

# Sales Commission Applications – Build vs. Buy

## Overview:

Prior to the 1980s almost all companies built their own software applications to handle their needs. Every company believes that its business processes and challenges were so unique that only software built by its own business and IT staff would be adequate to solve the problems. Surprisingly this process extended to applications such as General Ledger and Payroll. An interesting example is the case of PG&E, the energy firm on the west coast. PG&E actually built its own operating system.

Over time, with application vendors such as Dun and Bradstreet, Oracle Applications, PeopleSoft HR, SAP R/3, QuickBooks, Peachtree etc, most business applications have been standardized and created as installable packages. Still the choice between Build and Buy is often debated within many companies. This article considers the choices in the context of the need to have a working sales commission application.

‘Buy’ typically defines the purchase of a standard vendor package for the application, being used by other companies in the marketplace. ‘Build’ typically defines the building of the application with internal resources, external resources or a combination of both. This article discusses the benefits of a Buy choice compared to a Build choice.

## Sales Commission

Sales commission, unlike many other applications tends to be fairly unique from company to company, though there is significant commonality within an industry. The other unique aspect of sales commission is the flexible nature of customer needs. It is fairly common for a company to change its commission plans year to year or even within the year. This is unlikely to happen with applications such as General Ledger and Payroll. This critical need for a high level of flexibility requires a very dynamic application that can accommodate itself to changing needs. To make this possible the sales commission application is normally built around a rule-based architecture.

## Buy

Buying a packaged application typically consists of the following steps:

Requirements: Understanding the business requirements and documenting

Vendor review: Reviewing vendor information and comparing to requirements

Purchase: Choosing the appropriate vendor and purchasing

Testing: Testing the vendor solution in the customer quality assurance environment

Implementation: Going live with the vendor solution

Maintenance and Support: ongoing support for the application for the life of the application

Most resources typically come from the application vendor. Internal resources are limited to the business subject matter expert (SME), project manager and if really necessary, technical resources to do simpler tasks such as report writing, extracting data etc.

Buy timeframes are typically shorter, since the software is already built and ready to go.

## Build

Building an in-house application typically consists of the following steps:

Requirements: Understanding the business requirements and documenting

Resource sourcing: identifying internal resources or reviewing external resources and selecting

Knowledge transfer: business will have to teach the technical resources about the business application

Design: Design of the application

Development: Development of the application

Testing: Testing the solution in the customer quality assurance environment

Implementation: Going live with the vendor solution

Maintenance and Support: ongoing support for the application for the life of the application

Build projects typically require a business SME, a project manager, developer resources to code the application, other resources for report writing, extract of data, etc. The developer resources can be internal or sourced from external contract firms. The time requirement on the SME and project manager will be far higher because of resource sourcing, knowledge transfer and design functions.

Build timeframes are typically much longer, because of the effort and time involved in Knowledge Transfer, Design and development.

## **Factors**

### **Business Knowledge**

The amount of domain knowledge and experience as it relates to sales commission is a strong factor to consider in the choice of Build vs. Buy.

#### Buy

Application vendors, who have built the packaged application, typically have dedicated resources that have studied the particular application area. They would have gathered requirements from a lot of customers and prospects to build the application. Even though the vendor resource, may not know the subset of business needs that makes any customer unique, they tend to know the vast majority of business practices involved in the concerned business area. They do not have to be taught many of the concepts and practices involved.

In the sales commission area, the typical application vendor resource will already be aware of concepts such as compensation plans, splits, overrides, draws, SPIFs, etc. As a matter of fact, customers typically rely on the vendor's expertise in the broader areas of sales commission to help them determine the correct business practices to follow.

#### Build

The developer resources used for building the application can be chosen to make sure they have the right technical background to use the underlying software. But invariably, they will not know most of the business processes, terms or requirements involved. In most cases the entire basic concepts of sales commission, crediting of transactions to payees, calculating flat commission rates vs. tiered rates, calculating commissions on order, invoicing or payments, etc have to be explained to the resources involved. This typically takes large amounts of time from the SME, who normally already holds down a full-time function.

One customer explained it like this: "I asked for a car to be built, but I got a car that could go from A to B, but did not have the ability to reverse or have rear view mirrors". The custom-built application for this customer could not handle negative numbers in returns and consequently did not process them at all. The customer paid out more commissions that he should have.

