PACKT
PUBLISHING

Maximize Your Investment: 10 Key Strategies for Effective Packaged Software Implementations

Grady Brett Beaubouef

From Technologies to Solutions

Maximize Your Investment: 10 Key Strategies for Effective Packaged Software Implementations

Accelerate packaged (COTS) software implementations, increase returns on investment, and reduce implementation costs and customizations

Grady Brett Beaubouef

PACKT

Chapter No. 2

"Focus on Business Results"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.2 "Focus on Business Results"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Grady Brett Beaubouef, PMP is a project manager and solution architect for enterprise packaged software solutions. Brett has over fifteen years of packaged software implementation experience across several implementation roles including Project Manager, Solution Architect, Functional Lead, Technical Lead, Business Analyst, Software Quality Analyst, and Trainer. Brett also worked in a thought leadership role for the #1 business software maker, focusing on implementation methodologies, project assessments, and accelerated implementation services. Brett has a B.S. in Computer Science from LSU and is both a Project Manager Professional (2004) and a Certified Information Systems Auditor (1995).

Maximize Your Investment: 10 Key Strategies for Effective Packaged Software Implementations

Starting back in the 1980s, Commercial Off-The-Shelf (COTS) or packaged software such as Enterprise Resource Planning (ERP) applications were deemed to be the panacea to business pains caused by operational inefficiencies and disjointed applications. This resulted in an exponential growth in the ERP marketplace. To quickly meet this demand, ERP vendors and implementation partners used existing Software Development Lifecycle (SDLC) methodologies like Waterfall as the de facto implementation approach for ERP. This quick-fix decision resulted in an approach that was not effective for packaged software implementations, and that caused several issues. These included:

- Unnecessary requirements being captured (i.e., requirements gathered based upon limitations of existing systems).
- Requirements validation happening late in the implementation cycle.
- Highly-customized solutions that left customers with the same or even more challenges.
- Unrealized business results and benefits because the implementation focused only on the software.

In my fifteen years of implementation experience, I have been fortunate to play the roles of Information Technology (IT) auditor, functional consultant, technical analyst, programmer, Data Base Administrator (DBA), business analyst, solution architect, and project manager. Through my experiences, I have formed the following observations:

- Customers are more concerned with implementing successful business solutions, not just installing software products and technologies.
- Leading implementation methodologies are not focused on all of the components of a business solution. These components must work in unison in order to generate business value.
- Every business solution implementation is a "point-in-time" solution.
- Flexibility in a business solution starts with a flexible implementation approach.

Over the past two decades, the ERP industry has made incremental improvements in the implementation of enterprise business software—specifically, in the areas of implementation tools like industry-specific preconfigurations, online software product setup assistants, and data conversion tools. These improvements provided value from an efficiency perspective, however, there was little accomplished to address how to make ERP implementations more effective at delivering business value.

The strategic value of purchasing ERP—or any—packaged software is to reduce the customer's Total Cost of Ownership (TCO) for their existing business system, as well as allowing the customer to focus on more strategic objectives. The customer would pay some type of maintenance fee to the software vendor, who would then provide support and upgrades. In theory, this approach seems mutually beneficial to all players. However, the reality is that packaged software customers have not been able to experience the lower TCO due to the following:

- Initial implementations are taking longer and cost more than originally planned.
- Software upgrades are more costly because implementation approaches focus on turn-key, point-in-time business systems, and not on putting the customer in the best position to leverage future COTS software upgrades.
- Customers were never given the complete holistic approach needed to optimize their new enterprise business solution.

If you think I'm only speaking of software, then I suspect that you are one of the many people who believe that ERP has been a tremendous disappointment.

