The Template Approach to ERP Implementations:

Why NOW is the Time

January 2017

Inside you'll discover:

- what the template approach is and how it works
- the benefits of the template approach
- how the template approach compares to a traditional ERP Implementation
- which companies are good candidates for this approach and which are NOT
- how this approach makes an ERP system more affordable to mid-market companies
- why, even with a detailed template approach roadmap, many consulting companies would be unable to lead a template-style implementation

In The Know
Solutions Group

Process first. Technology second.

ITK Solutions Group | www.itksolutionsgroup.com

Table of Contents

In The Know
Solutions Group

Abstract

The Template Approach to ERP Implementations:

Why NOW is the time

Early versions of a Template Approach failed. It was ahead of its time. But there has been an inflection point where the right technology and standardized best practice processes have given rise to a legitimate alternative to the traditional ERP Implementation. That is why we think you may be interested in our template approach. You can experience similar results we've seen with other clients if you meet the qualifications of a company well-suited to this approach.

In this white paper you will find out:

- what the template approach is and how it works
- the benefits of the template approach
- how the template approach compares to a traditional ERP Implementation
- which companies are good candidates for this approach and which are NOT
- how this approach makes an ERP system more affordable to mid-market companies
- why, even with a detailed template approach roadmap, many consulting companies would be unable to lead a template-style implementation

In The Know
Solutions Group

Introduction

INTRODUCTION

The Template Approach to ERP Implementations:

Why NOW is the time

The traditional ERP Implementation is not dead but it now has some serious competition. The accepted way of approaching ERP Implementations is being turned on its head thanks to the modern template approach. Here is what we used to "know" about typical ERP Implementations.

- large, complex projects
- expensive and time consuming
- only accessible to mid-market to enterprise-size companies due to the expense
- lengthy discovery process to understand how things are currently being done

You are a good candidate for this approach if…

- Small to midrange companies ($100 million-$2 billion in revenue)
- Have own processes (but don't need to, standard processes usually work)
- Don't know the best practices (and so would get a big benefit from learning them, ex: entrepreneurial company, old workforce, workforce without much turnover)
- Large companies for certain processes (accounts payable, core financials, etc). These processes should not be unique
- Retail, Telco, Utilities and Distribution

You are NOT a Good Candidate if…

- A company that is cutting edge (with unique processes to go with it)
- Highly technical businesses with set, custom processes
- Highly complex conglomerates (Fortune 50)
- The internal Culture of your company is not self-directed (need hand holding, have difficulty tracing processes independently)

Why a template approach to an Enterprise Resource Planning (ERP) project?

- Reduction of risk. Starting with proven best practices for each process reduces complexity, cost, time to final solution and accounting risks from the adoption of atypical business processes
- Reduced project timeline and cost
- Faster adoption of more efficient processes
- Better project Return on Investment (less cost + earlier adoption = higher realization of project financial objectives)

Background

BACKGROUND

The Template Approach to ERP Implementations:

Why NOW is the time

The problems that the templated approach seeks to address are:

- Too much time and energy (and in turn money) is spent on a discovery process that is not always needed at the level of detail of a traditional ERP Discovery
- Some elements of an implementation should be done according to industry best practices (an industry template) leaving time and energy to focus on areas that are unique to your company
- The implementation itself is too time consuming
- Blue sky requirements allow companies to dream but why reinvent the wheel when battle test processes yield outstanding results
- Not all companies can afford an ERP System

Why early versions of the Templated Approach Failed

The problem was that ***in the early days there were no universally adopted software tools or accepted ways of doing business processes***. Below are some other issues.

- No established (or documented) Industry best practices
- Early approaches were too rigid.
- Preconfigured systems used to mean one size fits all industries (but no one well)
- Problems occur when you change how someone expects it to be done and how you ask them to do it
- No one wants to be first, the pioneer

The business world is littered with early pioneers with breakthrough ideas that failed. They came before the inflection point, where key technologies and accepted behaviors made the breakthrough possible.

The key inflection points:
1) industry processes or best practices established
2) Verticalization. The consulting industry became staffed with industry professionals (like ITK's approach) which lead to an understanding of best practices by industry
3) Natural Selection. Time has proven the importance of specific software solutions. For example, Retail companies have specific needs like store reporting and clusters by region. For government, budgeting is critical.

