



**The 6 Phases
of *Any*
Business
Software
Implementation**

By Art Olsen, CPA
Solution Architect at PC Bennett Consulting, Inc.

Phase 1 | Discovery & Planning

Goal: To thoroughly understand the business' needs

Discovery and planning is the first stage, lasting up to roughly 4 weeks, where the main goal is for the implementation team to get to know the business and its processes. Discovery starts during the sales cycle (if it is done right) and discovery and planning continues on after the sale is made, as the implementation team works with the project team to create a solid project plan – the end goal of this phase of the software implementation project.

This involves looking at a company's corporate structure, lines of business, departments, product lines, etc. The team will also dive into the business processes such as quote to cash, procure to pay, cash management, and period closing.

5 Steps for a Successful Discovery

Step 1: Establish the Project Team

- The project team should represent all aspects of the business. There should be someone from Sales, Customer Service, Purchasing, Materials, Operations, Accounting, and IT.
- Team members should have strong knowledge of current procedures in their area and be active system users with hands on experience, knowing what the current system does as well as its limitations.
- The team leader should be someone who is higher up in the organization chain – having an understanding of how the business operates and where the business is heading. Often times an business management software project can be interpreted by others as just an IT project, particularly if a person with an IT background is the team leader. Watch out for this. It must be seen as a project that involves and affects the entire business – not just IT.

"The best project team I have worked with was at a company that had a retail store, an online store and a production facility that built products for stock and for specific customer orders. This was a fairly complex system with two systems involved and a custom application to hook them together.

At the kickoff meeting the President started the meeting by explaining to the project team members that they had been selected because they were the best from each department and as such, were being entrusted to figure out the best way to implement the system. The project team leader was the Controller and he took full ownership of the project and did an excellent job of facilitating the work. As she concluded her remarks at the kickoff meeting, the President transferred authority to the team to complete the project and asked for frequent updates on the status. She stayed actively involved at a high level making sure the team had the resource and bandwidth to complete the task."

Step 2: Hold Discovery Meetings

This is where members of the implementation team interview members of the project team to get an in-depth understanding of various parts of the business at detail level. The team will want to know what are you currently doing, and what you want to do in the future? They will discuss current processes thoroughly and identify pain points with the current system.

Step 3: Document Key Processes and Requirements

Overall, this round of documentation will become the foundation for configuring the new system and also helps the implementation team with building out the project plan.

- Meetings with the implementation team and the project team, as well as executive members – looking for red flags, or gaps between the business' needs and system capabilities. These holes can greatly affect the requirements of the new system and the better they are understood, the better job the implementation team can do at guiding the project.
- Discussing broader strategy with the executive team. For instance, if the company is planning on opening 30 new stores in the US, there are most likely important implications on how the new system should be set up.

Step 4: Identify Potential Gaps, Risks, and Solutions

Continuing from the discussions in the previous step, the implementation team will work to:

- Identify all of the areas where the current system doesn't meet the company's needs
- Based on those findings, begin formulating potential solutions for the new system. Areas that require workarounds outside the system like extensive spreadsheets are prime candidates.
- Identify any risks that might push out the implementation date or that make the conversion more complex. Risks to identify might include things like complex chart of accounts, inaccurate financial records and operational data, significant planned absences of project team members during the implementation and so on.

"In my experience, if you find a system that addresses 90% of your needs out of the box you have made a good selection. The balance of those needs can be addressed by adapting your procedures to the way the software works, customizing the software or a combination of the two. Systems like Acumatica are highly customizable. Arming the implementation team with the knowledge of your company directions and needs will arm them to guide the project team through selecting the best approach to address the remaining 10%."

Step 5: Build the Project Plan

The implementation project plan will serve as a road map for the implementation. Depending on the project it may be fairly detailed or it may be higher level. It will include information from the last three steps and evolve as you work through the upcoming Design and Development processes. Think of it as a sort of backlog list of tasks used to guide the implementation team, the development team and the project team. It is a communication tool for all involved to have visibility of open items and priorities.

The bottom line for the discovery process is that the implementation team and project team must both understand and agree on the business processes and the objectives of the project. The project plan is the end goal of the discovery process and it is the foundation of the rest of the project.

