A GUIDE FROM CLD PARTNERS

HOW TO AVOID THE **COMMON PITFALLS** OF IMPLEMENTING CERTINIA

certinia

CLD Partners

Introduction

If you're considering implementing the Certinia (formerly FinancialForce) or if you have already*, we believe this document will be useful to you regardless at what stage you are in the journey.

Over the past 13 years, we've helped many enterprise clients across multiple industries implement Certinia. Because of this, we've seen what makes an implementation successful and what pitfalls create headaches down the road. We've assembled a list of **27 Pitfalls** to avoid and grouped them into 5 categories:

1. Project Execution
2. End to End Solution & Integration
3. Training, Change Management & Adoption Testing
4. Data Migration

This document is organized so you can get a quick glance at the implementation pitfalls under each category listed above. We described challenges we've seen with real examples. And then more importantly, we've made it easy to view the recommendations on how to avoid these pitfalls (or recover from them if you are in the middle of implementation now). We list helpful questions and considerations for your team so you can avoid the pitfalls.

Don't have time to read the whole doc right now? Skip to the end for two printable, quick reference lists you can distribute to your team.

Let's Dig In.

*** If you have been using Certinia for some time, remember your Certinia solution should evolve as your business changes and grows. If you haven't reviewed your solution recently, consider doing so with a CLD health check.**

Questions about your implementation? Contact us:
info@cldpartners.com

or

Talk with a expert.

#1 Project Execution Pitfalls

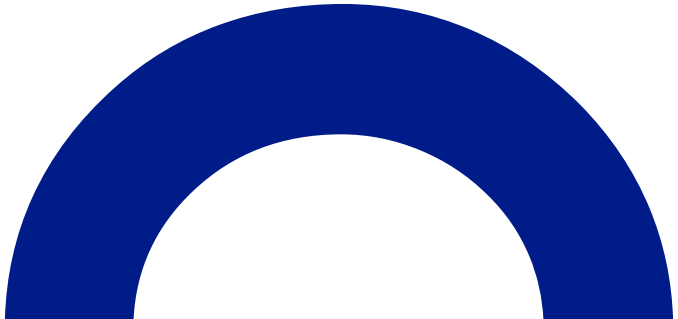
There are some common themes that apply to virtually all technology implementations. However these are the pitfalls we see most often with respect to project execution.

1. Not engaging all the stakeholders at the beginning of the project
2. Underestimating the time investment needed from key stakeholders & resources
3. Not clearly understanding decision making structure (boards) or processes
4. Underestimating the time needed for decision making
5. Overlooking downstream or upstream system impacts
6. Incomplete requirements before design
7. No representation from other geographies or business units during design
8. Low engagement from those participating in design
9. Not having a dedicated environment for the Certinia implementation project

When one or a combination of these pitfalls occur, it leads to these issues:

- Rework later in testing phase because new vital requirements are shared by the users or variations across regions are discovered,
- Extends the overall implementation
- Likely budget increases needed to complete the project

We respect your time and want to be good stewards of your budget as much as you do, so that's why we recommend the items below to mitigate and avoid the issues.



Here's how to avoid the Project Execution Pitfalls

When possible, identify one leader / decision maker

Although this is not always possible in large organizations with multiple stakeholder groups, we've seen clients be very successful when there is one person to make a final decision with respect to the project. This way if there is disagreement, it can be taken to one person for resolution.

If more than one lead is required for the implementation, the next best thing is to know and define the stakeholders and decision making processes.

Identify the stakeholders that will be affected by the change

When Certinia projects start, typically one group is leading the charge - sometimes this is the professional services group, the IT group, the finance group, the CIO, or the CFO.

We've also seen the human resources or resource management groups lead the effort because they want to utilize the rich features for skills management, resource allocation, utilization, etc.

Be sure to identify the stakeholders at the start of the project and keep in mind, *this likely includes secondary groups that are upstream or downstream from the professional services processes.*

Inform all the people/groups at the start of the project and provide updates throughout. It's even better if you can get one representative from each of the groups to be the spokesperson for his/her respective group.

Set expectations for participation and confirm project schedule is realistic

When CLD kicks off a project, the first thing we do is discuss the project schedule with our clients - this is the best time to determine if the timeline is realistic and to learn about other milestones. For example, maybe a new version of Salesforce is being pushed out org wide around go-live and there is a 2 week blackout period when changes can't be made to the baseline. Or perhaps quarter-end will be happening right when go-live is projected. We help our clients identify these potential risks to the implementation schedule.

This is easier when there is one leader who can direct how resources spend their time. It's best to remember:

> *Everyone has a day job but it will be important to block time off in schedules to support the design sessions, design reviews, user acceptance testing and other key stages in the project.*

Engage all your stakeholders up front

There are two easy ways to do this: (1) Inform your team about Certinia capabilities before the project starts, and (2) Determine if and when they should participate in the Design process.