The Traditional ERP Implementation

The **traditional enterprise ERP implementation** follows these basic steps using the Microsoft Sure Step terms for these phases.

To understand the importance of the template approach, it's crucial to understand what a traditional ERP Implementation looks like.

While other methodologies use different terms for these phases, the activities that occur in each phase are generally the same.

- Discovery (requirements gathering).
  - Review legacy processes
  - Gather requirements on new system.
- Analysis
  - Train core team on software (basic understanding, to support the implementation team and many key client setup/process decisions).
  - Set up test system for modelling of different configurations (this is working through a prototype of different processes with varying configurations to best satisfy requirements).
  - Classify requirements as 'fit' (new software does this) or 'gap' (new software does not do this).
  - Some new requirements will continue to be added in this phase as more detail is understood.
  - Decide what data will be migrated and what the strategy will be to migrate data (how it will be accomplished, methodology for clean-up).
- Design
  - Complete modelling including future process flows.
  - Finalize future system configurations.
  - How will we handle gaps?
    - What is the workaround (including stop doing whatever it is, so everything has a workaround)? Is it acceptable?
    - Should we customize?
  - For customizations and integrations, write FDDs (functional design documents) and TDDs (technical design documents).
  - Begin testing data migrations.
  - If any customization code is required to accomplish the data migration, the FDDs and TDDs should be completed in this phase

In The Know
Solutions Group

The Traditional ERP Implementation

- Development
  - Complete the software code for FDDs and TDDs, including testing and acceptance.
  - Continue testing data migrations.
    - If any customization code is required to accomplish the data migration, that development should be completed in this phase.
  - Test completely configured solution, including customizations (this is usually several rounds of testing with different objectives).
  - Stage production hardware.
  - Map cutover strategy and plan, including timing of data migrations.
- Deployment
  - Train end users on the software.
  - Conduct final user acceptance testing.
  - Perform cutover plan activities and 'go live' (begin using the new system and stop using the old system).

Solution

THE SOLUTION

The Template Approach to ERP Implementations:

Why NOW is the time

The central premise of the template approach is to use standardized business processes for most areas of your business and to give special attention to key areas of your business that are unique.

That means not giving extra time or energy to the elements that are already delivering outstanding outcomes in your industry. Fans of the 80/20 (or Pareto) Principle will see that we are focusing our energy on the 20% of processes that will deliver the most value to your company. Which means that the 80% of processes that are standardized are given the time and energy they are due, but no more. Let's take a closer look at how this works in reality.

A **template approach** takes the phases from a typical implementation and eliminates the steps in red. In addition to the steps in red being eliminated, we expect the steps in orange to be reduced. Why remove the steps in red? Because spending time on legacy processes that will be replaced with best practice standard processes is a waste of time. The steps in orange, because of standardization, will only require a fraction of the time normally allotted in an ordinary ERP Implementation.

- Discovery (requirements gathering).
    - Review legacy processes
    - Gather requirements on new system.

- Analysis
    - Train core team on software (basic understanding, to support the implementation team and may key client setup/process decisions).
    - Set up test system for modelling of different configurations (this is working through a prototype of different processes with varying configurations to best satisfy requirements).
    - Classify requirements as 'fit' (new software does this) or 'gap' (new software does not do this).
    - Some new requirements will continue to be added in this phase as more detail is understood.
    - Decide what data will be migrated and what the strategy will be to migrate data (how it will be accomplished, methodology for clean-up).

- Design
    - Complete modelling including future process flows.
    - Finalize future system configurations.
    - How will we handle gaps?
        - What is the workaround (including stop doing whatever it is, so everything has a workaround)? Is it acceptable?
        - Should we customize?
    - For customizations and integrations, write FDDs (functional design documents) and TDDs (technical design documents).
    - Begin testing data migrations.
        - If any customization code is required to accomplish the data migration, the FDDs and TDDs should be completed in this phase

- Development
    - Complete the software code for FDDs and TDDs, including testing and acceptance.
    - Continue testing data migrations.
        - If any customization code is required to accomplish the data migration, that development should be completed in this phase.
    - Test completely configured solution, including customizations (this is usually several rounds of testing with different objectives).
    - Stage production hardware.
    - Map cutover strategy and plan, including timing of data migrations.
- Deployment
    - Train end users on the software.
    - Conduct final user acceptance testing.
    - Perform cutover plan activities and 'go live' (begin using the new system and stop using the old system).