Phase 2 | Design & Configuration

Goal: To translate documented requirements into real configurations

The design stage is the part of the implementation where the focus is on creating a system configuration that will maximize the system benefits and ROI received by the company and their personnel. The implementation team will work to translate the client's system requirements into potential configuration options. Then, they will work closely with the project team to establish those configurations and implementation strategies that will then lead into the development phase of the project.

6 Steps of Implementation Design

Step 1: Gather and Review Master Records

Early in the design phase the implementation team will want to review the client's master records. This is data that is the foundation of the current system such as:

- Chart of accounts
- Subaccounts if applicable
- Customer master
- Vendor master
- Item master

This information along with the discovery information will form the basis for the implementation team to begin discussions with the project team about how to start configuring the new system.

Step 2: System Orientation & Walkthrough

This usually follows the flow of various business cycles and starts the process of mapping out the procedures identified during discovery. The training and walkthrough can take a few hours to a couple of days depending on the complexity of the business.

As an example, we might open the accounts payable module and walk through setting up a vendor (contact info, payment methods, terms, etc.), entering an accounts payable invoice, showing the approval processes and the payment cycle steps. This allows the project team to see how they would do their jobs in the new system.

"A few years ago during the discovery process I asked a project team member to explain their current steps to generate an invoice for product shipments. She smiled and said, "Don't laugh. We get the shipping documents from the shipping department, then take a 7 part invoice and insert it into a typewriter and type the invoice", all the while watching the incredulous look on my face. She went on to explain where the copies went and that they then entered all of the invoice information onto a 13 column pad. She concluded by saying that they had been using this procedure for 25 years and were looking forward to changing their approach. They were very excited to see how the new system would eliminate the typewriter and 13 column pad and they went on to a successful implementation."

Step 3: Establish Initial System Configuration Settings

Establishment of the initial configuration settings includes tasks like designing the chart of accounts in combination with the subaccount structure. This is a key decision point and should be done with the goal of supporting the reporting requirements that arose from the discovery process. Done correctly the proper configuration can support things like companies, divisions, departments, product lines and so on.

This is also the time to decide what the format of customer and vendor id's – should they be simple numeric values, abbreviations for the names or some combination thereof? Advanced software systems like Acumatica and should support both to give you full flexibility in running your business.

If the client is a distributor and/or manufacturer, design of the item numbering scheme and bill of materials and resources is also critical.

Step 4: Create a Prototype

The teams will work together to build a prototype system configured to meet the client's business needs. The system will consist of sample customers, vendors, items that feed financial activity into the accounting system using the first pass at a chart of accounts and subaccounts. Again the teams should look to see that the client's reporting requirements can be met. This version of the system will be used to train the project team members and prove out some of the decisions made earlier in the design process.

Step 5: Define User Roles

Modern systems rely on user roles to control security. These role-based systems can be simple or complex depending on the size of the business and internal control requirements. For example, a sales person should not have access to the cash receipts function, or an AP person to be able to cut a purchase order. The project team will define these roles and users who will be using those roles. This especially applies to Acumatica, which has particularly sophisticated security functionality.

Step 6: Document System Procedures

Along the way the project team will document how the system should be used by creating procedures for each function. The procedures should include which position submits the initial invoice, how it is reviewed and how it is approved. The project team puts these types of procedures in place to support initial end-user training and the training of new people in a consistent manner. This is probably the most difficult part of the design process for most project teams. Some more advanced systems, such as Acumatica, allow client-specific procedures to be easily incorporated into the "Help" system as part of the user interface so that they are always a click away for the user's reference thereby allowing for the consistent use of the system.

"I have been brought into numerous companies to review how they were using their current system after a few years and frustration levels with it were very high. In all cases there had been significant turnover and many if not all of the original project team had left the company. In many cases users are shown about 10% of the functionality of the system. They get frustrated with it and start building workarounds usually in spreadsheets. Having current and readily accessible procedures helps reduce this problem."

Phase 3 | System Development

Goal: To prepare the new system environment for data migration

Like any development project, development does not begin with actual development work. Each member of our software development team has a wise quote on the back of their business cards. It reads, "First, solve the problem. Then, write the code." Development must not begin until proper planning and documentation has been done. If not, a severe waste of time and energy is sure to follow. This is why the software development process begins only after the Discovery and Design phases have first paved the way.

The same is true for business software implementation projects – Discovery and Design should be complete before the Development starts.

What's in the Development Phase?