We believe it's important to train your team and stakeholders on Certinia capabilities at the project start - *BEFORE* design starts. (You'll read more about why this is so valuable later in the training section).

> TIP: Create an agenda with design topics and sub topics to review and determine which stakeholders should participate in your design process before it begins.

If you're not sure where to start, here's how we do it:

1. We use a standard workshop agenda template and then we tailor it based on the project scope and we send it to our client for review prior to the start of the design sessions.
2. On the workshop agenda, we list stakeholders that you might need to identify and invite to the design sessions.
3. In a typical implementation, we have anywhere from 6-10 workshops to ensure we cover all aspects of your project (which includes workshops for Integrations & Data Migration).
4. Here's what a Workshop Agenda looks like (this is day one showing two workshops).

Time	Workshop Name	Topics	Subtopics	Estimated Duration	Suggested Attendees	Suggested Prep (Client)	Suggested Prep (CLD)
8:30 AM - 12:30 PM	**Workshop 1: Organization Configuration**			**4 Hours**			
8:30 AM		Time Periods	Time Period Types Time Period Naming	30 min	Business Finance Project Management		Sample time period listing
9:00 AM		MultiCurrency	Project currency Resource Currency Conversions for Billing, Costs, and Expenses	30 min	Core Team Finance	List of currencies that need to be supported (if not already defined in SFDC advanced multi-currency)	
9:30 AM		Regions, Practices, Groups	Project Region Assignment Resource Region Assignment Reporting Access Control Transaction "Follow Rules"	1 hour	Core Team	Region List or Chart - Geographic List of Lines of Business, BUs Org / Department List / Chart	Sample R/P/G listings
10:30 AM - 10:45 AM	Break			15 min			
10:45 AM		Work Calendars & Holidays	Resource Scheduling Effect on Utilization Calculations	1 hour	Core Team	Detail on work calendars / holidays to be supported for client, by region	Sample Work Calendar / Holiday List
11:45 AM		Global Projects	Internal / Global Project Listing Project Access by Region, Practice, Group	45 min	Core Team Human Resources	List of client internal projects for time entry, utilization, expenses, etc.	Sample Global Project listing with flags for self-staffing, utilization,
12:30 PM - 1:00 PM	Lunch			30 Minutes			
1:00 PM - 5:00 PM	**Workshop 2: Project Setup**			**4 Hours**			
1:00 PM		Project Creation and Setup Processes	Opportunity / Project Relationship Sales Order / Contract / Project Relationship Process Participants (Sales, Services, Ops) Process Flow(s) Project Updates and Synchronization	1 hour 30 mins	Core Team CRM Group Salesforce Admin Professional Services Group	Diagram or description of desired project setup process List of contract- or order-related data elements that will drive project setup and downstream operations (forecasting, billing, rev rec, etc.)	Preliminary process flow diagrams
2:30 PM - 2:45 PM	Break			15 min			
2:45 PM		Project Structure and Data Elements	Project record structure (budgets, milestones,...) Project-related data elements Project Type, Billing Type, Service Type, Dates, etc. Vendor Budgets (for subcontractors)	1 hour 15 minutes	Finance	List of service types (Implementation, add-ons, etc.) that need to be tracked/reported on.	Preliminary project structure diagram / description
4:00 PM		Change Orders, Project Updates, Shutdown	Change orders Updates Project Closure: Process, Validations, Notifications	1 hour		List / Description of key events in project lifecycle	Preliminary process flow diagrams
		Recap / Action Items Review		15 min			

Invite stakeholders to design sessions & design review

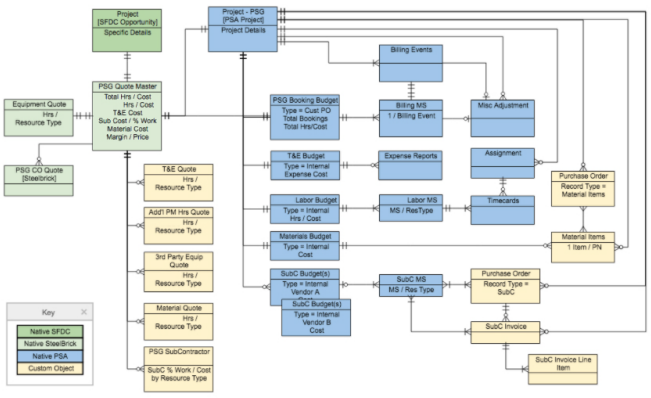
It's vital to have all the stakeholders in the room when you talk about your end to end process and ideal solution at points throughout these sessions. They may not need to be there the whole time, but they DO need to be informed from the beginning to ensure upstream and downstream processes/integrations run smoothly. For example, it's possible the Professional Services group may not know all the requirements the finance/accounting and/or workforce management groups have.

