

CA Agile Central for SAP® Delivery

In today's fast-changing, competitive landscape, pre-defining a fixed scope across a multi-year SAP® implementation can result in an implementation that does not meet client requirements. For many organizations, this results in extensive customizations, fragile integrations and—ultimately—a poor user experience.

On the contrary, the agile variable-scope approach allows you to adapt to changes, risks, dependencies and feedback while still delivering working software on schedule. You manage risk by ensuring the teams always work on the highest-value features first. When the unexpected happens—as it always does—you'll still have delivered the highest-value features.

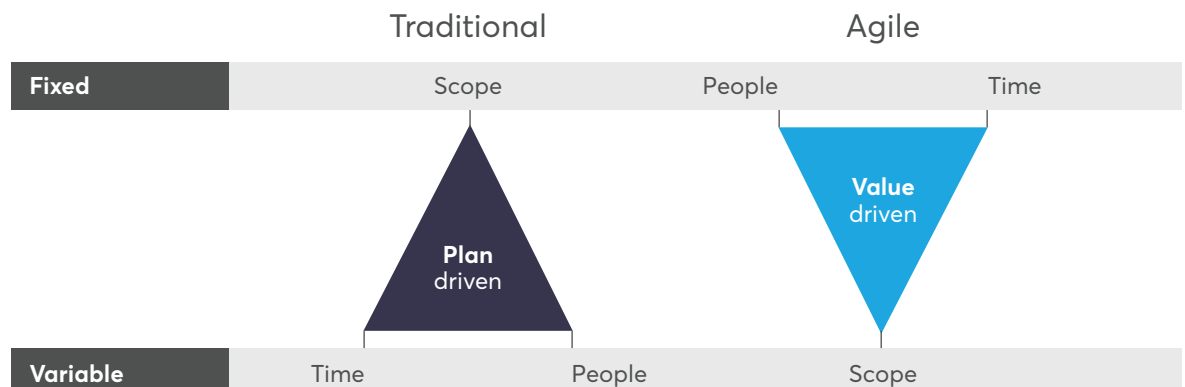
Agile vs. Waterfall for SAP Delivery

All projects need to be managed within common constraints: scope, people (resources) and time. Waterfall, the traditional approach to SAP implementations, fixes the scope so time and resources can be planned and controlled. In contrast, agile methodology assumes resources and time are fixed and sees scope as variable.

Traditional Development	Agile Development
Plan-driven	Value-driven
Fixed scope	Variable scope
Variable people (resources)	Fixed people (resources)
Variable time	Fixed time

Key Benefits

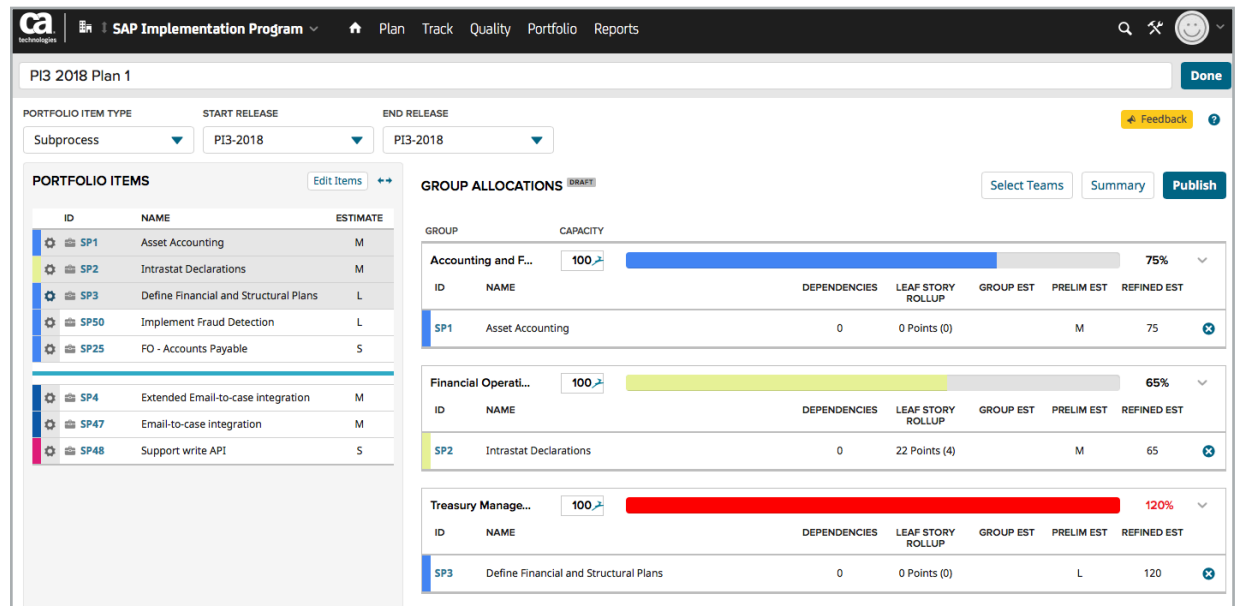
- Reduce costs of delivery and reduce risks of project failure
- Deliver the highest-value features first, accelerating time-to-value
- Stronger engagement with and input from business stakeholders
- Higher-quality products aligned to business outcomes