Vision of the future

To get to a point where we can get customers to experience the full benefit of their ERP investment, we must EVOLVE our way of thinking on ERP implementations—or any packaged software implementation. The ideal COTS software implementation approach would focus on maximizing the "out of the box" value that packaged software can provide to a customer. The implementation approach would naturally filter out requirements that did not provide quantifiable business value, and keep the focus on the customer's value-added strategic requirements. There would be no need for post-production support provided by implementation partners because the customer would be confident in supporting their business solution. Upgrades are done in weeks instead of months or even years. The project team would have a common language (technology, business, software) that they could speak, in order to collaborate effectively. Validation of business requirements would happen early and often. Organizational change would be manageable because it would be minimized. Business sponsors and end users would see and touch the business solution months before end user testing. Individual project team meetings would generate more decisions and less action items. Implementation costs associated with packaged software would be less than the normal—which is four to six times the cost of the software—because implementation partners would spend more time enabling customers to lead, versus performing staff augmentation. Customers would be left with an actionable roadmap to further leverage packaged software functionality as their business model evolves.

The implementation of packaged software is more than just installing and configuring software: it is the implementation of a business solution. A business solution is far more than just the software. Today, the majority of implementation approaches are guilty of focusing on one of the obvious components (products and technologies), or all of the components separately but not in unison throughout the implementation lifecycle.

Purpose of this book

The main objective of this book is to discuss an approach to implementing COTS software that will maximize the total ownership experience. This book focuses more on the art of effectively applying implementation methods for COTS implementations. As the IT market continues to move to a component-based software development paradigm, using available enterprise-wide software packages and "best-of-breed" applications in the marketplace, I believe this approach will become more relevant.

This book also serves as a challenge to implementation partners and internal Information Technology (IT) organizations that support packaged software implementations. This book will address the change in approach and direction we need to take as we evolve packaged software implementations to the next level, and generate greater business value for our customers. I will discuss different implementation methods and demonstrate how these perceived competing approaches can actually complement one another—it's just a question of identifying the appropriate level, and knowing when to apply these different disciplines. This book will identify the major factors that must be considered when defining the appropriate implementation approach.

I will also spend time in addressing packaged implementation perceptions and myths, in order to better define expectations of a COTS implementation. I firmly believe that business process methodologies like Business Process Management (BPM), Business Process Re-engineering (BPR), Lean, and Business Quality (e.g., Six Sigma) can and should intertwine within a COTS implementation lifecycle.

Out of scope for this book

The purpose of this book is not to rehash ideals and concepts that are present in the general mindshare of customers, implementation partners, and specialists regarding a successful implementation. This book will not define a new implementation methodology—there are plenty of methodologies out there, with their inherent advantages and disadvantages. This book will provide guidance on determining how to appropriately apply methodologies for packaged software implementations.

What This Book Covers

Chapter 1 is an introduction, covering the ten principles for implementing a business solution.

Chapter 2 covers how to ensure that packaged software implementations focus on value-added business results.

Chapter 3 covers how to effectively create alignment with implementation partners by performing a formal knowledge transfer of the existing business solution.

Chapter 4 outlines a key approach that implementation partners can take, in order to ensure long-term success for their customers.

Chapter 5 discusses a best implementation practice for gathering and validating business requirements for packaged software.

Chapter 6 discusses how to select and apply relevant disciplines when supporting the customer's unique implementation.

Chapter 7 discusses a best implementation practice for the initial implementation of the packaged software.

Chapter 8 discusses a software change strategy for COTS software that will minimize potential risks and maximize opportunities for additional software value.

Chapter 9 outlines the negotiation strategy required for implementing COTS software that will maximize the ownership experience.

Chapter 10 defines a new project role that will increase the success of the business solution implementation.

Chapter 11 focuses on effective knowledge generation.

Chapter 12 revisites the principles expanded upon in this book, and discuss how to evolve our strategy for implementing packaged software.

The *Appendix* is a summary of challenges.

2
Focus on Business Results

Begin with the end in mind. – Stephen Covey

Too often, in packaged software implementations, we readily equate software features with business results. We declare success when software goes into production. We assume (or hope) that the packaged software will drive the desired business results. What we sometimes fail to realize is that software is only one component of a holistic solution that drives business results.