Historically we come onsite and talk through your end to end processes using a whiteboard – these days we can do this remotely. During these sessions you can expect:

1. **A complete walkthrough of the system functionality and features:** Leveraging what you learned in the Fundamentals course, we ask you questions about your processes (EASY!) so that we can identify the features and configurations you need.
2. **A Complete Design Specification Document:** We'll take lots and lots of notes that we use to create a complete design document that captures all the epics and stories required for your implementation. (Not stories you read, but stories that describe the discrete items we'll build after the design is approved.)

CLDPartners
CLOUD ERP. REALIZED.

PSG Quote / Project ERD



Functional Design and Process Flows

This section covers the business and system processes associated with creating a PSG quote with supporting estimate components and the associated PSG Project in PSA. Diagrams where applicable are used in conjunction with supporting text to describe the process steps and relationships between elements of the system design.

PSG Quote & PSA Project Shell Creation and Update Process Overview

PSG Quotes are created using a SB quote on the same SFDC Opportunity as the Equipment Sales Team uses to produce an Equipment Quote. The process for creating the PSG quote and related PSA Project shell is depicted in the diagram below and described in the sections that follow.

CLDPartners
CLOUD ERP. REALIZED.

Project Awarded	This stage indicates that the PSG Project is ready to be staffed. Resources are assigned. Code will change the PSG Project to this stage upon the PSG Quote moving to the "Project Awarded" state[35]. PM can submit POs to vendors.
In Progress	PSG Project activities underway; the Project Manager will move the project to this stage after completing the resource assignments.
On Hold	PSG Project activity on hold - will restart at later date. The Project Manager will manually move the project to this stage if needed during the project lifecycle.
Work Completed	All work is done; Time entry is closed.
Project Completed	All PSG Project activity Completed; Billing should be completed prior to moving the project to "Completed". Vendor invoices and Subcontractor invoices may come in during the Completed stage. The Project Manager will manually move the project to this stage.
Cancelled	PSG Project Cancelled; Time entry and billing will no longer be permitted once the project is in this stage. Project Manager will change the stage of the PSG Project to Cancelled if the PSG Quote stage moves to "Cancelled".
Warranty	PSG Project reopened, or additional work assigned after Project Completed date. Time entry and billing permitted.

Custom buttons on Projects

The following buttons will be added to the Project Page Layout.

Button Name	Description
Create Change Order[36]	Triggers code to create shell Change Order Quote and takes user to new quote
Manage Materials[37]	Launches Material Mgmt UI
Close-out Subcontractor Work	(Future Enhancement) Discussed having feature to close budgets and milestones for specific sub.

PSG Project Custom Fields

The following table summarizes the custom fields that will be added to the PSG Project.[38]

Table: Custom PSG Project Fields

Field Name	Type	Description
QB ID	Text(5) (External ID)	Used for projects migrated from Quickbase to capture the Quickbase project ID
PSG Master Quote		

[35] Add validation to prevent Stage = Project Awarded if PSG Quote is not Approved
[36] Add button for Create Change Order on Project Page Layout
[37] Add button for Manage Material on Project Page Layout
[38] Add Project Custom Fields noted in the table in Section 4

These are pages from a design document. We use footnotes to clearly denote configurations and they are hyperlinked to stories for easy tracking.

Have a dedicated development environment for the Certinia implementation

We recommend clients set up an environment that has a recent copy of their production salesforce org. This environment should be dedicated to the Certinia implementation.

It's not always possible in larger organizations, so if you cannot have a dedicated org, be aware that if development is occurring by other teams, it could affect the Certinia project. We've seen development get held up while code conflicts are resolved with another project team.

#2 End to End Solution & Integration Pitfalls

It's important to consider the entire process up front and when you do it usually involves considering integration / interaction with upstream and downstream systems. It's important to consider the whole solution even if you are not tackling the whole solution in the current scope. Here are the pitfalls we've seen most when it comes to looking at the end to end solution.

1. Only focusing on part of the solution
2. Trying to over-automate or over-complicate the solution
3. Making too many changes at one time
4. Misalignment of Core Enterprise Processes that interact with Certinia:
    a. ERP/Order Management
    b. CRM/CPQ
    c. Billing and Invoicing
    d. Revenue Recognition
    e. Expense Management

How to avoid the end to end solution & integration pitfalls

We recommend you consider the end to end solution to avoid designing in a bubble even if you plan a phased roll out and/or manual processes (e.g. processes triggered by a person rather than automation).

Some clients simply use PSA and create Projects From Closed-Won Salesforce Opportunities and Invoice directly from Certinia. Others might integrate PSA and other Certinia apps with corporate systems like SAP or Oracle, Workday and Concur for CRM, Order Management, Billing, Expense Management, and Resource Management.

Understand how Certinia will integrate with your CRM and/or CPQ solution

Certinia is great because it uses the Salesforce platform, so naturally it's important to understand how the CRM processes will fit together with service delivery processes.

