



# **How to design and execute an operations improvement strategy through software**

[accendero.com](http://accendero.com)

2235 E. 25<sup>th</sup> St. Suite #190 • Idaho Falls, ID 83404 • [info@accendero.com](mailto:info@accendero.com)



## Summary

Software is often marketed as a key pillar to increase operations efficiency, but the practice of finding efficiency through software is far more difficult than simply making a purchasing decision. This whitepaper presents a walkthrough to design and execute a holistic strategy to increase operations efficiency using software and an agile implementation process.

## Example scenario

The head of operations has been tasked with improving corporate profitability by lowering operations costs. Based on conversations with others on the field, the operations head decides to modernize process management and increase efficiency with software improvements. The modernization bids come back much more expensive than desired and she starts to hear horror stories from colleagues about expensive rollout failures. Concerned that her department will suffer a similar fate, the project grinds to a halt in indecision.

## Definitions

### *I. Operations systems and areas of concern*

**Enterprise resource planning (ERP)** – ERPs are common across many enterprise sectors and handle operations needs common to all. These concerns include logistics, inventory, finance, human resources and project management. Some companies have a single integrated system and some use a variety of single purpose solutions. These systems do not need validation in most quality regimes, but can require it depending on the products being shipped and tracked.

**Process control/Business process management** – Process control management (PCM) or business process management (BPM) systems ensure that data being collected and/or generated by the company about their particular operations is entered, integrated, and stored correctly. These systems are highly specific to a market sector or enterprise use case. They contain proprietary workflows and quality assurance checks as well as integrate with process instrumentation. These systems can integrate with ERP, quality systems and document repositories and enforce standard operating procedures (SOPs) if present in the company. This system must be validated in most quality regimes.

**Document repository** – Software that controls and organizes enterprise documents. It is a required platform and must be validated if the company has an electronic quality management system.

### *II. Software implementation definitions*

**Fully custom** – Software system is completely implemented by enterprise using internal or contract engineering. All cost and implementation risk is assumed by the enterprise. All intellectual property is owned by enterprise. All code is maintained and tracked by enterprise.



**Customizable** – Software is partially implemented by enterprise using internal or contract engineering to customize a publicly available framework. Customization cost and risk is assumed by enterprise. Customized intellectual property is owned by enterprise. Framework intellectual property remains held by framework implementer. Code is maintained as defined by contract between enterprise and implementer. This is a less costly implementation solution for a customized end product, but can be more complex to implement owing to the requirement of a suitable customizable framework as the foundation.

**Enterprise (Single purpose)** – Software is fully implemented by 3rd party. Cost outside the license fee is assumed by 3rd party. Risk is contractually assigned. All intellectual property remains held by 3rd party. All code management is handled by 3<sup>rd</sup> party and not available to enterprise. Software does not intend to solve all enterprise needs for an area of concern instead attempting to be the best-in-class for a specific enterprise need.

**Enterprise (All-in-one)** – Software is fully implemented by 3rd party. Risk is contractually assigned. Software is usually customized to an end user at a negotiated cost, but all intellectual property remains held by the 3rd party. All code management is handled by 3<sup>rd</sup> party and not available to enterprise. Software intends to solve all enterprise needs for an area of concern instead of attempting to be the best-in-class for a specific enterprise need.

## Walkthrough

### 1) Make a list of all of your operational needs.

- In software terminology, these are called requirements. One method to write requirements is to structure the requirement as a sentence of the form: “[user] can [action]” where user is the type of person within the organization and action is their required business process need.
- Be as comprehensive as possible. Strong implementation strategies protect against most “hidden requirements.” Hidden requirements are requirements which are not clear or documented from the project outset and can cause future project issues.

### 2) Assign the requirements to the appropriate operations software system.

ERP	PCM/BPM	Document repository
-----	---------	---------------------

**3) If applicable, designate which requirements require regulatory oversight and might require validation and/or input from the head of compliance.**

**4) Pick an implementation method for each system which will apply to the system requirements.**

- This decision should be based on costs, resources available, technology available and regulatory considerations.

**5) Calculate the 3-year ROI for each requirement based on expected implementation and validation costs. Rank requirements by corporate priority.**

- If priorities are unclear, ROI can be a first stand-in as a good priority.

**Be careful when picking an enterprise all-in-one implementation method!**

- These methods work best when requirements have already been fully developed and have been stable over a long period of time.
- These solutions do not work well when there is the possibility of hidden requirements or requirements are still evolving.

Requirement	System	Implementation cost estimate	Cost savings estimate (3-year)	ROI

**6) Select a set of requirements, implement, validate (if necessary) and roll out.**

- Increasing the number of implemented requirements in a single development round increases the risk for failure. A good practice is to pick smaller, but still useful sets of requirements to implement at one time.
- Because validation carries a cost, there is a tendency to increase the number of implemented requirements to minimize validation costs. These validation costs must be appropriately balanced with the increasing risk of failure as the development round becomes larger and more complex.
- Good planning allows for using modern development techniques, which will significantly decrease the overhead of a software development lifecycle process that includes validation.



**7) Allow the newly implemented system to be used in production for an extended period and assess success.**

- Evaluate the expectations, cost projections and efficiency improvements.
- Revise the table from step 5 (ROI estimates) based on in-use experience.
- Update requirement priorities and chose next implementation target

## **Summary**

By using an agile, evidence-based approach to make improvements, a head of operations can de-risk common implementation failures and ensure cost and efficiency benefits over the long term.

Accendero Software, Inc. can walk you through this process from strategy to design, implementation and validation. Contact us today: [info@accendero.com](mailto:info@accendero.com).

*©Accendero Software, Inc. 2019*