

Software Implementations – “Out-of-the-Box” or Intelligent Modification

What do all retailers have in common? The answer is simple. They strive to provide the right product at the right time to the right location. That’s where it ends. The biggest challenge retail software vendors face is how to provide an all-inclusive “out-of-the-box” solution that satisfies the unique requirements for a wide-spectrum of retail organizations.

The challenge is even greater for the retailer. What package will best suit the needs of the organization right off the shelf? The day of the custom “homegrown” merchandise system has come and gone, and this raises the primary question every retail IT executive faces when making package selection decisions. Do I change my business processes to fit the software solution, or change the software solution to fit my business processes?



Integrating people, process and IT

www.rpesolutions.com ■ 813-490-7000 Ext. 7353
info@rpesolutions.com



“Out-of-the-Box” Software Implementations

Software vendors are very adamant about discouraging their customers from modifying proprietary base code. The risks to the retailer are spelled out clearly in every software contract. Do it and you relinquish your product support and jeopardize the ability to install future version releases. Today’s retail software packages are built around the ever-changing best practices of the retail organization. They are designed to be a cost-effective means for retailers to implement and apply these best practices with a relatively quick ROI and low cost of ownership.

The benefits of implementing a retail software package without modifying a strand of code are evident. The overall cost of ownership is far less than the incremental/ongoing costs involved in coding, testing, documenting and training users on a modified version of the vendor package. In addition, implementing a vendor’s software package without modification eliminates the risk of losing vendor product support – specifically bug fixes/patches and new functionality releases.

Software vendors typically release new versions of their software for a couple reasons:

- To apply major bug fixes/patches found by their customers
- To address technological changes including data base platform upgrades and performance needs
- To apply constantly changing business requirements and functional gaps identified through the best practices approach

The number one fear that any IT department who has modified a software vendor’s base code is, “How do we handle patches for bug fixes and version releases for enhanced functionality?” This fear alone usually is enough to end any thoughts of going down the modification road.

The greatest challenge a retailer faces when implementing an “out-of-the-box” solution is change management. The expectation is that the business has to be rebuilt around the functional limitations of the software package.

Business process redesign is an unavoidable requirement when implementing a new software solution. The concept of change almost always creates push back from the user community with thoughts of changing how they have to do their job, not to mention the fear that the new system may altogether eliminate their job. The change management aspect of the implementation can be addressed through formal business process review where “as is” and “to be” processes are defined, documented, and implemented. With time, the fear of change shifts from denial to acceptance as the efficiencies of the new software and business processes are realized.

Intelligently Modified Software Implementations

The vendor package selection process should determine which software package has the least number of functional gaps. Functionality gaps are inevitable. In some cases, an “out-of-the-box” software solution is just not a viable solution for a retailer’s business and the decision to build an “in-house” solution results from this.

Sometimes these critical functional gaps are found after the software package selection process and the retailer has two options:

- (1) Postpone the implementation process, or
- (2) modify the code to satisfy the business requirements.



Modification to a software vendor's package is a viable solution only when the retailer has determined that the benefits outweigh the costs. In some cases, a single critical functional gap identified in the software can drive this decision. In other cases, depending on the severity, the gap in functionality identified by the retailer finds its way into future software version releases. The question is whether the retailer is patient enough, or confident enough that this will happen.

Postponing the implementation process is a costly decision, and modification becomes a necessary evil. Vendor contracts have been signed, hardware purchased, the consultants are onsite, and management has an expectation for IT to deliver. The result – proceed with modification to satisfy the business requirements. As long as the retailer is diligent in deciding what should be modified based on the severity of the gap, there are realized benefits to modification.