Leveraging a services quoting solution on Salesforce like **PSQuote** greatly enhances the quality of data that can be integrated into Certinia. It brings the services estimation data on platform and can be connected seamlessly with PSA closing the gap between CRM and PSA.

Whether you use PSQuote or still quote in spreadsheets, here's a short list of questions to consider as you get started:

- What information from Opportunities will be used to create a project
- What information from Quotes will be used to create a project (if any)
- At what stage of an Opportunity do you want to create a project
- Is there an order management system that needs to be considered when implementing the Certinia

This is a complex part of most Certinia Implementations and can't be summed up in only 4 questions so take time to work through the hand-offs between the two systems.

Our clients are streamlining their sales to delivery handoff while improving their services forecasting and resource demand planning with PSQuote. Plus it simplifies the integration between CRM and PSA since its a readily available app on the Salesforce AppExchange.

Does this seem overwhelming, don't worry, we can help.

Visit psquote.com to take a product tour.

Think about how PSA and revenue recognition processes will be synchronized

Many organizations struggle with revenue recognition. If you plan to use the revenue management module, then you'll need to consider the items on which you want to capture revenue transactions.

Revenue recognition is fundamental to assessing a company's performance and prospects. When it is not properly adhered to or managed, it will affect the balance sheet. Certinia enables businesses to gather information from revenue generation activities, and create "Rev Rec Workflows" based on Salesforce platform trigger events. Certinia can utilize revenue information from projects, opportunities, contracts, and other objects to keep your PSA and revenue recognition processes in sync for accurate reporting and financial compliance.

Clients typically handle revenue recognition in one of these ways:

- Project Percent Complete
  - Typically used for fixed price projects where a percent complete is calculated from underlying PSA timecard and assignment / schedule data.
- Timecard and Expense "Deliverables"
  - Revenue is recognized as Time and Expense is approved under this method
- Equal Split over a period of time:
  - Typically used for subscription or license type revenue where recognition needs to be split evenly between a start and end date

With the above methods in mind, here are some important considerations for PSA setup:

- How is PSA information rolled up with a "bottom's up" approach to calculate project percent complete?
- Do any projects include multi element or multiple performance obligations where revenue to recognize must be allocated according to ASC 606 guidelines?
- How do change orders or adjustments to timelines contribute to additional revenue or revenue claw backs?

There are many variations to the above examples of how PSA data can be fed directly into the Certinia Revenue Management tool. We help our clients design a solution that provides complete control and visibility of their revenues.

Determine how you will bill & invoice clients

This is usually something known prior to starting a Certinia project, however it's important to decide whether you plan to bill out of Certinia or determine if you need to Integrate with a corporate ERP system for Billing & Invoicing.

If you plan to bill out of Certinia, then consider the following:

1. How many invoice templates need to be set up?
2. Will you use Conga or another tool for the templates?
3. What level of line item detail needs to be on an invoice or do you need flexibility depending on the contract?
4. Who will be performing the billing event generation?

If you plan to integrate billing information to another system, then consider the following:

1. Will you integrate the billing event details directly?
2. What level of detail is needed in the other system?
3. Does any of the data need combined/summarized based on type (Time, Expenses, Milestones) in PSA or will this be done in the other system?
4. What invoice information be sent back to PSA for the Project Manager to see

If you're not sure how it should work, consider postponing automation

Sometimes it's better to get your solution live and then add in the *right* automation after you and your team fully understand how to best leverage the system in the 'real world'.

Be wary of over customizing your solution

Similar to the point above, it's sometimes best to use your solution and get some run time in the system. This will help you verify the customization you want is the customization you need. Just because we can build it, doesn't always mean we should… and don't worry, we'll tell you this when you ask!

In some cases, it's better to delay adding customizations until a later phase to implement after there has been some run time in the system. Even when there is clarity on how the system works, we sometimes find clients choose to delay some customizations when there are many new process changes. This eases the change management burden.

#3 Training & Change Management Pitfalls

Training and change management play an important role in the successful launch and adoption of a new system. Changing business processes need to be clearly communicated before roll out, users need to be trained on the key system elements related to their job functions, and a champion network needs to be established to help promote the new system and provide localized, front-line support after launch.

Here are the most common implementation pitfalls related to Training & Change Management:

1. Not providing up front training to stakeholders on what FinancialForce can do, which can lead to too much automation, over customization or trying to implement processes not conducive to FinancialForce
2. Not having a clear mandate that users must move to the new system
3. Not preparing a training environment or plan for end user training
4. Not communicating to users about the new implementation
5. Assuming that the internal training team will:
    a. Be available
    b. Have enough knowledge of PSA to build content
    c. Accurately estimate the amount of time / effort to build content, prepare trainers, and organize training events

Training

In the context of this document, Training refers to teaching users about the technology and processes in the solution. Training can be provided to any kind of user that will interact with the new system. It's up to you who, how and when you'll train your resources. This can range from end users who enter time and expenses only all the way to "Power Users" who administer and manage approvals and core processes in Certinia.

