uTip	Guidance from the Functional Sizing Standards Committee	

Using Function Point Analysis (FPA) in Commercial off the Shelf (COTS) Software Acquisitions

uTip # 01 – (version # 1.0 *04/22/2013)*

Author: Daniel B. French		
Reviewers:		
Bonnie S. Brown	Tammy Preuss	Adri Timp
E. Jay Fischer	Roopali Thapar	Charles Wesolowski
Steve Keim	Peter Thomas	

uTips (Usage Tips) provide insight into potential uses of function points to support an organization's business needs. While uTips provide insight on usage opportunities, they do not provide detailed direction on the application of the IFPUG FPA method in a particular situation. When necessary, the uTip maybe be followed by additional content on the topic providing specific how-to guidance. uTips are not rules, but interpretation and application of the rules, and provide guidance using a realistic example to explain the topic being covered.

This uTip is focused on describing how the IFPUG FPA method can be used in COTS Acquisitions. This uTip includes insight but is not an exhaustive examination of the subject.

Introduction

As businesses look to their Information Technology (IT) organizations to maximize their return on investment (ROI) and turn these organizations from cost centers to profit centers, they are turning increasingly to the use of COTS software acquisitions to achieve this.

While in many cases this may be a wise strategy towards fulfilling the business needs, there is no guarantee that the use of COTS will result in a cheaper, faster, and higher quality solution for meeting those needs over greenfield (in-house) development.

There is a prevailing belief, or rule of thumb, within the IT Industry that if a software solution can be purchased that will provide at least 80% (the 80/20 rule) of the desired business functionality it will be more cost effective to buy rather than build. However, this "rule of thumb" may not always hold true and many programs end up being over budget, late or even ultimately cancelled.

This white paper will address how the use of the International Function Point User Group's (IFPUG) Function Point Analysis (FPA) methodology can help business decision makers make to correct choice when faced with a build vs. buy choice as well as determining which COTS acquisition is the best choice.

Types of COTS

Before discussing how FPA can aid in the acquisition process, we should define what COTS options are available.

The traditional definition of COTS software is the software sold by a commercial software vendor, usually with a maintenance contract, for a defined period of time. The vendor will also perform any configuration and customization of the software required. However, there are other options when it comes to acquiring software. One such option is Government off- the-shelf (GOTS). This is software developed by government agencies that is available for little or no cost to other government agencies. There is also Free and Open Source Software (FOSS), which is usually developed by volunteers who work together on projects collaboratively and release the software free of charge under open commons-type agreements.

While there are cost advantages to using either GOTS or FOSS over more traditional COTS, there are also drawbacks. The greatest of these is the lack of any type of technical support. Generally, the customer is usually "on their own" with regards to implementation and maintenance support. There may be online user group support available, but it can be sporadic, incomplete, or inaccurate. In some cases there may be the option to purchase support provided by the owners of the software. These types of engagements tend to provide a more professional grade level of service and are generally inexpensive when compared to traditional COTS software vendor support contracts.

Use of Function Point Analysis to assist with COTS Acquisitions

There are a number of ways in which Function Point Analysis can help to ensure that the best choice is made during the procurement process. Function Points were developed to answer just these types of business questions while providing the information needed to key decision makers that will facilitate this process.

The initial step that should be taken is to perform a Function Point count on the desired user requirements. All too often organizations are reluctant to initiate a count until the requirements are "finished." The reality is that in the vast majority of programs, the requirements are constantly undergoing review and change as part of the software development process. This is true regardless of which development methodology is used, so it can be detrimental to the program to wait.

One of the great strengths of function point analysis over other software sizing methods, particularly SLOC, is that a reasonably accurate size can be estimated early in the software development lifecycle. There are a number of techniques such as Early and Quick Function Points ™ and Fast Function Points which can be employed when the requirements are not at a granular enough level to perform a comprehensive FPA.

Once the key stakeholders are in agreement that the requirements are complete, have been signed off on, and are now under change management, the function point analysis can then be updated to more accurately reflect the true size of the program.

Now that the software size has been established, the next step entails surveying the market space to determine the potential vendors who can provide the desired software solution, either completely or perhaps partially.

If the products are in a mature market space and are not employing cutting/ bleeding edge technologies, there are likely to be numerous products available. Examples of mature COTS product spaces include accounting systems, telecommunications, Business Process Management (BPM), or Configuration Management (CM) applications. These types of products also tend to have more options available in the FOSS or GOTS areas as well. For more technically advanced or specialized functionally, such as targeting systems, radar, or medical systems, there may be few commercial options available.

Once the market space has been defined, market surveys can be conducted, as well as performing industry days and product demonstrations to further refine the field of potential candidate vendors.

At this point, if primary cost data is available from vendors, a build vs. buy analysis can be initiated. This entails developing a size-based cost model using the function point size of the requirements (parametric or analogy-based estimates work best at this point) to determine if it may be more feasible to develop the solution in-house rather than purchase a COTS product.

Once the independent estimate is developed, keeping in mind that it is still considered a high level Rough Order of Magnitude (ROM) estimate, particularly if analogy is used; a determination can be made as to whether or not it is advisable to move forward with a COTS acquisition. If the cost comparison does favor the in-house option, it is critical that the resources with the requisite skill sets are in-house and will be available to work on the program.

If a decision is made to pursue the COTS acquisition, the function point analysis can be used for a number of purposes. First, function point analysis is performed on the candidate products. In some cases, the vendors have already done so and may be willing to provide the data. However, if not, a function point analysis can be performed using training manuals and data schema (if available). If a demonstration version of the software is provided, that can be used as well. It should be noted, that depending on the number of candidate products, availability of documentation and the vendor's willingness to support this effort, it can take a considerable amount of time and effort to complete this activity, and this should be factored into the overall project schedule. Despite this, the potential benefits gained from this effort far outweigh the costs.