The benefits to intelligently modifying an "out-of-the-box" solution include:

- Satisfying the end user requirements
- Eliminating any critical functional business gaps not satisfied by the vendor's software solution that may have an impact on the retailer's competitive advantage and ultimately their bottom line
- Eliminating the fear of change management, specifically business process redesign

Some would challenge that the elimination of change management is actually a weakness. Why? Often new software systems are selected and implemented to streamline business processes. Retailers can argue it either way in this option and it depends on the overall cost-benefit analysis involved. Though the option to modify is more costly, and the risks involved with bug fixes/patches and new version releases are greater, the cost and risks can be justified by the retailer's ability to address critical functional gaps that can make or break their bottom line.

Which Decision is Best?

So, which is better? It's a case-by-case decision. If the retailer's business model is not complicated and has the flexibility to mold its business around the functionality of the software package, then "out-of-the-box" is the right decision. In addition, if the retailer's IT budget is small, then the costs involved in staffing the in-house resources required to support a modified package may not be feasible. Leave it to the software vendor to support the software.

Option 1: no modification

- Do not change vendor package
- No cost incremental cost, low risk, vendor support

Option 2: modify vendor software

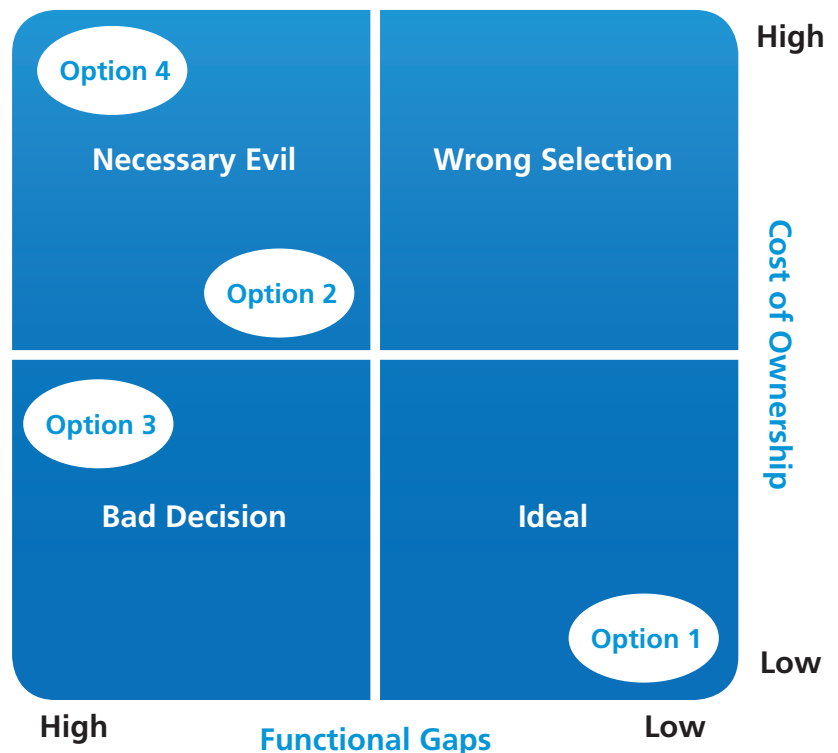
- Modify code to meet functional requirements
- Additional incremental cost, ongoing costs related to managing, high risk, no vendor support

Option 3: postpone implementation

- Do nothing, keep existing system
- Sunk cost to software licenses, hardware, consulting fees

Option 4: custom in-house solution

- Made to order system
- High upfront and maintenance cost, high resource requirements





Larger retailers with more complicated business practices tend to be good contenders for modification (Option 2 in Illustration). They tend to have the capital resources and IT head count necessary to manage a modified software package. Typically, these retailers manage their business in such a unique way that modification is the only option to run their business on the new software solution.

In some instances, the retailer may have enough influence on the software vendor to incorporate the functional gaps into the base code without the need for modification. Yet, this is not always the case and when the retailer's competitive advantage is impacted, modification is essential.

Often the first step in the process is to conduct a more detailed cost-benefit analysis to assess the organizational impact. Once it is determined that the vendor software requires modification, the retailer should be diligent in conducting discovery sessions to identify all functional gaps, distinguish the necessary modifications from the nice to haves, prioritize them and develop a strategic plan to implement.