Change Management

Change Management can mean different things to different groups of people. When we talk about change management in this document, we're talking about *organizational change management. This means we'll be focusing on communicating:

- How introducing different and new systems and processes may disrupt folks day to day work.
- How you might also be impacting a person's existing role (asking them to do something more than they do today or asking them to do something different) - For example: *maybe you'll ask Project Managers to approve Timesheets in PSA and they don't do that today."*

* "The clearest definition of this type of organizational change management (OCM) is provided by Sheila Cox [(a pioneer in the field of change management)] of Performance Horizons who states: "Organizational change management ensures that the new processes resulting from a project are actually adopted by the people who are affected."" from CIO.com

Here's how to avoid the Training and Change Management Pitfalls

Train the core project team up front on what Certinia can do

Sometimes folks don't fully understand all the capabilities of the Certinia and that's why we train up front. Here's how we do it:

- We provide a 2-day in person or 6-day remote Fundamentals training course to the participants you identify on the system capability that comes out of the box.
- We have written training materials that we provide to each student
- Each student will be set up with an account in a sandbox (you'll all work in the same sandbox just like your future system will be)
- The training is mostly hands-on. Our instructor will inform you of key concepts, processes and terms and then this will be followed by hands on exercises in the system

If you plan to do this on your own, then you might consider asking Certinia to provide several demonstrations on how the system works to your core team. Focus on the core processes you'll be tackling, and ask your implementation partner specific questions about system functionality.

Create a simple Change Management Plan

Create a change management plan and include these items below. When it's documented, it will be easy to execute.

1. **Identify the change manager** - this is the person who will communicate to the users that the new system and processes are coming.
2. **How you plan to communicate** - do you plan to use just email, have a conference call, hold a lunch & learn, post it in a zoom channel or a combination of methods - remember it doesn't have to be complicated as long as you use communication to users in a way that's typical for your company.
3. **A schedule of communications** - this can be as simple as having a few "lunch & learns", giving a quick update or a schedule of email communications that will be sent to the users at given points throughout the project - if you have a bigger implementation across regions/continents - then it might need to be more elaborate.
4. If applicable, **list out the possible functions/tasks that will be a change for users** - This does not have to be a full exhaustive list. These items will become clear after the design is finalized and before UAT starts. **Note:** Don't forget to include reporting - **if you are not migrating your data to the new system, then your users will need to understand how to access and view the data they need to do their job.**

Communicate regularly with end users at key milestones

Communication can be as simple as using email to tell users and stakeholders that the Project is starting and providing a tentative schedule.

Or, you can get more detailed and put this information in the change management plan. Consider communicating to your users/stakeholders at these key milestones:

- At the project start
- To ask for volunteer testers to participate in User Acceptance Testing
- Letting folks know that testers have be identified
- When the design has been finalized (and a heads up on likely process changes)
- When UAT is starting
- A week before the new system goes live and include who and how they can contact someone with questions
    - We've seen some clients be very successful when they have set up "office hours" to answer questions from users at given times in the first few weeks.
    - We've seen clients have dedicated email addresses or places to post questions for real-time help
- After the new system goes live reminding folks to login and how to get help

Try not to over complicate the end user experience

To do this, simply take time during the design and project phases to ask:

- ”Is this necessary?”
- “Why is this necessary?”

These two questions will go a long way to ensure that you’re keeping the user and the overall end to end solution in mind. And it's why these are common questions we will ask our clients throughout the solution design phase.

Here are real-life questions that we’ve asked our clients:

1. **Is task time entry really necessary? If so, why?** We’ve had clients answer “yes, because we have to invoice at the task level because the contract requires it” or “no however we want to have a better understanding of what folks are doing”. In the latter case, consider requiring daily or weekly notes instead.
2. **Do you really need to hide certain fields from a certain user group? If so, why?** There may be legitimate reasons (subcontractors will be working in our system and shouldn’t see bill rates or certain government clients might require protocols that require restricting content.) We tend to recommend clients should always start from a position of complete transparency – it's good business and it reduces implementation and testing time. So, someone should have to have a good reason to restrict why another employee at the same company who could bring new insight should not see/do something in the new system. There are special circumstances and by all means those should be accounted for and backed up with a solid answer to the “why” question.
3. **Is Private-sharing necessary?** If so, why shouldn’t all project managers see all the projects (even those that are not their own)?
4. **Do your billing/invoicing folks spend a lot of time correcting records (time, expense, invoices)?** If so, determine should they be doing this and dig into why the data is incorrect in the first place and fix it at the source. Also it's usually best when the person who entered incorrect data should always be the person who has to fix it. This usually transfers work from downstream to upstream, decreases the overall amount of work, and increases the quality of the data from the get-go.

