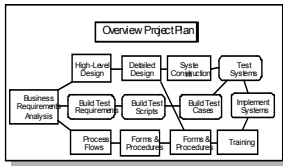


# A Systems Implementation Project Planning Guide

---

*Solutions & Project  
Management Services  
for  
Systems & Operations Projects*

---



# Systems Implementation Projects Are Often Complex

**Software Packages** Many systems implementation projects today involve the integration of purchased, enterprise level software packages. Unless there is a unique competitive advantage, it is simply not cost effective to develop your own corporate systems when similar systems are in use by most enterprises. Two examples of systems that are common across all organizations are human resources systems (e.g., payroll, benefits, etc.) and accounting systems (e.g., GL, accounts payable, accounts receivable, etc.). There are also examples of common systems that are shared within industry groups. For example: in banking (Demand Deposit Accounting, Savings, Consumer Credit, Small Business Lending, Collections, etc.), brokerage services (Equities trading, Fixed Income trading, Crossing, Portfolio Accounting, etc.), and insurance (Claims review and processing, Underwriting, Document Storage and retrieval, etc.). To meet these common needs vendors have developed an ever expanding list of commercially available software packages. In some cases the choice of vendors and software packages is huge, and getting bigger all the time. However, when an organization finally finds and purchases a packaged software product that meets its business needs, its work has really just begun.

**Beyond Shrink Wrap** After typically screening a number of vendors, evaluating responses to RFPs (Requests For Proposals), and then carefully selecting a vendor and a software product, the work really begins. These software packages are not as simple as installing “software in a box” (a.k.a. Shrink Wrap software) purchased from the local computer store. How do you get from your purchase of a software package to having the system installed, integrated and working effectively within your organization. These projects are complex because they involve so many players – user departments, systems development, audit, and computer operations groups to name a few. They also involve many different activities that must be closely coordinated in order to get the software installed and functioning successfully.

One solution is to use the vendor or an experienced consultant to help with the implementation and integration project. This can often be very cost effective, but you should still know what needs to be covered on the project plan so that you and your organization can be sure all bases are covered and you can be an effective and involved partner in the project. Also, even vendors and experienced consultants can overlook some aspect of the project that might be unique to your organization.

One of the factors that make corporate software so complex is the need for customizing and for interfacing with other systems. It is always best to use the software without modification, and most enterprise software packages have various parameters that can be adjusted to customize, at least to some extent, the application for your particular organization. If there are some features that the product does not have that your organization simply must have, then you have to accept that the project will be significantly more complicated. Another area of customization that usually must be done is to interface the new system with other systems within the organization. Two examples of systems that often need “to” or “from” interfaces include General Ledger and Human Resources systems.

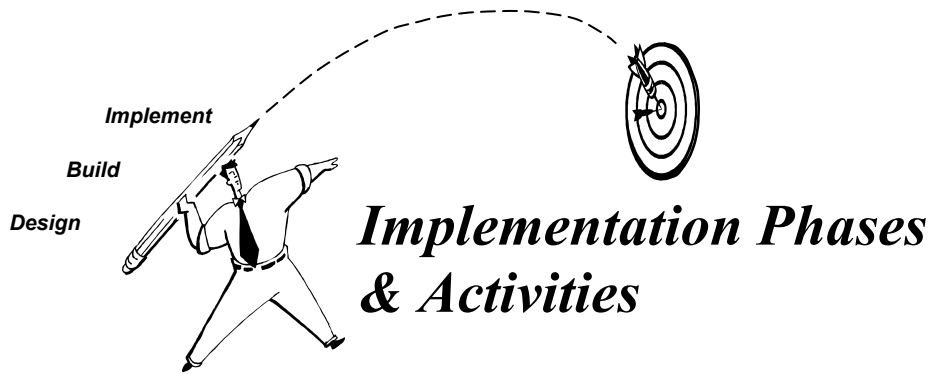
Most enterprise software applications are purchased with the objective that today’s work will become more automated, and that both work quality and work quantity can increase with the use of the new system. This means that in addition to installing, customizing, and testing new software, it is necessary to redesign core processes and procedures. This also means that workers will need to be trained in the use of the system and the new ways of doing business with that system.