What about the template approach allows the removals and reductions?

First, let's consider the elimination of the requirements gathering and legacy process review portions of the engagement. We want to understand only enough about the legacy processes and system as it pertains to the new system. We will enter the engagement with suggested best practices process flows for each area of the business already mapped in the future software. Rather than start with a generic set of training companies and training data, we will stage the client test system with a blank database to which a sample test company has been added and sample configurations completed according to best practices for the client industry. This is called 'pre-configurations' and means we have begun setting up the client software prior to arriving on site. These configurations are changeable and open for discussion, but if we feel we know what some of these configurations are or should be, it jump-starts the implementation process.

THE SOLUTION

The Template Approach to ERP Implementations:

Why NOW is the time

Note that all the above requires senior consulting skills in both the software and business process knowledge.

We will begin the engagement with a series of workshops during which the client will be trained on each area of the software as we discuss the suggested future process flows and what might be adapted to client preferences with configuration. We will have a mindset during this portion of the engagement that the client will adapt to the software and that software may be adapted to the client if it can be done with configurations (and not customizations). There will be gaps that still arise, inevitably (things the software does not do that some folks at the client think that it should), but this mindset should lead to fewer gaps than a more wide-open requirements process because we have an adaptation mindset.

If you plan to adapt, you must know how to do it: the procedures to adapt and configurations to adapt. Further, the consultant must be comfortable with conflict and disagreement, defusing interpersonal discussions and providing leadership to work through processes and obstacles. It will be most effective for everyone to know they must end with the best solution with the least customizations. Fewer gaps means less customizations, less development, less testing and a less complex solution that still satisfies 80-90% of the company's objectives. This should yield a superior ROI: the reason the solution was selected and purchased should be fulfilled while the cost of the solution (and future maintenance) controlled.

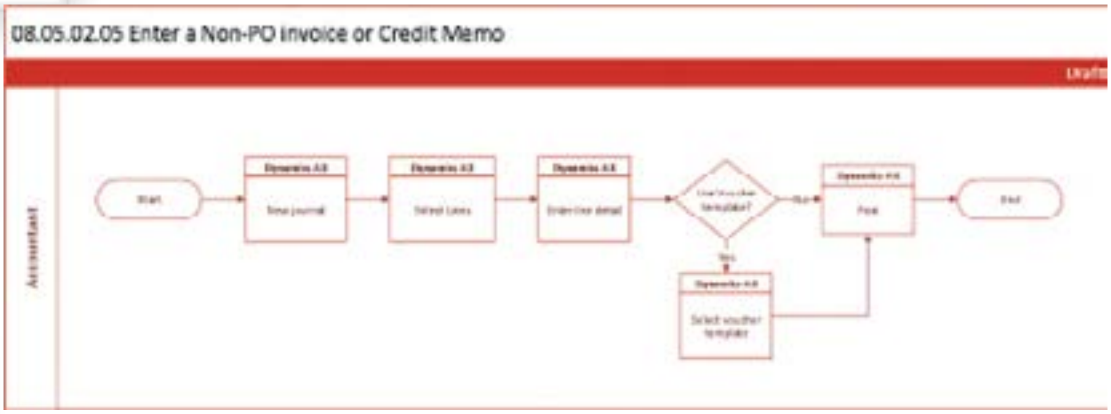
How do you run a template approach? Let's examine a sample process to walk through how the implementation might work. A simple process, such as non-purchase order (non-PO) accounts payable invoices would be a nice starting point. So, before we start working with the client, we would have a fully operational shell setup for the client in a test system. We would have run through a few non-PO AP invoices to make sure the system is working correctly. We would have the basic suggested process flow in hand, such as the following (process developed for Microsoft Dynamics AX, but would be similar in any software):

THE SOLUTION

The Template Approach to ERP Implementations:

Why NOW is the time

Process Overview: *Invoices not initiated from a purchase order and not related to inventory items (such as utilities or rent) may be entered using this process. This process can also be used to record credit memos not related to a purchase order or to inventory items.*



What would we do in the template workshop meetings between the client and the consulting team?