Step 1: Customization

Modern business management systems have a great amount of flexibility built into them and there can be a variety of approaches that are possible to address the way customers want to run their businesses. Sometimes during the Discovery and Design phases the implementation team in coordination with the project team determine that some level of customization of the new software is required to best meet the organization's needs. Here, the implementation team will build, deploy, and test any required features that were identified during the design phase. This includes modifying or building new screens, developing new processes that fit the system, writing new reports, and building system connections.

Step 2: Configuration of the Go-live System

All of the decisions made regarding needed configurations in the design phase get deployed here. This is where the implementation team will load the chart of accounts, as well as any static data such as the customer master, item master, vendor master, historical financial activity, etc... Other configurations such as formatting the chart of accounts and sub accounts, as well as setting any numeric sequences for customers and vendors will also be done. Customer classes, vendor classes, item classes and deferral codes for deferred revenue situations are also examples of areas that will be set up during the development phase.

Step 3: Simulation of a Live Environment

Now that a lot of the data has been added to the system, it's time to run a conference room pilot – walking through the entire order to cash cycle for the first time in the new system. This preliminary simulation of the live environment is important in providing another opportunity for the implementation team to test the design of the system to see that it meets the business requirements.

- Offers a great deal of training value for the project team, risk-free
- Have everyone agree that the system has been configured in a way they want it to run the business.
- The project team should understand how they will use it to do their jobs once it goes live.

The implementation team in conjunction with the project team will look at the results of the conference room pilot testing to determine whether the system is ready for go-live.

Step 4: Development of End User Training

Now that the approach to using the system has been developed the project team should create training material for the end users. The members of the project team should be subject matter experts in their area of the business and should prepare the training materials for their areas. The procedures developed earlier should be the foundation for the training so that the users learn to use the system in the way the project team designed it to be used.

Digging Deeper with CTO, Chris Bennett

In designing a new system, or new integrations to a current system, it can be tempting to just start building out the new code. Often people forget that the most important stage of any software development project is building a detailed design document. PC Bennett strongly believes that a successful development project is entirely dependent upon a thorough and complete design that has been reviewed, revised, and approved by the end-users and major stake-holders of the software.

A successful design will include:

1. A screen mockup and complete description of every major screen in the system, including a description of each field on the screen and its purpose.
2. A complete description of every major process in the system (flow of data, calculations, posting processes, etc.) with diagrams (such as flowcharts) as appropriate.
3. A report mockup and complete description of every report to be built for the system, detailing the source of the data to be included, filtering, sorting, and subtotalling options.
4. A detailed description of the technology to be utilized.
5. A complete and detailed field-by-field table specification for each new table to be built for the system.
6. A list of existing table "objects" to be utilized or to be integrated with, along with a field-by-field list of custom fields to be added to these objects (note, a table "object" may refer to a virtual table entity from the existing application's perspective; this entity may be comprised of one or more actual SQL tables).
7. A data dictionary tracking all of the data fields to be used throughout the system (mapped to the tables using them).
8. A chart of table relationships (showing key field links between them).
9. Data Flow Diagrams as appropriate to explain how data "moves" through the software system.
10. A description of the user-security methodology to be utilized.
11. List of anything that will NOT be included in the system (so that there are no invalid assumptions)."

Phase 4 | Testing

Goal: To confirm that system functionality aligns with requirements

This is where the organization will really start hammering on the new system. The testing phase as part of a business software implementation will follow after development, where most of the configurations and customizations were made. End users also began training in a special training environment with sample data. There's no defined boundary between development and testing. In fact, there will be significant overlap between the two throughout the process. To review the development phase and other phases in the implementation process, more articles in this series can be found [here](#).

5 Crucial Points for Testing a New System

Step 1: Execute User Acceptance Testing

Developing and executing the UAT plan will largely fall on the shoulders of the project team because they know the business. In order for the testing process to be successful, the users have to define what it is they want to see, and then develop a testing plan to the level of comfort need to accept the product as deployed. The implementation team will work with the project team to develop the testing scenarios to ensure all parts of the system are tested.

Step 2: Import Sample Data

Part of the go-live process involves loading static and dynamic data. Static data are elements that don't change frequently such as customers, vendors, inventory items and so on. Dynamic data changes frequently and includes things like accounts payable and receivable invoices and inventory quantities.