Considering all of the possible activities that may need to be done when installing and integrating a new enterprise software product, how do you make sure that you have haven’t overlooked any key activities? Wouldn’t it be nice to have a checklist? That’s why we developed the Systems Implementation Activities Planning Guide, a checklist of activities that should be considered when implementing a new system.



**Activities Checklist** To help with your planning process we developed the Systems Implementation Activities Matrix. This tool is an organized checklist of activities you should consider when embarking on a systems implementation and integration project. It is designed to give your organization a jump start in the implementation planning process. The matrix should be used as a checklist that prompts you to consider activities that might be applicable to your organization and your unique project. If a particular area of the matrix is not applicable, you can feel free to skip it. But for all those that are, you should define the underlying tasks required to accomplish the described work.

The matrix is organized around two dimensions – project implementation phases down the side, and implementation activity areas across the top. Before explaining how best to use the matrix, let’s explain each of the two dimensions.



**I**mplementation Phases – All projects go through a life cycle beginning with defining how the new software package will be used in your organization (requirements) through the end point of the project – a successful and effective implementation. Our activities matrix has been organized around six generic implementation life cycle phases.

1. *Business Requirement and Proposed Solution* – this is the phase where your business requirements are finalized, the software package is learned, and a solution using the package is defined to meet the business requirements.
2. *High Level Design (Functional Specifications)* – the planned solution is further clarified by functionally specifying how the system will operate.
3. *Detailed Design (Design Specifications)* – in this phase detailed design specifications are developed (e.g., table values are defined, specifications as to exactly how reports will look and work are developed, etc.).
4. *System Configuration, Customization and Development* – the system is “programmed” by setting up its parameters and tables with the values defined in the phases above. Interfaces, data conversion and customized programming are also done in this phase. Quality assurance (systems and user testing) is done here.
5. *System Implementation* – in this phase the system is implemented and operations are converted to the new system.
6. *System Support and Maintenance* – this is the post implementation phase where the system is turned over to the normal support and maintenance process.

Most organizations use a standard development life cycle that they use when building or customizing systems. This life cycle breakdown is not meant to replace your own organization’s system. It is generic and can usually be easily mapped to the internal system used at your company.

**I**mplementation Activities Areas – These are the possible areas of project activity you may need to plan for in your project. These areas are categories of activities that are found on most systems implementation projects such as: testing, training and communications, and interfacing to and from other systems, etc. Not all of these categories may be applicable to your implementation project (e.g., maybe no interfaces with other systems are required), but it is important to consider each activity area carefully before deciding that it is not applicable.



**Using the Checklist** To use the Activities Matrix start with the first lifecycle phase, “Business Requirements and Proposed Solution”. Look at each of the possible activities in this phase. Carefully consider if each is applicable to your particular project. Check all those that are. Continue this process for each of the other lifecycle phases. Now decide who should coordinate the detailed planning for each of the implementation activity areas where you have checked activities. For example, who should coordinate planning for testing. Have that person hold a meeting with all of the parties who should be involved in that activity. Starting with the activities checked off on the Activities Matrix, begin the detailed task planning process for each of those activities. The first step is to develop a task list outlining the tasks that will need to be done to complete each activity. Next you will need to estimate how long each of the tasks will take, and decide what other tasks (predecessor tasks) will need to be done before your task can start. At this point a project planning tool like Microsoft Project is useful for organizing and linking your tasks into a coherent project plan.