1. Train the client on standard software accounts payable settings and processing. A general session on this part of the software would usually take place prior to reviewing all the processes that fall under this functional business umbrella.
2. In a session specific to the above example 'Non-PO AP Invoice Process', review the above process and process flow together.
3. Discuss the client business use cases for this process (business scenarios, where we expect to use this process).
    a. Reviewing each business case, what are details we need to capture for each case? Examples: method of payment override, multiple cost centers to charge, etc.
    b. Should we add any boxes to the process flow or remove any?
    c. Is there more information that we need about any of the use cases?
    d. Are there any use cases where the process does not work?
4. At the end of the meeting for this process, we may have adjusted the process and we may have completed the work expected for this process or need to have a follow up meeting to finish discussing the use cases or issues with the process.

Once a process is initially agreed upon, what next?

The consulting team (perhaps with client team assistance) will tweak the configurations of the test system and work through the process in the test system. Making an instructional document of the process in the system to support the process flow is extremely useful at this time. This allows the members of the team who are less confident (yet) on the new software to trace the process and agree with its completeness and satisfaction. This document is not an end user training document (as it contains thin instructions and assumes some software knowledge), but it will be the basis for a richer document to be prepared by the client training team later. This sample process document (which our company calls a 'Business Process Walkthrough' or 'BPW') might look as follows. Again, the sample system here is Microsoft Dynamics AX, though the template implementation process should be product agnostic.

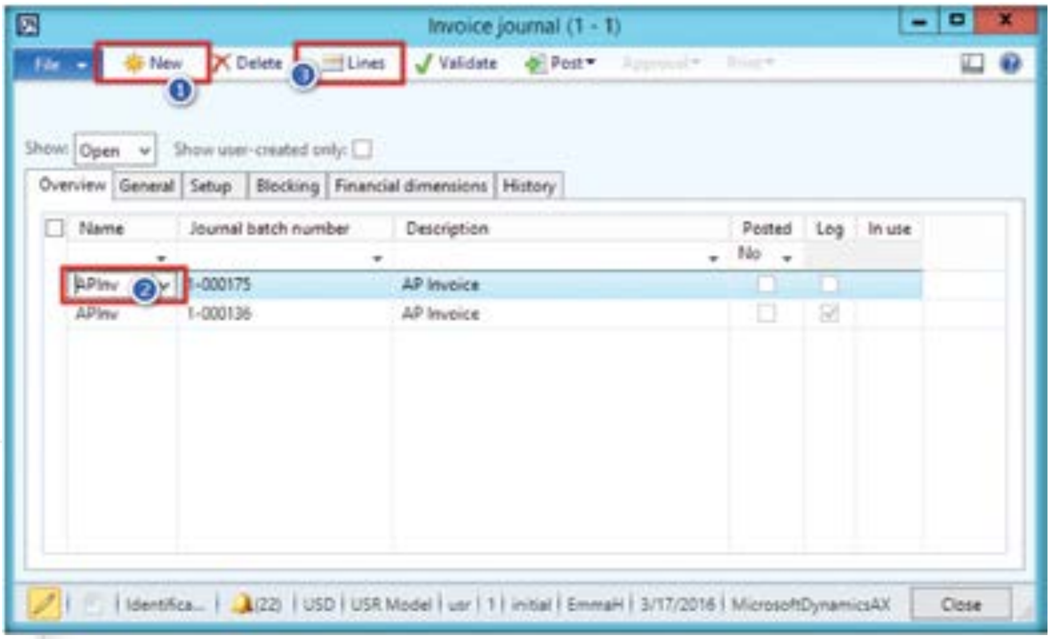
Future Process steps in MS Dynamics AX:

Begin Sample Business Process Walkthrough

CREATE INVOICE JOURNALS

Invoice Journals are to be used when simply entering an invoice for payment that is not related to a PO.

1. Go to Accounts Payable > Journals > Invoices > Invoice Journals
2. 'New' to create a new Journal and enter logical description Ex. TX Power Co - MM/DD/YYY.
3. Click the Lines Button.