Be willing to re-engineer a process

Sometimes folks are eager to implement but don't want to change how it's always been done. Similar to the point above, it's important to ask these types of questions when you want to implement the same process in the new system. Consider:

- Does the process work well now? Why or why not?
- Were there any prior system dependencies that drove a process to work the way it did?
- Will the process still work well in the new system?
- Is there any way to streamline the process?

This is also an area where your implementation partner should be able to advise you on options as well as how they've seen other clients implement similar processes. If you have *big* questions, consider **talking with our experts.**

Do you have a BIG question?

Contact us: info@cldpartners.com

or

Talk with an expert.

#4 Testing Pitfalls

Test preparation takes time, but is critical to the success of testing. A thorough test plan, a complete set of test cases, training, and a testing schedule that allows end users to validate the system are absolutely necessary. This section mostly relates to User Acceptance Testing (UAT) but some implementations require System Integration Testing (SIT) and Stress Tests (e.g. can the system handle the volume of records common to your day to day business).

Preparing for UAT will help you find issues before going live when changes are easier to make and retest. (Once your system is live, you can expect a longer amount of time to create, test and deploy a fix because you're dealing with a production system.)

Expectation management is also important - defining how feedback will be triaged and addressed before end users start their testing helps control the inevitable scope creep that occurs during UAT.

When UAT is run well, you'll build user support for the solution, and it will foster user adoption which is also very important. **(A great system is only great if it's used.)** Pro-tip: if you think you'll have a resistor in your ranks, maybe identify them as a tester so they can bash on the solution during UAT rather than at go-live.

Common Testing Pitfalls

1. Poor test preparation/management (Not Writing Test Cases)
2. Shortening Time for User Acceptance Testing if the schedule slips
3. Not Training Testers on the environment prior to testing
4. Not conducting System Integration Testing for Integrations

Here's how to avoid the Testing Pitfalls

Write Test Cases

This may seem like a no-brainer yet sometimes writing test cases falls secondary to the other work so it's important to plan for it and do it. The best time to write test cases is after the design is finalized but there's no reason you can't start to flesh out test cases based on your processes during design. The folks who will be overseeing day to day operations in the system are likely the best folks to write the test cases with help from special users (such as a resource manager, project manager, Certinia administrator, and finance manager).

The effort to write test cases will provide two additional benefits: (1) the process of writing the test cases in and of itself is a form of education/training to those who will be overseeing operations, and (2) it helps you confirm your processes. You may even find that the test cases will mirror the training guides you need to create for your users.

Review the test cases with your stakeholders to make sure all the important tasks and processes will be tested.

If you're not sure where to start, keep these things in mind:

- Make sure each test case is discrete and not too broad
- Each test case should have the following items:
  - A test name (e.g. Create a Contact Record for a new PSA resource)
  - Steps on how to complete the test such as
    i. Click on the Contact tab.
    ii. Click on New Button
    iii. Enter name Joe Finn
    iv. Enter…
    v. Click..
    vi. Verify…
  - A role that should perform the test (e.g. Resource Manager)
  - Expected result statement (success or failure as you may want to create negative test cases to confirm access and/or validations)
  - Who performed the test
  - Actual result of the test
  - Notes

Other items to keep in mind:

- Some clients choose to create their test cases based off of the user stories generated at the beginning of the project.
- Test cases may need to be grouped together to test a whole process
- A test case may need to be performed by different roles with different expected outcomes (e.g., a Project manager can create an assignment, but a finance manager is not permitted to and should not be able to save a new resource)
- Some clients track test case execution in a tool like Jira, Salesforce, and others use Excel. Be sure to track test case execution, verify the tests were completed and then address any issues detected during the test execution. Keep in mind some tests might fail due to how the user performed the test or due to a bug. It's good to catch both during testing.

Here's an example of what your test case template could look like:

ID	Functional Area	User Story Title	Test Script	Actor	Pre-conditions	Expected Result	Tester	Notes:	Pass/Fail
2	Contracts	Attach documents to on object			Object exists for attachment	Documents attached			
3	Account Mgmt	Create a client account for PSA projects				Client account complete			
4	Resource Mgmt	Create a new hire PSA resource	Navigate to the Contacts tab and click on the New button 1. Select Record Type = PSA Resource, click on the Next button 2. Complete all required fields: Account Name = Company Account First Name Last Name Email address **PSA Resource Information** Start Date = hire date Resource Role Salesforce User Work Calendar Region Practice Is Resource = True Is Resource Active = True External Resource = False Exclude from Missing Timecards = False Exclude from Resource Planner = False Exclude from Time Calculations = False Exclude from Time Variance Calculations = False 3. Click on the Save button		User license is created	User set up complete			
5	Resource Mgmt	Create a sub-contractor PSA resource				User set up complete			
6	Project Creation	Create a billable Project from an Opportunity			Opportunity exists and is Closed Won	New Project created			

Determine which environment to use for testing

Most often we've seen clients choose one of these two ways. There are pros and cons to both approaches.