Often it makes sense to export that data out of the old system and import it into the new system. Part of testing should include the export and import processes. Particularly with dynamic data you will want to ensure that process works efficiently so as to minimize system downtime during conversion.

Importing some static and dynamic data enables the project team to perform system testing with familiar information and better enables simulating running the business in the new system.

Step 3: Adjust Configurations

During the "testing" phase the teams may determine that some slight adjustments are necessary to some configurations in order to optimize the use of the system in running the business. If adjustments are required, they are made in both the test and go-live systems.

Step 4: Establish "Cut-off" Strategy

Transitioning from the old system to the new system can take a variety of paths. In the past, companies converted systems one module at a time, or they would run both systems in parallel. The problem with those approaches is the workload for users more than doubles. They have to build interfaces to get the systems to communicate or work in both systems, and reconcile differences.

The typical approach now is to build confidence in the new system and user training through thorough testing, and then cut over all at once. At this point in the process the teams should have a full understanding of how to do the conversion. They should put in place a plan to finish training other users and bring over the static and dynamic data.

"I worked with a well-funded start-up who had about \$1,000,000 in payables just prior to go-live. The controller concluded that it would be much easier to just pay off all the open invoices rather than spend the time on data import and validation, and it certainly was. I have been involved in over 150 implementations and this was the only time a company took this approach."

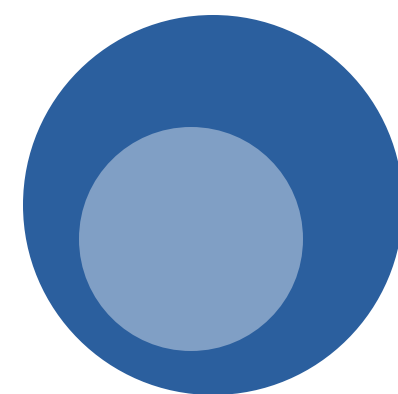
Step 5: Simulate Running the Business

Just prior to the go-live decision, the teams should work jointly to perform one final test of the system, sometimes referred to as a conference room pilot. The project team should prepare a final test scenario that simulates running the entire business in the new system. Hopefully this will either confirm that the team and the system is ready for the final step – going live.

Step 6: Deliver End User Training

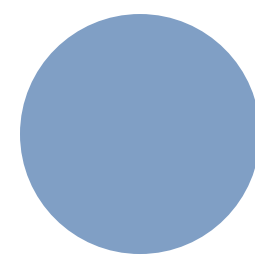
Once the confirmation has been made by the project team, members fan out to train the end users in their respective departments. I've found that it's helpful to set up a designated training environment where there are a few workstations, a whiteboard, and projector for employees to practice and receive training instruction.

Phase 5 | Deployment



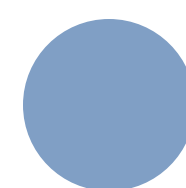
Goal: Go-live

This is where we pull the trigger and migrate to the new system. Previously I wrote about the processes leading up to deployment, which included discovery, design, development, and testing. By the end of this phase of the project, the entire organization will have switched to the new software system with clean, reconciled financial and operational data.



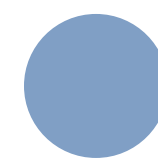
Step 1: Assess End-user Proficiency

End user training began in the development stage and will continue into deployment until all employees who will be using the new system have been sufficiently trained. Project team members will need to become subject matter experts within their respective areas and should also be given enough time to train others on the new system. For example, a controller will learn all about accounts payable and receivable, and general ledger. Then, the executive team needs to give the controller enough time to train the AP clerks, AR clerks, and so on. Before going live, all users must have enough time working with the system to comfortably do their jobs.



Step 2: "Go/No Go" Decision

As the project approaches the scheduled go-live date, the project team and implementation team should work through a final evaluation of system and user readiness by simulating running the business. Based on that testing, the teams should have a discussion to decide whether or not they are ready.



Step 3: Load Static and Dynamic Data

With the teams concluding that they should go live, the data load process starts. During the testing phase, the processes to load should have been tested. Now that it is time to go live the teams extract data from the legacy systems like vendors, customers and inventory item master records as described previously.