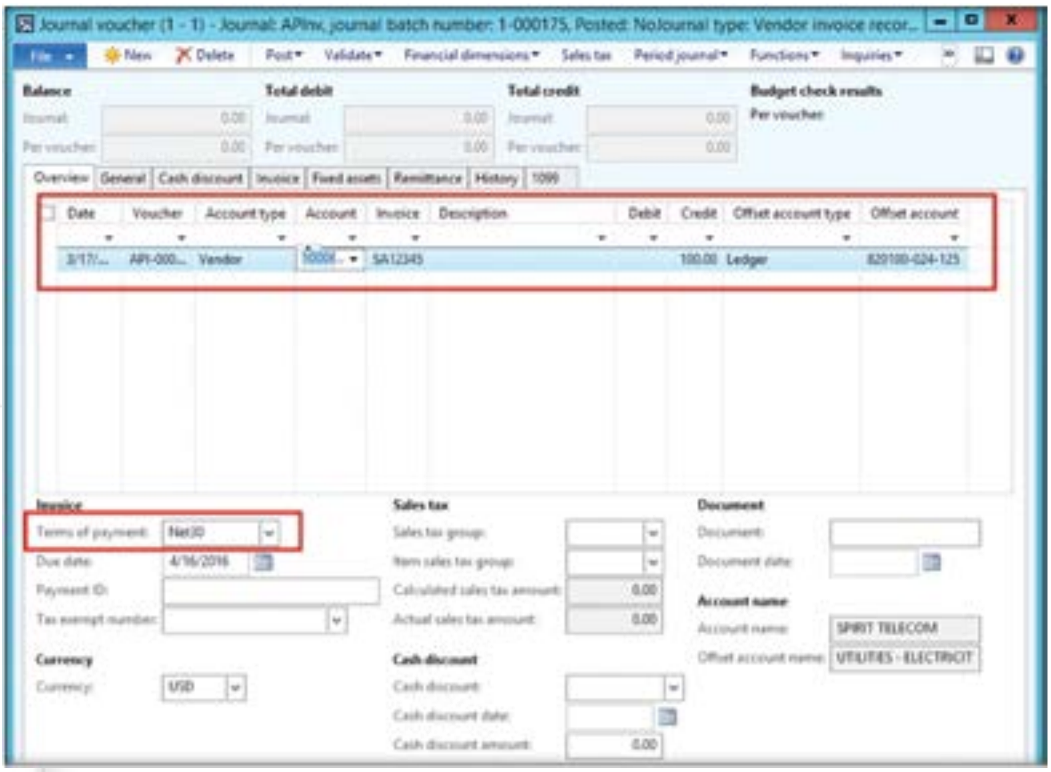
In The Know Solutions Group

Sample Business Process Walkthrough

Voucher Template

Use voucher template to populate lines based on prior entries (dollars or percent), otherwise fill out the invoice as shown.

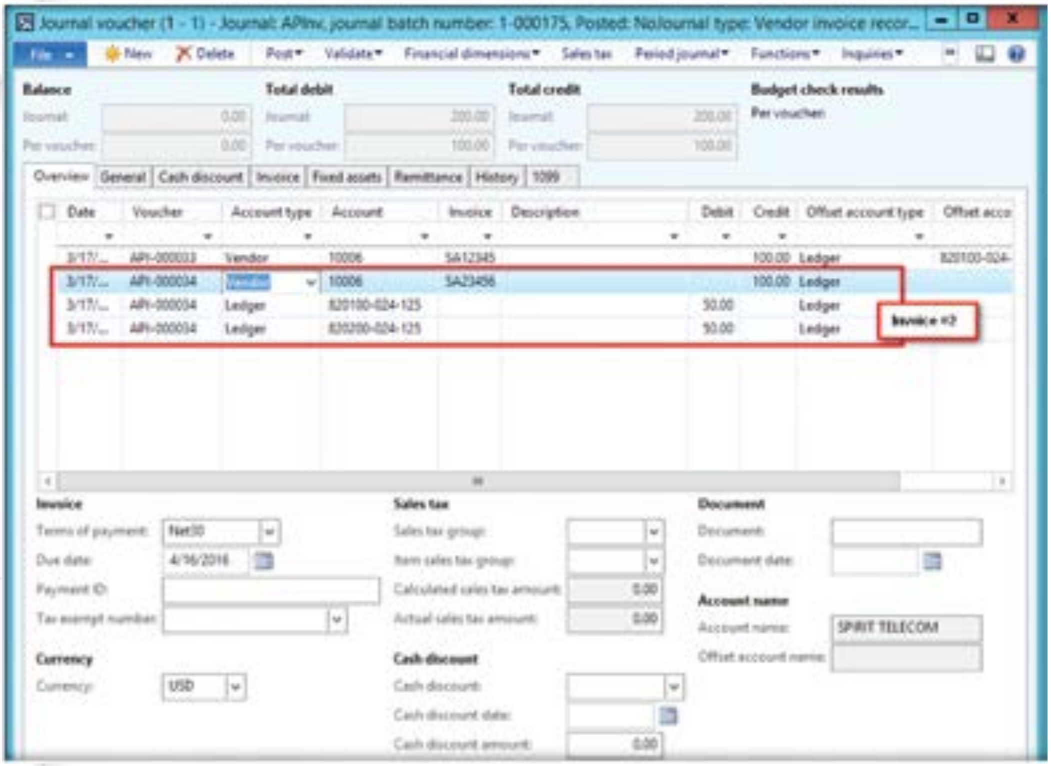
4. Enter Date, Account Type, Vendor Account, Invoice number and Description text.
5. Enter a credit amount.
6. An offset account number can be entered if invoice is to be expenses to only one account.



