of the program, and the source code. The owner decides who works on it, and what is done with it. If a vulnerability is discovered, the owner must assess the risk and set the time-line to fix it.

COTS products are designed to be generic and function well enough in a variety of environments. Therefore, there is a good chance that a COTS product will lack some specific features that are necessary to get full benefit from any given security infrastructure and will require modification. The choice is then between added security risks or added TCO.

Custom software allows for better testing of the product in the operating environment. Testing a COTS product in the operating environment is usually inefficient and impractical. Because there is no access to the source code, it is not possible to predict where testing efforts should be directed. Testing techniques such as "black box" or "fault injection" are possible, but impractical for large systems like ERP applications. Even in smaller applications, they are less efficient and effective than testing techniques that are possible when the source code is available.

If there are problems with the commercial product, it is not possible to work on the source code directly, and the vendor must be relied on to fix it, or a custom software add-on must be created to address the problem (Miller, 7). This dependency and possible addition to the TCO should be kept in mind when deciding between custom and COTS software.

With a custom solution, the source code is available to the testing team, who can then employ efficient test strategies that will give a much clearer picture of how the new software functions within the software environment.

Magenic
Custom solutions that fit. Guaranteed.

Custom vs. Off-the-shelf

Magenic solutions are the perfect fit

A superior design team (Magenic Studios) with an excellent quality assurance and testing infrastructure (MARQ Test Automation) can begin work in the requirements phase of the project to ensure that the software will integrate seamlessly and safely with the legacy systems; reducing costs, and optimizing the benefits of the new software.

When it comes to balancing cost, usability, stability and security, Magenic is an industry leader in producing innovative custom software solutions that transform businesses and drive growth. A co-sourced delivery model that combines optimal amounts of onsite, onshore, and offshore resources tailors each project's team to that client's specific budgetary, business, and technological needs.

In one case, a customer found that their COTS solution came with burdensome licensing fees. Magenic designed a custom system that met additional business needs and integrated the new solution seamlessly with the client's legacy systems. Over the last four years, Magenic has saved the client millions in licensing fees during an ongoing partnership while providing service, maintenance, and enhancements as needed. (See Case Study #1)

Magenic consultants take the time to get a full picture of the client's current technology needs, business goals, resources and current software environment. Solutions are designed with the customer's priorities in mind.

For example, a medical device manufacturer's financial team needed to replace a risky and unmanageable manual system that tracked and reported on high-value business data with multiple and constantly changing variables. Magenic created a reporting solution that prioritized the business needs over the IT needs. The solution improved the user experience, provided more flexibility and adaptability than the COTS solution, and provided users with easier, faster access to the data that they needed. The reporting features from the legacy solution were preserved, and new reporting capabilities were added. Magenic's assistance in training accelerated what was already a smooth transition to the new technology. (See Case Study #2)

Magenic's proven ability to deliver optimal solutions is the

result of a select talent pool that directs the right talent and skills to every element of the project. Leading experts can be engaged from requirements design through implementation and maintenance. Testing by an internal quality assurance team reduces risk and error throughout the process. A unique architected agile development methodology gives clients constant insight into their projects. The generation of production-ready code and artifacts keeps the project on track and managed more efficiently.

The job doesn't end when the project is "finished". Ongoing support and maintenance supplies an unparalleled level of dependability. For example, when a biomedical resource provider lost their primary internal resource for maintaining their software during a critical transition, they turned to Magenic for support and maintenance. Magenic provided the client with cost certainty and stability (See Case Study #3). As a privately-held company with 16 years of experience and over one thousand satisfied customers, Magenic can be counted on for continued support.

Contact Magenic to engage a team of professionals who create, support and maintain custom solutions that transform businesses and drive growth.

Resources

Burleson, Donald. *Selecting an ERP System: Build or Buy?* TechRepublic, 2001. <http://www.techrepublic.com/article/selecting-an-erp-system-build-or-buy/1040167>

Couts, Todd C. and Gerdes, Patrick F. *Integrating COTS Software: Lessons from a Large Healthcare Organization*. IEEE Computer Society, Washington, D.C. 2009

Miller, Craig. *Security Considerations in Managing COTS Software*. Cigital, Inc. Dulles, VA 2006

Shelton, Cindy. *Business Process Reengineering with Commercial off the Shelf Software*. Defense Acquisition University, Defense A&L Journal, Ft. Belvoir, VA, October 2010

Magenic Case Study #1

<http://magenic.com/Portfolio/GlobalInvitroDiagnosticsLeader.aspx>

Magenic Case Study #2

<http://magenic.com/Portfolio/VascularSolutionsProvider.aspx>

Magenic Case Study #3

<http://magenic.com/Portfolio/SharePointBiomedicalResourceProvider.aspx>

Engage Magenic today online at magenic.com
or by calling our sales line at **877.277.1044**

Magenic[®]
Custom solutions that fit. Guaranteed.