Another slippery slope we deal with during the implementation is the division of labor in order to accomplish project activities. We divide work based upon specialization. This leads to project members having ownership for a piece of the work and limits the project team's ability to validate that the desired business results are achieved. This division of work will also result in additional effort required to organize, direct, and control the work.

The implementation of packaged software is only part of the greater effort of creating a new business solution—an alignment of people, business processes, and technology to support and drive business results. The objective of this chapter is to identify the key techniques that we can employ in order to ensure that every project team member, and thus the project, focuses on the value-added business results.

Challenging today's mindset

Before we can address the limitations of the traditional approach, we must first challenge our current thinking in regard to how we define project objectives, success measures, and scope.

We focus only on what we measure

How can one determine whether the project is focused on business results? I believe that what we measure is a good indicator of what we focus on. Consider the following measurement examples:

- Number of lines of code
- Project budget
- Project schedule
- Number of test defects
- Customer satisfaction

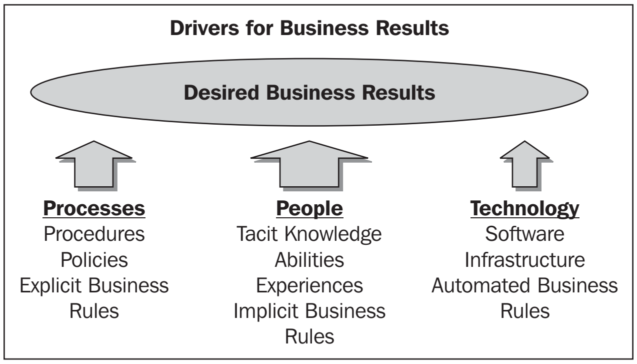
How do these metrics relate to achieving desired business results? I am not inferring that theses metrics are wrong, but what I am saying is that performance-related metrics focused on project execution should not be the only metrics that we utilize.

Project scope fixates on software features

Too often, I have observed packaged software implementations only focus on a subset of a business process, based upon the software features implemented. If you have a project that interacts with a business process then the implementation project needs to examine the entire business process to ensure that the desired business results are achieved. Otherwise, you assume that the upstream and downstream business activities will behave as they did before the packaged software implementation. What is more important to a business—installed software or reliable business results? I am persuaded to conclude that reliable business results are the focus of our customers. Now, let's spend a little time to address the main drivers for business results.

Focus on key drivers for business results

Let's briefly take a look at the key drivers for business results.



What is important to note here is that the implementation of a business solution not only considers packaged software but also the business processes and people that can have a greater impact on business results.

Technology is a great enabler that can play a role in generating business value. However, technology is not always the right approach to address problems inherent in a business model. Automation of a non-value-added business practice is not generating business value for the customer. "Making decisions and implementing them is where business value is created in an organization."[1] Technology on its own does not have the capacity to consider a diverse range of issues and then act. Knowledge is an attribute of people—not software. Technology can store information, business rules, and other data that can facilitate the decision making process. From a knowledge management perspective, information only generates value for customers when information is applied and a decision is reached. "All of the work that goes into development is not adding value until the software is in the hands of the customer." [2] It is people, not technology, that still make the business decisions that drive results.

Challenge to packaged software providers, and implementation partners

What are the strategic and tactical business decisions that your packaged software can support the customers in making? Leading packaged software providers and implementation partners should be able to provide this information up front to their customers. If the packaged software providers and/or implementation partners cannot provide these decision points then it begs the question of how well the provider/partner understands the underlying business model. If the software provider does not have a competent understanding of the underlying business model then the chances are that there will be significant gaps between the packaged software and your existing business model. Gaps always result in costing the customer money—either as software customizations or business process changes within your organization.

What results generate business value?