Sample Business Process Walkthrough

Terms of payment should default from the vendor (which also defaults due date). Method of payment should also default from the vendor.

7. If there are multiple accounts to be debited, you can also simply 'arrow down' to create a new line on the form (as shown below):



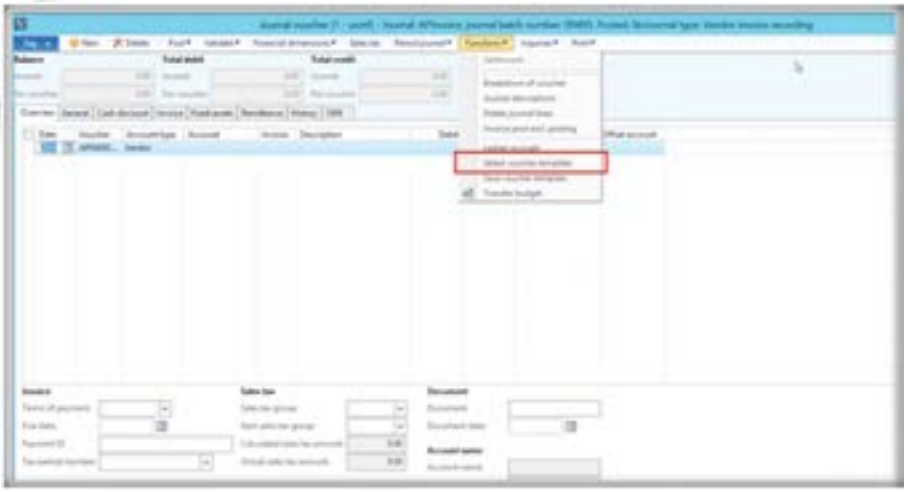
8. When finished entering invoices, click the 'Post' button and select 'Post'.

Sample Business Process Walkthrough

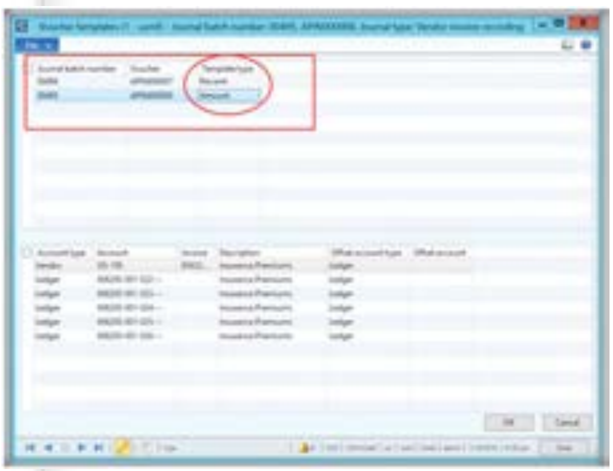
VOUCHER TEMPLATE FOR INVOICE JOURNAL (AMOUNT)

After creating the journal batch and opening the lines form

1. Choose Functions > Select voucher template



2. Select from the list of existing voucher templates. There are 2 types of templates possible. Percent and Amount. Selecting **Amount** template type will use the same amounts the template was created with.