Half of the battle in formulating realistic project plans for a systems implementation is identifying all of the tasks that will need to be done. The objective of the Activities Matrix is to help you include all of the key activities required for your successful systems implementation project. But don’t just rely on the Activities Matrix alone, talk with people in your organization who have planned and managed successful projects. Ask these people and others to review your plan for completeness. While it is not possible to think of every single task on a major project in advance, it is possible to plan for most of them. The better the planning, the more accurate your estimates will be for the budget and duration of the project. Also, the more likely the project will achieve the on time, on target, and on budget project success that you and your organization are seeking. Good luck to you and your project team!



**Cliff Consulting, Inc.**

**Your Partners for Project Success**

**OVERVIEW OF SOFTWARE PACKAGE IMPLEMENTATION TASK PLANNING ACTIVITIES**

| Implementation Phase |   | Implementation Activity Areas   |  |   |  |  |
|----------------------|---|---|--|---|--|--|
|                      |   | A. User Requirements  | B. Systems Environment   | C. Training & Communications  | D. Security and BPR  | E. Interfaces  |
| 1.                   | <i>Business Requirements, RFP and Solution Selection</i>        | Document high level business requirements. Include data needs, business functions & logic to be supported, as well as flexibilities needed to accommodate both needs now and in the future. Develop Request for Proposal (RFP) based on requirements.         | Determine systems standards required for the target enterprise. Include network, OS, database, language, and platform constraints. Incorporate any systems requirements or constraints into the RFP if acquiring software.   | See demos of prospective systems. Review manuals of target systems. Interview and conduct site visits with users of target systems. Begin communications with all effected personnel about the new system, its benefits for them, and the time table.                       | Incorporate security requirements in the business requirements and RFP. Review proposed systems for corporate security standards that are required.  | Determine scope of interfacing required based on work flow issues and business requirements. Incorporate interface requirements in RFP.  |
| 2.                   | <i>High Level Design (Functional Specifications)</i>            | Learn the new system's features and capabilities. Develop an application solution document to map requirements to the system capabilities. Specify for each business requirement the way the system will be configured or customized to meet the requirement. | Develop systems environment and regions migration plan. Establish development, test, and other physical environments as required. Install vendor software, OS, network, and change control software as required. Design systems support solution for new software. | Develop training plan & attendees list. Both users on the project and technical staff attend appropriate training classes in the new system. Read systems manuals, "play" with demo systems to learn appropriate functionality. Define empl./cust. communications strategy. | Define high level security requirements at the user type level (for: data, screens, tables, menus, etc.) as required or allowed by the application and systems software. Establish security administration standards and procedures. Define planned BPR (Business Process Resumption). | Document existing interfaces and high level requirements (formats, record layouts, header and trailer records, etc.) for interfacing to required systems (e.g., payroll service). Define high level interface requirements for new system. |
| 3.                   | <i>Detailed Design (Design Specifications)</i>                  | Document detailed requirements specifications for screens, reports, table set-ups, calculations and algorithms, etc. Revise mapping document to reflect final design implementation and work flows.   | Determine server and client hardware requirements. Allocate equipment and space for new application support. Install standard software. Define network support requirements.   | Determine approach and training guide format to be used for user training. Publish employee communications per plan.  | Define security profiles for users and/or user groups. Specify security tables. Develop security audit and control procedures. Complete detailed BPR process design.   | Develop detailed interface data maps with data recodes and conversions as required. Develop interface job requirements including media, timing, protocols, etc.  |
| 4.                   | <i>System Configuration &amp; Customization and Development</i> | If design changes are required during configuration, make sure that requirements documentation is updated.  | Establish development and test environments and define migration controls. Develop systems operations documentation. Develop and test batch processing jobs for application and interfaces. Establish & test production environment.                               | Develop system user's guide. Develop user training materials. Develop terminal operator quick reference materials. Publish appropriate employee communications per plan.  | Load system security tables, set security parameters, and establish system security. Test all security features and functions. Document and test BPR procedures.   | Code, unit test, and systems test interface programs.  |
| 5.                   | <i>System Implementation</i>                                    |   | Go live with new system and systems administration and maintenance procedures. Back-up, archive and decommission old systems. Monitor system performance and tune as needed.   | Implement employee or managers help line if required per communications plan.   | Begin security administration under new system. Implement BPR.   | Implement new interface programs for all "to" and "from" systems.  |
| 6.                   | <i>System Support and Maintenance</i>                           | Keep requirements documentation up to date when specifying changes to be made to the system.  | Update system "run" documentation as required.   | Publish post implementation communications per communications plan.   | Periodically review audit reports and security system. Keep-up with security administration, delete access when no longer needed. Periodically test BPR as required by corporate standards.  | Make sure changes to interfacing systems are communicated so that interface programs can be modified and tested when necessary.  |