1. **Test in the development environment** - the same environment used by the developers to build everything. This can often be easier and faster since there is no need to deploy changes to another org, and additionally you may have some data readily available for testing. However, if development is occurring in an agile fashion (sprints), and there are other development teams using this org, think about if or how it will affect testing in the same environment.
2. **Test in a separate staging environment** - This path certainly insulates your testers from any changes developers may be making. You'll have to ready the environment prior to the start of testing by deploying the solution, setting up users, staging data, and so on. Changes for retesting will have to be deployed to the new system rather than retesting in real-time.

Identify testers and ensure that all necessary roles and stakeholders are represented

If testers are not identified ahead of time, it can throw off the schedule to secure their time to test, get them setup in the system, etc.

Be sure to account for different roles performing tests/functions in the system. If the system administrator does all the testing - everything might work as expected because that person has some bypass permissions -- but if another role tries the same tasks, it could fail. For example if your PM tries to approve a timesheet and it fails - this is a good catch and means you might need to check the permission sets and/or controls are configured correctly!)

Plan for setting up your users ahead of time in the test environment - if its a PSA Project, this means you'll need to make sure that each tester has these items at the very least:

- A salesforce login
- The right permission set(s) applied granting access to objects and fields
- A permission control if they are going to be using staffing, time entry and other standard PSA features
- A contact record set up indicating they are a PSA resource
- The necessary region, practice and group designated for their testing

It's important that end users do the actual testing. They will likely uncover key issues because they may test in a way that the designer/builder never anticipated (which is part of the purpose of testing).

Plan for staging data to support testing

Plan for staging data to support testing -- depending on how the test cases are written, it might assume that the certain items are set up and ready for the team to test. For example, maybe your test environment doesn't have the full price book setup and entries need added prior to testing or possibly opportunities will need to be created based on real life opportunities for people to test project creation.

Set & manage user expectations (define how user feedback will be addressed)

It's common for users to request new features or changes when they start testing. We tend to find users will suddenly remember their pain points in the prior system and want to fix them once they are testing the new system out. We have also seen users desire that the new system work exactly like the old system and will make requests based on familiarity with current processes.

We recommend that you anticipate requests for a multitude of out of scope enhancements. The best thing you can do is define how UAT feedback will be triaged and addressed before end users start their testing. Otherwise you might find yourself overwhelmed with feedback coming from the users.

We recommend you have a way to capture feedback and a way to respond. At the start of UAT, tell your testers that the purpose of testing is to verify the solution meets the approved design. Let them know they can submit feature requests but not to expect them at go-live.

Ensure there is enough time set aside for testing

If you do this, then it will be less likely that you'll need to find places to cut the schedule (including testing time). Even better - build buffer time into your project schedule. Think about company holidays and common vacation times especially across regions.

Keep in mind that if you've built more functionality than you originally planned, you need to set aside more time for testing that functionality, not less.

#5 Data Migration Pitfalls

Poor planning for Data Migration can impact the solution, the project time and lead to panicked, last-minute work just before go-live, a time that's already stressful and overburdened.

Data migration is complex.

Common Testing Pitfalls

1. Underestimating time it takes or the transformation that will be needed to migrate the data
2. Thinking that it can be handled independently of the solution design
3. Waiting too long to start Data Migration Planning and Tasks
4. Assuming that the source data won't need to be cleaned manually
5. Assuming that the source data contains all the necessary information for PSA – it usually doesn't and this means that you will have to make decisions about how to supply that "missing" data.

Here's how to avoid the data migration pitfalls

Develop a data migration strategy at the beginning of the project

It needs to start with the development of a migration strategy that defines what is being migrated, what isn't, and why. This helps avoid surprises down the road. For example, some folks choose not to migrate completed projects which means this data will not be available for reporting. It's always better to discuss this ahead of time rather than right before go-live or run into questions after go-live when folks can't get the data they are looking for.
Also we find clients are most successful when they create the data migration strategy when design starts. We always include it as a topic in our workshop agenda.

It's also best when you can actually look at the legacy system and talk through the data that you want to migrate over, because this way you'll ensure that the design can support any legacy data migrated into the new solution.

Don’t wait to look at what data you want to migrate over

Following the creation of a plan, there needs to be a thorough analysis of source data, development of transformation logic and (often) tools to automate the transformation, and finally lots of trial runs into a sandbox with business user validation of the migrated data. All of this takes a lot of time.

If you wait to look at the data, chances are you will find a legacy data field that requires modifying the design, possibly impacting development and/or requiring retesting. *The data is never as clean as you think it is.*

Create a field level migration plan

It takes effort but it’s well worth it. A field level migration plan is what it sounds like: It's a place where you write down every field in the source system and identify where it will go in FinancialForce. This is sometimes easier to do in reverse - start with FinancialForce. First review the core FinancialForce objects and fields on those objects. For each object, determine where the data will come from or if you won’t be migrating any data into that object. Here’s an example of one such plan.