3. The journal lines will populate. Update the date fields and add an invoice number on the Vendor line.

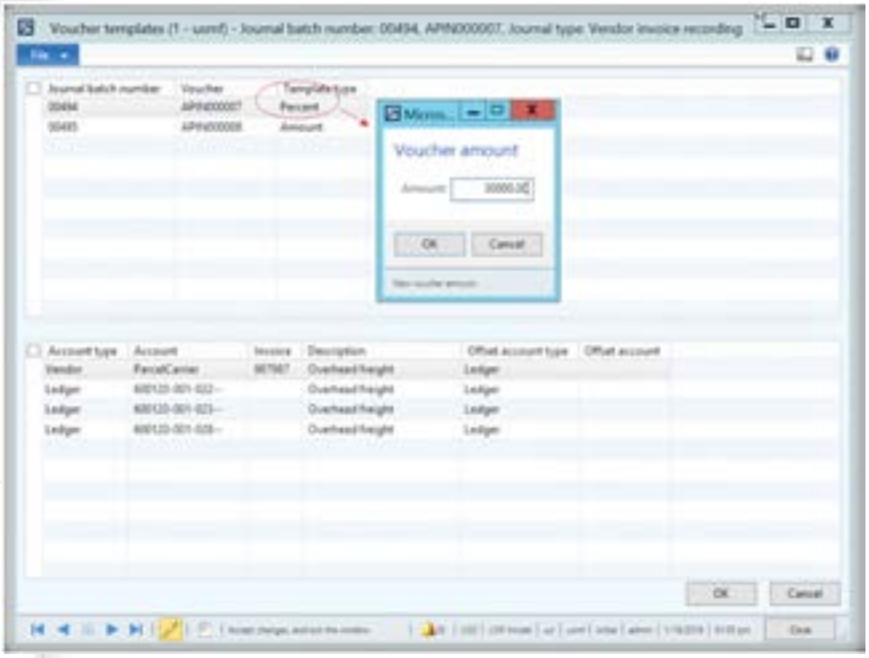
4. Post the journal.

Sample Business Process Walkthrough

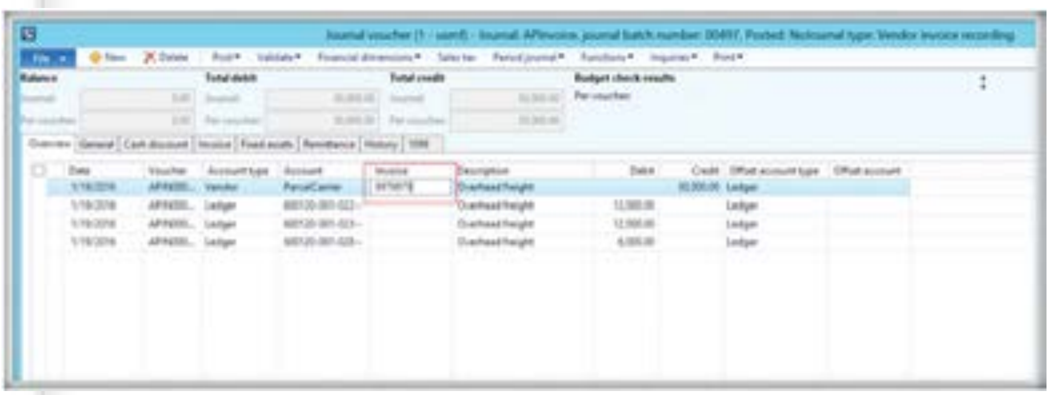
VOUCHER TEMPLATE FOR INVOICE JOURNAL (PERCENT)

After creating the journal batch and opening the lines form

1. Create a new invoice journal. Open Lines and Choose Functions > Select voucher template. Choose a Percent type template.
2. Enter the total amount of the payment. AX will distribute the amount based upon percentages from the original voucher template.