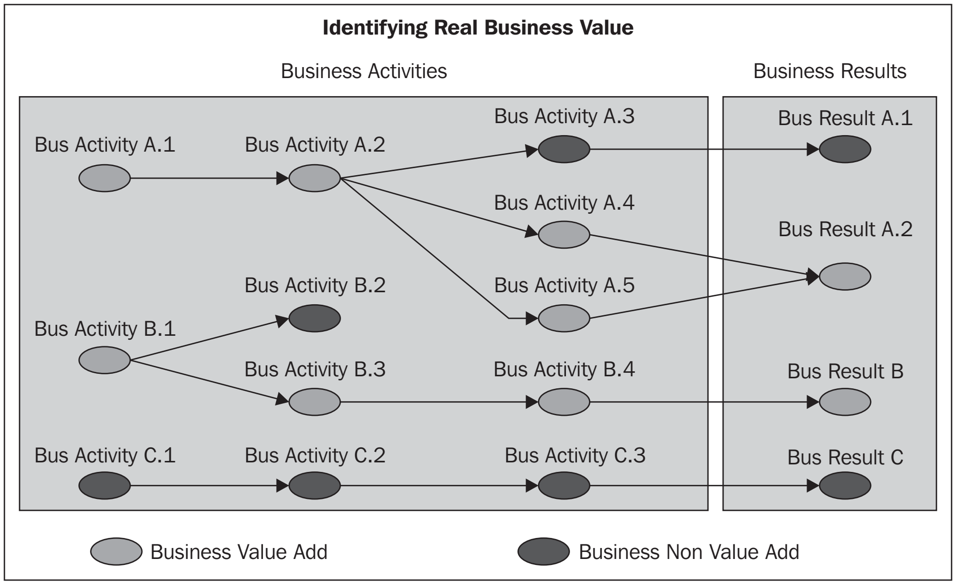
Early in my consulting career, I was naïve enough to assume that all business results generated by the customer's existing business model added value. I did not question the requirements defined in the customer's RFPs (Request for Proposals) and quickly proceeded to perform a fit/gap analysis. What I have learned from experience is that there are customer requirements that support non-value-added business activities and results. What I come to understand is that customer required can be based upon both needs and wants.

The traditional approach to requirements gathering and selection is reactive and full of non-value-added effort. The first step is to gather all requirements—regardless of whether a requirement generates real business value. The second step is to categorize and prioritize both the value-add and non-value-add business requirements. The third step is to perform an analysis on whether the packaged software can support the requirement. The final step is to conduct a formal fit/gap session to share the results of the first three steps and determine which requirements will be selected. There are three fundamental challenges with this approach:

1. Wasted effort is spent gathering requirements that support no business value.

2. It is easier to lose focus from the critical requirements that add significant business value. It increases the probability of missing value-added requirements due to a broader approach of capturing business needs and wants.
3. It sets the stage for stressful fit/gap sessions with various stakeholders fighting for their requirements to be selected.

A better approach is to minimize or eliminate non-value-added requirements from being captured up-front. Challenging requirements can be adversarial for IT and implementation partners. It is almost impossible if one does not have a competent understanding of the business model. A practical approach that has worked well for me is to begin with identifying the desirable business results, and then drilling back to the business activities and the corresponding requirements that directly support the result.



We will use the above illustration to reinforce some concepts. First and foremost is that the customer's existing business model will have both value-added and non-value-added business activities that will drive business results. Also, consider that the existing model may have both desirable and undesirable business results. Performing this type of analysis will take two iterations:

- The first iteration defines all of the business activities and associates these activities with the business results that they support.

- The second iteration focuses on all of the business results and categorizes as value-added or non-value-added. This activity is then performed for the associated business activities.

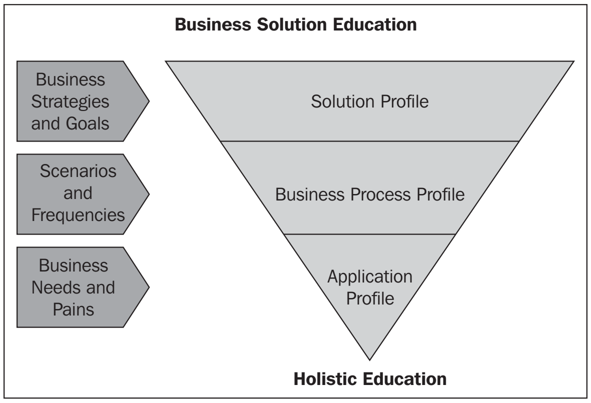
By conducting this analysis the implementation team can lay the foundation to identify the business requirements that directly support desired business results.