**OVERVIEW OF SOFTWARE PACKAGE IMPLEMENTATION TASK PLANNING ACTIVITIES**

| Implementation Phase |   | Implementation Activity Areas   |   |   |   |  |
|----------------------|---|---|---|---|---|--|
|                      |   | F. Conversion   | G. System Selection, Design & Customization   | H. Work flows & Forms   | I. Tables & Parameters  | J. Data  |
| <b>1.</b>            | <i>Business Requirements, RFP and Solution Selection</i>        | Include any unique data conversion requirements in both the business requirements and the RFP.  | Select candidate systems. Do preliminary screening (RFI) and select finalists for receipt of RFP. Evaluate responses and select system.   | Document existing work flows and procedures. Identify systems support improvements needed. Incorporate improvements in RFP.   | Document as a part of the business requirements any unique tables required to support the business. Include these special table and/or parameter requirements in the RFP. Address any date sensitivity needs as table values are changed. | Document any unique or key data needs in the high level business requirements. Incorporate key data or data structure (data model) requirements in RFP.  |
| <b>2.</b>            | <i>High Level Design (Functional Specifications)</i>            | Develop high level conversion plan and strategy. Develop first draft of data conversion maps including rules for recodes and data conversions.  | Install system and conduct initial product acceptance test. Establish global parameters and tables. Configure prototype and/or initial test, and development/customization systems. | Design streamlined high level processes and work flows. Inventory existing forms, and recommend changes to existing forms sets to match system and reengineered work flow requirements.   | Review new system tables and determine those required and their use. Review existing codes and change as required to make compatible with new system. Establish table administration standards and procedures.                            | Inventory all current data elements. Document any additional data elements needed or current data that should be eliminated. Document data history requirements. Establish data administration standards and procedures for the project. |
| <b>3.</b>            | <i>Detailed Design (Design Specifications)</i>                  | Design and finalize conversion plan. Plan should include specifics of conversion phasing (e.g., which populations will be converted when), if any.  | Document detailed, technical "internals" design required to meet customization specifications.  | Define table values for each functional area. Develop detailed screen design specifications for each functional area. Design new or modified forms. Document new work flows and procedures. Define error correction. Develop terminal operator's guide. | Define values and keys for global tables and parameters, and function specific tables and parameters. Develop detailed specifications for all required table changes and define new tables if required.                                   | Map data elements from current system(s) to new system. Document recodes and conversions required, and edits & other business rules to be embedded at the data level. Document data dictionary.  |
| <b>4.</b>            | <i>System Configuration &amp; Customization and Development</i> | Code and test required conversion programs. Develop conversion verification and balancing procedures. Develop detailed conversion scripts for "conversion weekend" including checkpoints and fall back plans. | Code and unit test all system changes including: modifications, conversion programs, interfaces, database changes, screen modifications table adds and changes.                     | Update on-line help panels consistent with new procedures.  | Load global tables and parameters. Load function specific tables and parameters.  | Determine current data and history data clean-up requirements. Clean-up data as required before conversion. Make final data dictionary changes.  |
| <b>5.</b>            | <i>System Implementation</i>                                    | Convert and/or enter data. Balance and verify as required data. Certify system data ready for production.   |   | Once system is converted and certified, cut-over to new forms, work flows and procedures.   |   | Convert data into new systems database and verify data completeness and integrity.   |
| <b>6.</b>            | <i>System Support and Maintenance</i>                           | Archive data from converted system.   | Update application customization documentation as changes are made. Stay up with installation of vendor releases.   | Update work flows and procedures as changes are made.   | Begin table administration and maintenance in the production environment. Make sure all table changes continue to be documented.  | Implement data administration procedures in the production environment.  |