Source Record.Field	Salesforce Value: pse__Assignment__c	Salesforce Data Type	Picklist Values	Default Value	Transformation Rule / Fixed Value
	QB Key	Text			eg: Resource Email + '|' + Project External Id
Resources.SFDC Id	Resource	Lookup(Contact)			vlookup
Resource Email + '|' + Project Extern	Schedule	Lookup(Schedule)			vlookup
Task__Customer_Project_ID	Project	Lookup(Project)			vlookup
Resource.Title	Role	Picklist	Control Engineer Mechanical Engineer Project Manager Sales Engineer Solutions Engineer Systems Engineer		
	Description	TextArea			
	~~Planned Hours~~	~~Number (16,2)~~			
	CurrencyIsoCode	Picklist	USD	USD	
	Billable	Checkbox		TRUE	

The field level migration plan is also a great place to note if any transformation is required, default values, or static values will be used.

There are many variations to the above examples of how PSA data can be fed directly into the FinancialForce Revenue Management tool. We help our clients design a solution that provides complete control and visibility of their revenues.

It's up to each client to determine what data will be migrated to FinancialForce. For example, some clients choose to migrate over timecard records and some do not. At a minimum, most clients will consider the following items for migration:

1. Active (In-Progress) Projects
2. Assignments for active projects
3. Milestones
4. Resource records

And if you have a lot of data that needs transformed, we have experts who can help with automating parts of the migration using SQL scripts, etc.

Conclusion

We hope you find this guide useful as you formulate an implementation plan. If you need help designing & implementing or reviewing your existing solution - Email us at info@cldpartners.com **We will make your Certinia implementation a success.**

SUMMARY OF
COMMON PITFALLS

A CHECKLIST

CLD Partners

Summary of Pitfalls

#1 Project Execution Pitfalls
1. Not engaging all the stakeholders at the beginning of the project
2. Underestimating the time investment needed from key stakeholders & resources
3. Not clearly understanding decision making structure (boards) or processes
4. Underestimating the time needed for decision making
5. Overlooking downstream or upstream system impacts
6. Incomplete requirements before design
7. No representation from other geographies or business units during design
8. Low engagement from those participating in design
9. Not having a dedicated environment for the Certinia project
#2 End to End Solution & Integration Pitfalls
10. Only focusing on part of the solution/not considering the upstream & downstream processes
11. Trying to over-automate or over-complicate the solution without taking time to test how it works together
12. Making too many changes at one time
13. Misalignment of Core Enterprise Processes that interact with Certinia
#3 Training & Change Management Pitfalls
14. Not providing up front training to stakeholders on what Certinia can do
15. Not having a clear mandate that users must move to the new system
16. Not preparing a training environment or plan for end user training
17. Not communicating to users about the new implementation
18. Assuming that the internal training team will be able to develop training
#4 Testing Pitfalls
19. Poor test preparation/management (Not Writing Test Cases)
20. Shortening Time for User Acceptance Testing if the schedule slips
21. Not Training Testers on the environment prior to testing
22. Not conducting System Integration Testing for Integrations
 # 5 Data Migration Pitfalls
23. Underestimating time it takes or the transformation that will be needed to migrate data
24. Thinking that it can be handled independently of the solution design
25. Waiting too long to start Data Migration Planning and Tasks
26. Assuming that the source data won't need to be cleaned manually
27. Assuming that the source data contains all the necessary information for PSA (it usually doesn't and this means that you will have to make decisions about how to supply that "missing" data.)

Summary of How to Avoid Pitfalls

#1 Project Execution Recommendations
1. When possible, identify one leader / decision maker
2. Identify the stakeholders that will be affected by the change
3. Set expectations for participation and confirm project schedule is realistic
4. Engage all your stakeholders up front
5. Invite stakeholders to design sessions & design review
6. Have a dedicated development environment for the Certinia implementation
#2 End to End Solution & Integration Pitfalls
7. Understand how Certinia will integrate with your CRM and/or quoting solutions
8. Think about how PSA and revenue recognition processes will sync
9. Determine how you will bill & invoice clients
10. If you're not sure how it should work, consider postponing automation
11. Be wary of over customizing your solution
#3 Training and Change Management Recommendations
12. Train the core project team up front on what Certinia can do
13. Create a simple Change Management Plan
14. Communicate regularly with end users at key milestones
15. Try not to over complicate the end user experience
16. Be willing to re-engineer a process
#4 Testing Recommendations
17. Write Test Cases
18. Determine which environment to use for testing
19. Identify Testers and ensure that all necessary roles and stakeholders are represented
20. Plan for staging data to support testing
21. Set & manage user expectations (define how user feedback will be addressed)
22. Ensure there enough time set aside for testing
#5 Data Migration Recommendations
23. Develop a Data Migration Strategy at the beginning of the Project
24. Don't wait to look at what data you want to migrate over
25. Create a field level mapping plan for data migration