How to focus on business results during an implementation

Now, it is time to turn our attention to practices that we can employ in order to better ensure that the packaged software implementation project focuses on business results.

Conduct business training

An undisputed implementation best practice is for the implementation team to take packaged software product training. I propose that another implementation best practice is for the implementation team to take training on the existing business processes that will be supported by the packaged software.



Understanding the business is the responsibility of every individual member on an implementation team. In fact, I would make it a prerequisite! Every individual project role will benefit from the training—including developers, software quality assurance, DBAs, data architects, implementation consultants, and especially project managers. Now, I am not saying that each individual needs to have the same business knowledge as the customer, but there is a basic level of business competency that each member should have. This training will enable a common language to be developed, and will improve collaboration.

Challenge to IT and implementation partners

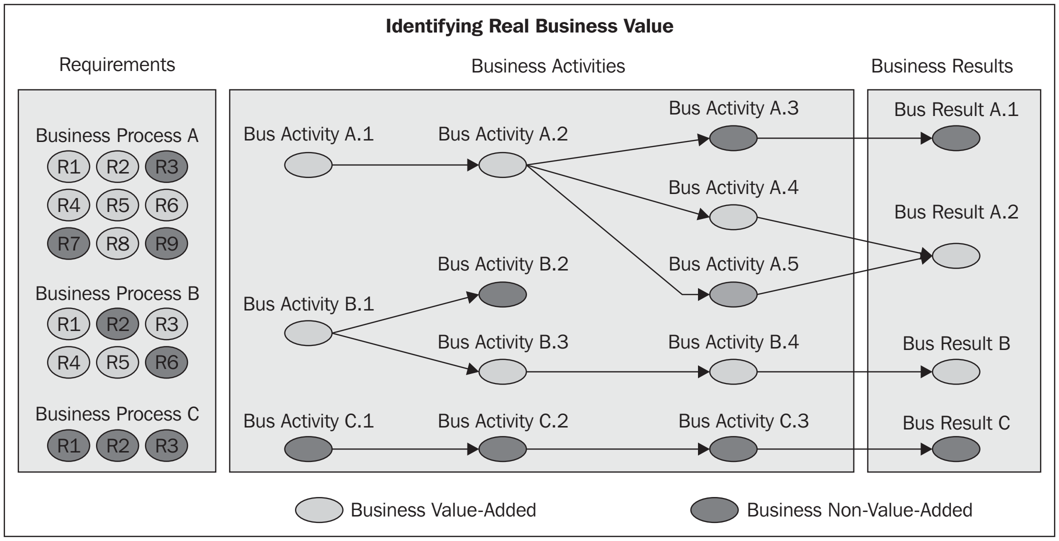
We should speak in business terms and concepts and not technical jargon when interacting with customers. The responsibility of translation should be on IT and the implementation partners, who should translate technical and software-specific concepts into existing business terminology. This practice improves communication and enables quicker decisions to be extracted from the Business. I understand that the traditional role of a business analyst is to translate between Business and IT—but I fear this is being used more as a crutch and not to encourage every partner to evolve their understanding of the business that they support. There is no better real-time, practical method to align IT with the Business than speaking the language of the Business.

Implementation documentation should be Business-oriented

Data models, generally speaking, end up being not very helpful for understanding the business model. Data models have a limited place and impact. What documentation is more important to your customer—an Entity Relationship Diagram or a Business Process Map? IT and Implementation Partners should have an appreciation for both; however, both parties should strive for the greater understanding on the Business Process Map. Remember that the value of packaged business software is that it should not be necessary to redesign the software. If you have to redesign packaged software then you did not pick the correct software package. Having implementation documentation (forms and templates) geared towards business users will result in encouraging the project team to be business focused.