Not knowing the domain of sales commission also means that the resource has no experience in the typical problems that can happen within the domain. For example, what happens if product is returned? At what commission rate should the product be taken back, especially if the commission rate varies over time? What if today commissions vary by client industries and tomorrow they vary by product families? This rigidity can at best cause time delays, or worse, calculate erroneous commissions.

Clearly there is a compelling business benefit to having knowledgeable people staff the project. It will save a lot of valuable business resource time as well as avoid future problems.

### **Total Cost of Ownership**

A key factor in contemplating a build-vs.-buy choice is to reduce costs. Superficially, there is a sense that build projects will cost less. But careful analysis shows that many costs are overlooked in making a decision.

### Buy

Application vendors' costs are typically very clear and upfront. The three components, licensing, maintenance and development can be known with a fair amount of certainty and budgeted for. Typically vendors have room to negotiate on one or more of these components.

Even though the initial cost of license and services could be high in comparison, over time, the cost tends to be less than the maintenance involved in custom-built projects. Implementing a packaged application typically takes considerably less time and effort than a custom-solution.

### Build

For custom projects, the amount of internal resources dedicated to the project tends to be far higher than packaged products. Much of the effort goes into the business people teaching the custom developers the business, before it can be automated. This affects both business and IT and there is an opportunity cost of this resource not being able to handle their own responsibilities.

A media firm decided to build their commission solution in-house. They hired a Visual Basic (VB) programmer at \$85/hr for two months and expected to be complete with their application. Inevitably, the project took longer than anticipated and after four months, the total cost ended up to be almost \$60,000. The application was too brittle to handle the subsequent commission plan changes and they ended up buying a packaged sales commission solution where, the license and services ended up costing less than \$40,000.

In the case of custom development, the design, development are additional phases in the project and tend to add to the time line and resource costs. Over time, the custom application needs to get modified to accommodate the business changes continuously and the costs involved are high. The application also gets more complex and the maintenance costs tend to grow with time

When all the costs are taken into account, packaged applications tend to be less costly to the customer than custom-built applications.

## **Application Functionality**

The comprehensive nature of the functionality in the application being implemented has to be considered in the Build vs. Buy choice. Customers tend to have very specific requirements surrounding their current business problem and tend not to be conscious of future needs. When their needs change, they are looking to their current application to accommodate their needs. If the live application is unable to meet those needs, they end up either curtailing their business changes or spending more money on replacing their systems.

In the sales commission area, applications tend to be able to calculate commissions for many different industries and use many different methods. For example, it will be possible to calculate commissions in many ways: flat rate; tiered rate; tiered rate based on quota; different rates by product; different rates by customers, etc. Another example would be where the customer can calculate commissions and bonuses on revenue, gross profit, quantity, etc. Customer will start with one kind of a plan and very likely move to a different type of plan over the next few years. So flexibility and rule-based application architecture will be important requirements for the software.

### Buy

Application vendors would have gathered requirements from a lot of customers and prospects prior to building the application. It is not uncommon to have the solutions built to handle the requirements of many different industries. So most of the business practices a customer will tend to use, now or in the future, will already be available in the application. In addition, the vendors get constant feedback from their customers and prospects as to gaps in the application and make constant upgrades to the application.

In addition, application vendors attempt to reduce support calls and so provide a lot of additional tools to help the customer be self-sufficient. Tools such as report writers, data extractors, email capabilities, logging facilities, etc, all tend to be built into the application.

Applications vendors would understand the amount of change in sales commission plans and will build in a lot of flexibility into the software. In addition it is common to see sales commission plans expressed as rules, in this kind of an application.

### Build

The developer resources will do their job diligently in asking questions and understanding the customer requirements. And the business SME will provide the exact requirements necessary to solve their current problem. But the operative word is 'current'. The solution will tend to match the current needs well. The solution normally does not take into account possible future changes. But sales commission plans change frequently. So very soon after the implementation, the customers find themselves calling in the resources for further work or abandoning the application altogether, for another attempt.

A computer hardware reseller built a commission solution to pay his reps commissions, on sales revenue. Within a year, he found that, sales people were discounting prices to get sales and impacting his profitability. He decided to change his sales commission plans to calculate commission on gross profit rather than revenue. But unfortunately the custom-built solution had no facility to handle gross profit.

