Project Lifecycle Pro

The Unvarnished Truth About Project Lifecycle Pro's Technical Design

The technical details of Project Lifecycle Pro (PLP) for the reader who is Salesforce and IT savvy. The target audience is a Salesforce admin/developer, or an IT manager/analyst that is exploring building Salesforce-native project management capabilities within their company. Particularly, if management is interested in a consolidated view of projects or are interested in establishing a project management office (PMO). I'll discuss the technical details of how PLP fits into an enterprise-wide project management solution and how it relates to other apps. I'll also discuss PLP's broad customization options. Lastly, I'll make an argument for PLP's price to value.

What is PLP and What It Isn't

PLP is intended to form the foundational custom project objects for an enterprise-wide project management solution. It was built with the intent to handle internal improvement projects like: six sigma projects, product development, mergers & acquisitions, re-engineering efforts, capital improvements, etc. It was also built with a heavy focus on supporting the IT system aspects of these projects, so it will support: custom development, package installs and system upgrades. It was also built to allow customer's to heavily customize it for their project management process and to integrate with existing apps or future PLP add-on applications to expand project management-related capabilities.

It is **not a scheduling tool**. Scheduling and creating work plans is only one aspect of project management and there are already plenty of existing tools. I don't have any intent on re-building a Primavera, MS Project or any of the Salesforce-native tools like Inspire Planner or TaskRay. I do plan on building standard integrations with scheduling tools in the near future so that users can view all their assigned tasks within a single location in Salesforce.

It is **not a professional services automation (PSA) tool**. If your business is providing services to clients at an hourly rate (i.e. consulting, engineering, architecture, etc.), and you want to manage these client service projects then PLP is not for you. There are already several Salesforce-native solutions which specialize in serving this industry: CloudCoach, Certinia are some examples.

What is Under the Hood

PLP consists of 22 custom objects, 14 custom tabs, 60+ custom flows, standard dashboards and reports. At the core is a Master Project object with master-detail or lookup relationships to related objects. Related objects are items like team members, milestones, risks, releases, requirements, components, impacts, etc.

PLP has been built with a data architecture and automation to support administrative functionality which is critical for a robust enterprise-wide solution:

- **Easy Project Sharing** - Rather than project managers having to be the owner of a project record and manually sharing access to the project, they can control access through some simple checkboxes. A single click and the project is marked public or confidential. Every Salesforce user has read-only access to a public project. To access a confidential project record and its related objects a user has to be added as a project team member. Project team members can be granted read-only versus edit rights with the click of a button.
- **Separate Sharing of Project Financials** - In my experience, companies don't want broad access to project financials. They don't want to broadly share some costs or sensitive benefits like labor reductions. This requires that the financials objects are separate from the Master Project object and related through a lookup with separate sharing automation. This allows only certain project team members access to financials, and they can be granted either read-only or edit rights.
- **Standalone Project Dependencies** - To manage dependencies between projects, project managers from both the supplying and dependent projects need access to the dependency record. This requires the dependency object to be related to both projects, yet have separate sharing automation so both project managers can access the dependency record though they may not have full access to both projects.
- **Flexible Rollup Capabilities** - PLP supports planning at three levels: program, project and release. Therefore, the functionality is required to rollup financials and other project metrics to all three levels. However in my experience, project managers often don't have all details in early project planning and want to use general estimates rather then exact summations from detailed records. This requires that the rollup functionality can be turned on or off. Therefore, automations have been built which can be manually controlled by a checkbox and are automatically updated as records are updated or added. Automated rollups have been designed to handle up to 10,000 records which should far exceed the requirements of any single project.
- **Associating Tasks to Projects** - A custom field has been added to the standard Salesforce task object, so it can be assigned to a PLP project. Automations had to be built to assign the master project to a task when a task is created under and associated with a project's related object.

Reporting and Dashboards

Extensive work was done so that PLP customers would have a functioning set of reports and dashboards right out of the box. Report views have already been built for every custom object, so users can self-serve to build custom reports and don't have to submit an IT ticket get a report view built. Sixty-plus reports are provided with the install to support the project management process and dashboards.

Three standard dashboards are provided with the app. These dashboards are customized for the project manager, sponsor and product owner roles. A nifty feature is that the dashboard content is automatically filtered for active projects where the user is assigned the particular role. For example, a project manager only sees data for their active projects. No need to manually update all the filters on the dashboard's underlying reports to add/remove projects as projects are started or completed.

Customization Options

Like all Salesforce-native apps, PLP is highly customizable. I've being doing project management for over twenty years and know that every company has their own twist on project management. Therefore, we've built PLP with this in mind.

Custom Fields and Page Layouts

Of course, you can add your own custom fields to all of PLP's custom objects. This is going to be most likely in the Master Project object, where you have your own custom meta data you want to track on projects. Like always, you can also customize the page layouts.

Custom Project Types

You can also add your own project types and project phase definitions. You simply need to add a custom report type to the Master Project object and use the Salesforce phase functionality to associate it with the report type. Once you have the custom report type, you can also customize dropdown lists or page layouts based on the project type.

Custom Rollups

In the case of rollup automations, we've allowed customers to clone the standard flows and customize them. This is in case you add a custom financial field or other project metric which needs to be rolled up to the project, release or program level. You can clone the standard rollup flows, and modify them to include the custom fields. You would then deactivate the standard flow.

Scheduling Tool Integration

We plan to build standard integrations with scheduling tools, but until then customers can fairly easily build their own integrations. In the case of a Salesforce-native scheduling tool, you simply need to add a PLP Master Project lookup field on the scheduling tool's project object. This will allow you to relate the schedule to the PLP project.

You can also then build automations to duplicate the scheduling tool tasks as standard Salesforce tasks associated to the PLP project. This will give users a single view to all other their assigned tasks in PLP and Salesforce. Just be aware, that in most cases the users will have to have a scheduling tool license to view the tasks that are sourced from the scheduling tool.

For external scheduling tools, you'll need to associate the PLP project ID and Salesforce user IDs to the schedule and then import the schedule tasks with this ID. You can either utilize APIs or periodically upload the data.

Price to Value

The standard price for PLP is $10,000 per year per company. There are discounts for small businesses and nonprofits. This is **5-10x less** than what it would cost to build the same functionality yourself, depending on whether you use internal developers or a Salesforce consulting firm. Sure, you could build the project object and related objects fairly easily, but the automated flows are a different story.

A site license is used because it best suits the intent of providing a single enterprise-wide solution for all projects. Anyone can submit a project idea. Anyone can view public projects. Anyone can participate on a project team. Anyone can run reports on project data. Most other Salesforce-native project management apps are using a seat license model, and they require a license for anyone viewing project data or assigned a task. This can become expensive very quickly. Overall, the site license caps the cost as your company grows and promotes wide adoption.

The cost is also a pittance when compared to the amount even medium-sized companies spend annually on projects. It requires only a fraction of a percent in productivity gains to justify the spend. It is also a reasonable cost for ongoing support and enhancements.

Conclusion

Project Lifecycle Pro has the data architecture and automation required to form the foundation of an enterprise-wide project management solution. It supports advanced sharing and rollup functionality that project managers need. It is also highly customizable to fit your company's specific project management process. For more technical details, review the admin manual in the links below. Please contact me with any additional questions.