Use value-added Business results to filter requirements

Let's expand on our previous illustration that identified real business value.



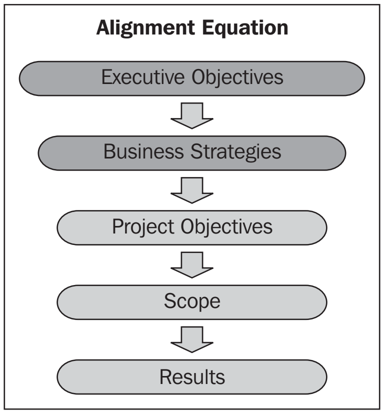
In this illustration, we have business results that are driven by a series of business activities (i.e., business process). Now, we have included a series of business requirements that support the activities for a given business process. The final step of this analysis is to categorize the requirements based upon whether the requirement supports value-added activities or nonvalue-added activities. By focusing on the value-added requirements, we can proactively select requirements and accelerate the requirement-gathering activities. This exercise is not only useful to the implementation partner but also to the Business. It is an approach that can help the implementation team to justify why certain existing business activities are not being carried forward into in the new business solution.

Challenge to customers

Customers always ask what they can do to prepare for their packaged software implementation. Performing a business value analysis on their existing business processes is a key exercise that will help to further refine focus and fit analysis—more so than requirements prioritization. For five years, I supported accelerated implementation approaches and I have concluded that focus is the greatest enabler to rapid implementation.

Project objectives should address business results

Implementing packaged software is not a key business result in and of itself. However, I have observed that the majority of packaged software implementation project objectives are technology oriented. The project objective and scope defines project focus and success criteria. It is key that the project objective clearly address of all the components of a business solution and is business-results oriented. A basic project management principle states that the implementation project objectives and scope should align with the overall executive strategy for a customer.


Alignment Equation

Executive Objectives

Business Strategies

Project Objectives

Scope

Results

Executive objectives are the highest-level goals for an organization. Business strategies are the high-level approaches to addressing the executive objectives. The project objective and scope are the corresponding tactical efforts required to support the business strategies. The implementation of packaged software is to support business results that satisfy executive objectives. A best implementation practice is to ensure that the project objective and scope can support meeting all of the expected business results by the packaged software implementation.

Summary

Before you can build a business solution, we must first understand the business, and the desired business results that generate real value for the customer. This is the first and most important step in implementing packaged software in such a way that we can provide maximum support to the customer's business value generation stream. Too often, Commercial Off the Shelf (COTS) software implementations focus on the individual functional pieces in silos and fail to validate that the end results—which are typically outside of the business software—generate the business value expected by the customer. Starting with the desired business results and working back to the value-added business activities that the packaged business software should support is an approach that will assist implementation teams in quickly focussing on what is important to the customer.

Alignment is key to being able to produce the desired business results. There are several key alignments that we need to develop and manage as part of the packaged software implementation. First is the alignment between the customer's executives and the project team with regard to project objectives and results. Second is the alignment between Technology (IT, Implementation Partner, Packaged Software Provider) and the Business with regard to the business model and the value-added business results. Third is the organizational alignment between the business model and the packaged software. These alignments cannot be one-time events, but must take place every day during the implementation project. In the next chapter, we will discuss how to effectively create alignment with implementation partners by performing a formal knowledge transfer on the existing business solution.

References

1. Dawson, Ross., *Developing Knowledge-Based Client Relationships*. Butterworth Heinemann, 2000, Page 7.

2. Poppendieck, Mary and Tom., *Lean Software Development: An Agile Toolkit*. Addison-Wesley, 2003, Page 1.

Where to buy this book

You can buy Maximize Your Investment: 10 Key Strategies for Effective Packaged Software Implementations from the Packt Publishing website: `http://www.packtpub.com/maximize-your-investment-10-key-strategies-for-effective-packaged-software-implementations/book`

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com

For More Information:
www.packtpub.com/maximize-your-investment-10-key-strategies-for-effective-packaged-software-implementations/book