The average custom development resource is not charged with building all the tools necessary to have a productive application environment. It is rare that customers ask the developer for report writers, data extractors, PDF generation, etc. The custom-built applications tend to rigidly satisfy the expressed requirements and won't be flexible to handle future changes. Rule based programming is complex and takes significantly more effort, and is typically not part of the charter of a custom project.

A comprehensive application preserves the options of the customer for future changes and reduces business impact and total cost of ownership of the product. Buy solutions will provide this in an economical manner.

## **Resource Risk**

Customers invest in the product choice they make and the resources that support the product choice. To support their ongoing business, they have a reasonable expectation that the resources involved with their product and especially the knowledge of the product is available to them.

### Buy

Application vendors are dependent on ongoing revenue from their products for their business survival and success. To that extent they invest considerably in research, development, consulting, support, sales and marketing functions. Their success depends on their expertise in their products.

Continuous availability of knowledgeable experts allows them to serve their customers effectively, reduce support costs and increase the possibility that the customer will buy additional products from them. So there is very little risk that the vendor will not have resources that understand and handle the customer's problems.

### Build

Customers choosing the build option either use their internal resources or hire resources with the right technical background for the job. After the job, the internal resource may be assigned to other projects that consume their time. External resources will generally be assigned to different projects for different customers. Worse still, the internal resource may leave and likewise the external resource, leaving the customer with an orphaned application with no way of modifying it.

A staffing firm had its technical resource build a very complex commission plan application in Visual Basic and Access. The resource quit a few months after the application was live and went to a competitor's firm. Very soon after, the application had to have some bugs fixed and some of the plans changed. The staffing firm was placed in a bind, with no easy way to correct the situation.

In the case of external resources, any time the application has to be modified, the resource has to be rehired. If the resource is available at that particular time, the customer has to pay for the time necessary to refamiliarize the resource to the application as well as the actual change. This typically tends to be a significant proportion of the original cost.

Resource risk is reduced significantly when the software is bought from a packaged application vendor. Application vendors can ensure future support and maintenance of the application.

## Quality

Software development is a complex endeavor and almost all software ends up having a significant number of bugs. Companies are also growing continuously and require their systems to have the ability to handle their growth. The better the quality of the software, the easier the user's experience will be in the long run.

### Buy

Application vendors invest a lot more into building their software than the average customer will. It is not unusual for vendors to invest millions of dollars into software that retails in the tens of thousands. The research and development process typically includes a quality assurance component. Software is typically released into the marketplace only after the quality is verified. The vendor is also getting bug reports from many different customers and typically have a bug fix and patch release program to continuously get the fixes out into the marketplace.

Applications are also designed to operate across many different sizes of customers and all customers get the benefit of performance goals specified for the largest customers. The customer gets the benefit of tried and tested code with high performance characteristics.

### Build

One of the biggest problems with custom projects is that the entire code base tends to be brand new. So the customer lives through the identification of resolution of bugs in the software over time, but in the real live production environment.

A manufacturing firm created a custom project to handle its commissions, where the commissions were paid by the client industry. Commissions were calculated on the complete invoice, so they only required one record per invoice. This worked in an acceptable manner for the few thousands of invoices they processed every month. Then their business dictated that they shift to commissions based on different products, which required that they calculate commissions at the individual invoice line level. This increased the amount of transactions they had to process ten-fold, and the custom-built system bogged down, unable to process the huge transaction load increase.

Because of the change inherent in commission plans, it is not predictable how much of a volume the application has to process. Growth can be rapid; the types of plans can change dramatically; so the typically custom project may not be robust enough to handle these performance needs.

Vendor built applications tend to have higher quality and can definitely handle higher performance needs.

## Conclusion:

Even though business operations have consistently moved towards packaged applications in large, medium and small enterprises, some companies still consider building their own application. Sometimes this happens even for standard business applications such as sales commission applications.

It is clear that in the case of most applications- and sales commission applications in particular- it is a much better decision to make the choice to Buy a packaged application, rather than the alternative.

## Author:

Gopi Mattel, an executive at CellarStone, Inc., a provider of sales commission solutions and services.

[gopi\\_mattel@cellarstone.com](mailto:gopi_mattel@cellarstone.com)

[www.qcommission.com](http://www.qcommission.com), [www.easy-commission.com](http://www.easy-commission.com), [www.qxchange.com](http://www.qxchange.com), [www.cellarstone.com](http://www.cellarstone.com).