3. Enter the date and invoice number.



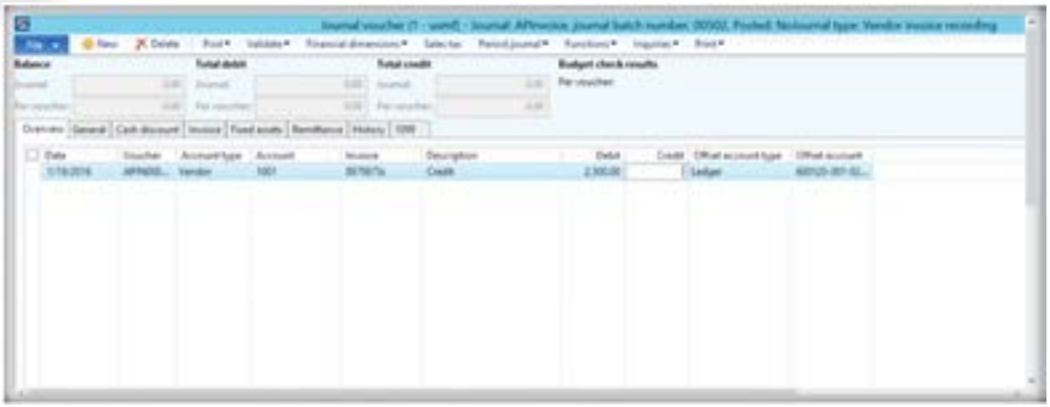
4. Post the journal

Sample Business Process Walkthrough

CREATE A CREDIT MEMO USING INVOICE JOURNAL

In the event that you need to create a Non-PO credit memo you would create an invoice journal as normal but

1. When entering the line item(s) you would Debit the Vendor account and credit the offset account(s).
2. Post the journal



What happens next?

As all the processes for the company are mapped, reviewed, adjusted and completed, the client test system will be built and we will have sample processes walkthroughs for every client process. If there is development planned, that will still take place (though it should be reduced from our template process and adaptation mindset). We should adjust the standard BPWs for customization steps, so the business has a full and complete flow for each process, including customizations.

Testing phases and beyond.

- The BPWs fit nicely together into an end-to-end testing plan, facilitating a thorough testing plan. Our client should be very familiar with the BPW format and content, so should be able to conduct most testing on their own with the consulting organization in a support role. This is a natural part of the education and turnover process.
- We would still run the usual performance testing cycles, which are separate from our template methodology (and not optional).
- The client go live (software turnover) would be conducted as with any other project.
- The data migration and cutover plans would be conducted as normal.

In summary, while we have fast-tracked the implementation, reducing investment and time without increasing risk, once the project enters the Development phase, we are back in line with standard project implementation methodology.

Conclusion

CONCLUSION:

The Template Approach to ERP Implementations:

Why NOW is the time

Our mission with this white paper is to go into detail as to: how the template approach compares to a traditional ERP Implementation; which companies are good candidates for this approach and which are NOT; how it makes an ERP system affordable to mid-market companies; the benefits of the templated approach and what the templated approach is and how it works.

Here is a summary of the results you can expect by using the Templated Approach with a team like ITK:

- Reduction of risk by starting with proven best practices
- Reduced complexity, cost, time to final solution
- Reduces accounting risks that occur when using atypical business processes
- Costs savings means the costs of an ERP system is no longer out of reach for smaller companies
- Shrinks project timeline and in turn cost
- Faster adoption of more efficient processes
- Improved project Return on Investment

Why is it important to have the right services partner for this approach?

The templated approach is predicated on the notion that you must know which areas to focus your attention on and which areas not to focus on.

You might be asking yourself why we've gone into such great detail about how we structure our template approach implementation. Couldn't another consulting firm just copy it? The answer is 'no' and it gets at the heart of the templated approach.

Not every consulting company can deliver this. This approach requires senior consulting skills both from a software perspective and a business process perspective. This approach is essentially taking a scalpel to the traditional ERP Implementation. When scalpels are involved you will likely agree that it should be in the hands of a capable practitioner. The results can be profound financially, from a time perspective and speed to value. But you must know what processes can be standardized and which ones must not be. If you remove a process that is crucial, or leave in something in that is not crucial, it will have a much higher leverage on the overall implementation simply because with fewer items, each item exerts a greater influence on the outcome.

In summary, your selected partner must:

- Understand, on a deep level, both the software and the processes the software is there to support
- Understand best practices for key industries like yours
- Know how to leverage software to solve business problems that goes way beyond best practices
- Have an understanding of business processes and software gained from experience using these systems in the real world, not just from implementing software

  (Note that the above is a unique combination of skills that are difficult to find in the consulting community.)

What Comes Next…

We covered all of these areas outlined above, but likely raised questions of your own. We welcome the opportunity to speak with you about how this approach can help your company in particular. Please don't hesitate to contact me, the author, Susan Alvarez, at salvarez@itksolutionsgroup.com