Upon completion of the COTS FPA, the first criteria for suitability can be determined. This is the application of the 80%/20% "rule", which states that if 80% of the desired functionality can be delivered by the COTS product, it is likely more cost effective to buy rather than build. However, this "rule" should be used with caution. Key to determining if it is applicable to the particular COTS product under review, the analyst needs to understand the relative importance for any functional gaps. This is why the program requirements should be rated, ranked or prioritized. There are a number of techniques that can be employed such as the Analytical Hierarchy Process (AHP), Theory W (or Win-Win), 100 point system, requirements triage or Numerical Assignments Technique to name a few. At the very least, they should be categorized as must-have, desired, and nice-to-have.

If the functionality gap lies mostly with nice-to-have or low priority requirements, then it is likely that the COTS product is a suitable candidate. This assumes that the missing functionality can be built relatively easily or de-scoped. However, should the missing functionality be mandatory functionality that would definitely have to be built, then perhaps another candidate should be selected which may still not meet the 80/20% rule, but provides a greater amount of the required functionality.

It should also be confirmed that the product is not providing a large amount of unneeded functionality (gold plating). If the product is modular and the customer can purchase only the modules that meet the desired functional requirements, then this can be avoided. In the case where modularity is not an option, then caution should be exercised if there is a great deal of unrequired functionality in the COTS package. This will avoid paying extra for unused functionality. Another factor is the additional costs associated with maintaining the unnecessary software.

As with any acquisition, keep in mind that the vendors are likely to assume or promise that their product will meet most if not all the functional needs of the customer. While this may be true, the purchaser also has to consider a couple of other areas. One area that often gets overlooked is the database. Does the product offer the ability to support the requisite data formats, mappings, entities, logical groupings, and relationships (e.g., primary/foreign keys) and is the product easily configurable to meet those needs? Failure to adequately analyze the database aspect of the software can have significant impacts to the overall cost and schedule of the program if there is extensive work that must be done in this area.

The other oft-neglected aspect is the potential impacts to the business processes. While not directly associated with FPA, the effect any new software has on the business processes associated with the application(s) should be considered. One way to address this is to discuss business processes during function point interviews with subject matter experts. Reviewing the function point analysis in relation to the business processes that the associated functionality supports will also help in making this determination. Failure to consider these aspects of the acquisition can result in turning what was to be a COTS product with little need for customization into a heavily reengineered product that ultimately is more costly, risky, and takes longer to implement than it would have to build the application.

In addition to Best Fit and 80%/20% determinations, there are a number of FP-based metrics that can help. One of the better ones is the Cost/FP, which will determine the best overall value. This should be broken down further by analyzing these costs in relation to the ratings associated with the user requirements, as the cheapest product may not deliver the most required functionality. This cost/FP metric can also be applied to the maintenance costs for the installed baseline. Taken with the number of staff or FTEs required to maintain the application can show which will be the more effective solution in the long run. Keeping in mind that any functionality that needs to be built or customized will increase these costs.

Defect density goals/incentives can also be developed based on function points to ensure the software is of sufficient quality. As cost, schedule, and quality are the 3 key metrics of software development, this metric provides a reliable metric for which to measure quality.

Configuration vs. Customization

Oftentimes when discussing COTS, confusion arises as to what the difference is between configuration and customization.

Configuration is defined as "The way a system is set up, or the assortment of components that make up the system. Configuration can refer to either hardware or software, or the combination of both.[1] " Another way to define it as it applies to computers:

> a. to put (a computer system) together by supplying a specific computer with appropriate peripheral devices, as a monitor and disk drive, and connecting them.
>
> b. to set up (a program) to enable it to run on a particular computer or for a particular application. [2]

[1] http://www.webopedia.com/TERM/C/configuration.html

[2] http://dictionary.reference.com/browse/configure?s=ts

Customization, on the other hand, involves modification to the source code to function differently or to make (something) according to a customer's individual requirements. [3]

If there is confusion over how the parties define each term, it will have an adverse impact on the engagement. Configuration should be a relatively straightforward process by which the product is set up to operate in the requisite environment. It generally does not involve significant amounts of time or effort.

If configuration is the term used, but meant to imply customization instead, this will likely result in program delays as well as cost and schedule overruns. Customization should be treated as software enhancement, requiring the associated level of analysis, design, and testing time and effort as newly developed software. Care also needs to be taken to insure that any customized code developed does not break the code that is untouched.

Similarly, making cosmetic changes to the software, such as changing themes, moving the location of particular fields on a screen, or other "look and feel" changes are akin to configuration. They usually do not require significant amounts of effort or time. IFPUG has developed a complementary sizing methodology for non-functional requirements, Software Non-functional Assessment Process (SNAP). For more information on how to use SNAP, see the IFPUG website (www.ifpug.org).

Changes to how functions operate, even minor ones, on the other, hand should be treated as customization. Again, here clearly defining the terminology can avoid problems.

Summary

While in many cases COTS may in fact be the most time and cost effective solution to delivering needed business functionality, it should never be automatically assumed that this is the case. There are multiple factors to consider when making a build vs. buy and the use of IFPUG Function Point Analysis can provide the necessary data to make the correct choice. Function Points provide the added benefit of being useful throughout the project lifecycle and their proper use can greatly increase the chances of a successful outcome.

Further Reading

N/A

[3] http://www.thefreedictionary.com/customization