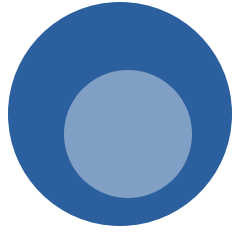


Once that data is loaded and validated, the team will extract the dynamic data and import it into the new system. That includes open invoices from accounts payable, accounts receivable, inventory balances, open sales orders, open purchase orders, open projects, etc.

The general ledger data is an area customers often inquire about. Some modern systems like Acumatica have the ability to load net change activity into the GL for any number of periods a company selects.

Many clients ask about bringing in transactional detail. The problem with bringing detail is that the transactions need to be replicated in the new system. So imagine loading each accounts payable invoice for say 2 years and then issuing a check to pay the invoices. Then you'd have to do something similar on the accounts receivable side, followed by reconciling the bank statement. Then you'd have to reconcile the activity in the new system compared do the legacy system, so you can see this would be a huge undertaking.

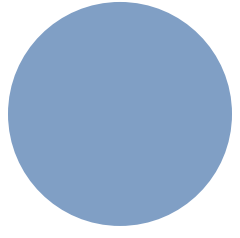
Instead we recommend that you bring in the net activity into the GL and keep your old system available for analysis when required. Depending on your system you could also export the transactional data to a database for subsequent querying.



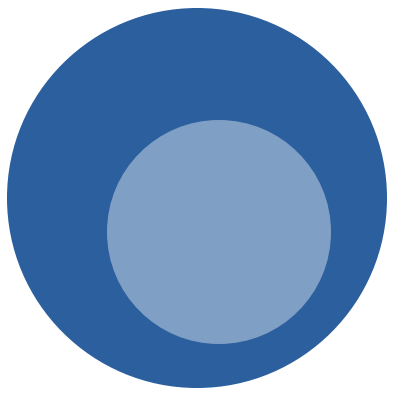
Step 4: Validate & Balance Against Legacy System

For each area financial and operational area the teams must validate the data that was loaded into the new system against the data in the legacy system. This is a critical step to ensure that the systems match and that you have informational integrity as users start using the new system. Accurate underlying data will help users and decision makers build confidence in the new system.

Step 5: Start Using It



Employees start doing their jobs in the new system and stop using the legacy system. Companies used to run parallel, which meant that during a painful transition period, people would do their job twice – once in each system, and then reconcile the differences. The problem is that it more than doubles the workload. At PC Bennett we recommend that instead of doing that, companies make a complete switch to the new business management system. Think of jumping out of an airplane with a parachute. This is why it is important for users to have had enough practice and training so that they are comfortable doing their jobs.



Phase 6 | Ongoing Support

Goal: To optimize system use & transition the project team

After the new system is live, there will continue to be a need for ongoing optimization of system use and productivity. There may be features in the software that the implementation and project teams decided to defer until after the system was live. Users may also find slightly different ways to utilize the system that help them be more productive.

Post Go-live

It's like using Excel. When you first start using the program you use basic functions as you build a foundation, then over time you learn more capabilities that address more complex situations.

Convert the Project Team

This optimization process begins as a joint effort between the implementation team and the project team, but shifts as the project team takes on more of the lead role. Eventually they will convert to more of an IT steering committee that looks at new system requests from users and compares them with the IT resources available.

Responsibilities for the steering committee also include ensuring that new users are properly trained in the use of the system. Studies have shown that a departing user transfers roughly 10% of their knowledge to the new user. And if that role has several changeovers in a period of a few years, the way the system is used may diverge from how it was designed to be used. This can lead to a high level of frustration with the system, a loss of productivity as users develop unnecessary workarounds and degradation in the quality of information available in the software.

"I have been called into over a dozen companies where the user frustration level reaches the CEO who then calls the software company to complain about the system. In all cases I found that a lack of proper user training and procedures leads to a lack of system understanding by users who then create workarounds to get their jobs done. One company had over 200 Access databases on the network to supplement the system – and there were roughly 50 users of the ERP system."

In the more severe cases, we formed a new project team and essentially re-implemented the system. A proper IT steering committee can help avoid those aggravations and expenses.

Identify Problem Areas & Develop Solutions

If there are some longer term structural changes such as acquisitions, or adding new business units there may be a need for rolling out new features in the system that were not necessary before.

So even though thorough planning and testing was conducted along the way, generally, there will always be a need for proper care and feeding of the system as the business progresses. This should be managed by the steering committee and members of the implementation team should be engaged where appropriate.