## OVERVIEW OF SOFTWARE PACKAGE IMPLEMENTATION TASK PLANNING ACTIVITIES

| Implementation Phase   | Implementation Activity Areas  |   |  |  |   |
|--|--|---|--|--|---|
|  | K. Screens   | L. Reporting  | M. Testing   | N. Project Management & Administration   | O. Vendor Related Activities  |
| 1. <i>Business Requirements, RFP and Solution Selection</i>        | Document any unique or key screens needed in the high level business requirements.   | Document business reporting requirements. Inventory existing reports. Identify any unique reporting requirements and document any special report writer needs -- include in RFP. Streamline existing report requirements.                     | Define a minimum set of software acceptance test functions which must be satisfactorily performed for product acceptance.  | Select vendor. Prepare software acquisition request/justification once vendor is selected. Prepare implementation plan and budget, get approval. Negotiate and finalize contract with vendor. Make subject to a product acceptance test. | Develop list of potential vendors. Screen vendors with RFI. Evaluate and select vendor through RFP, demos, site visits, and reference check outs. Negotiate contract terms and price with vendor. |
| 2. <i>High Level Design (Functional Specifications)</i>            | Review the new system's screens. Identify all changes and/or new screens that will be required to support the business requirements and new work flows. Establish screen administration and documentation standards. | Select report writer. Map system supplied reports to current reports, identify needs. Document new system reports to be used: "as is", to be modified, and to be developed. Establish report administration standards.                        | Conduct initial software acceptance test. Develop test strategy document. Based on business requirements and/or business processes and functions supported, begin documenting test cases which will be required for user acceptance testing. | Acquire system. Manage implementation project plan and budget. Conduct status and steering committee meetings.   | Contract signing. Vendor install support. Vendor training.  |
| 3. <i>Detailed Design (Design Specifications)</i>                  | Determine number of screen versions needed. Develop detailed screen design specifications for all global screens. Specify screen flows.  | Document detailed report specifications for all new and modified reports. Define reports production schedule requirements. Install and test ad hoc report writer. Define ad hoc report "templates" and selection/sort parameter entry panels. | For each test case, develop test scripts based on detailed requirements. Define test population approach (e.g., convert test data or enter it) and specify test data requirements.   | Manage and administer project. Conduct status and steering committee meetings.   |   |
| 4. <i>System Configuration &amp; Customization and Development</i> | Develop new and/or modified screens.   | Develop reports and test using report writer.   | Populate test data for systems tests and for user acceptance test (UAT). Conduct UAT, stress, interface, conversion, and parallel tests as required by testing plan.   | Manage and administer project. Conduct status and steering committee meetings.   | Vendor supported customization.   |
| 5. <i>System Implementation</i>                                    |  | Begin generating and using new reports in the new production environment.   | Conduct appropriate installation tests prior to production roll-outs.  | Manage the conversion process. Use escalation process, if needed. Complete project documentation.  | Monitor bug reports and make bug fixes in core product. Provide contracted services and service levels.   |
| 6. <i>System Support and Maintenance</i>                           | Make screen changes as needed in the production environment. Be sure to fully test changes before implementation. Update documentation for all changes.  | Use the reporting features of the new system or a report writer to create or make changes to standard reports and to generate ad hoc reports as needed. Document all new reports and changes to existing ones.                                | Package the test kit for regression testing of new releases of the software. Develop new tests and add to the test kit as changes are made to the system.  | Conduct post implementation project and system review.   | Provide vendor maintenance. Make product upgrades and version releases. Provide "help" line support as contracted.  |