Best Practices When Planning an Agile SAP Implementation

This section will describe key considerations for planning an agile SAP implementation based on system attributes that are unique to SAP and other large packaged ERP systems. Best practices for planning an agile SAP implementation include:

- **Prioritize work according to highest value.** More than 80 percent of SAP configuration work is pre-defined and has been implemented in other organizations. This means agile teams can leverage reusable, pre-built content to accelerate program delivery. However, just because user stories and system features may be pre-defined doesn't mean that every organization should implement those features in the same sequence. Each customer should sequence its implementation based on delivering the highest-priority features first.
- **Map interdependencies and sequence work appropriately.** Core modules of SAP often rely on a common set of application services (authentication, reporting, database, etc.). This means there are often significant interdependencies across modules. When planning an SAP implementation, a key step should be to identify all interdependencies over the course of an upcoming release (usually a quarter), then sequence work items in a manner that accommodates those interdependencies. Doing so will reduce program delays and ensure the continuous flow of usable software.
- **Focus on delivery team composition and organization.** Agile best practices recommend teams are persistent, cross-functional and self-organizing/self-managing. For SAP implementations, there should be no exception. Teams should be organized around a consistent set of features or business value streams so they can develop strong competencies in a specific functional area. For example, a larger SAP implementation may have a team focused on the finance to management business process, and another focused on quote to cash. Smaller implementations may organize teams around specific SAP modules (e.g., finance, HR, IBP, etc.).
- **Ensure strong program management for large, complex programs with multiple delivery teams.** As with any SAP implementation, agile SAP implementations are complex and often managed by a large, geographically distributed workforce. As such, strong program management practices and tooling that promotes traceability and accountability are essential. Agile tooling should support a teams of teams portfolio hierarchy to ensure program level visibility yet provide teams with a view of information that is only relevant to that team.
- **Synchronize sprint and release planning cadences across teams.** High levels of interdependencies across SAP modules also require high levels of coordination across teams. To ensure that work can be easily re-sequenced to accommodate newly identified interdependencies without interrupting sprints that are in progress, program teams should organize sprints and releases to be synchronized with all other program teams.

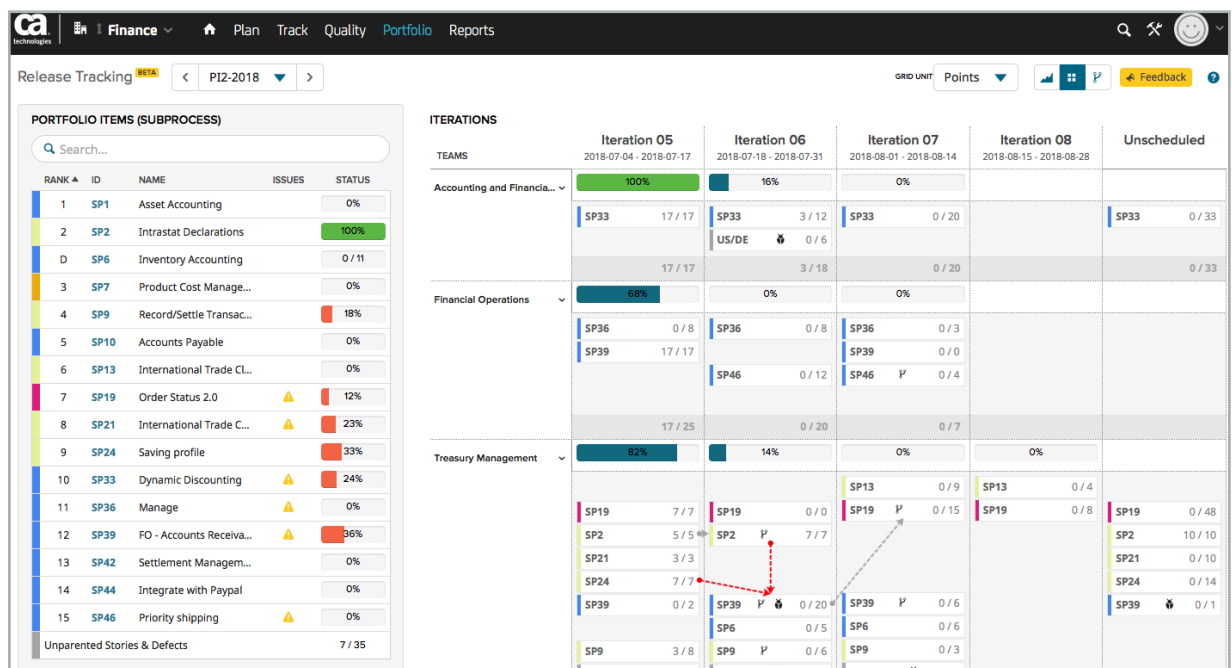


Why CA Agile Central for SAP Delivery?

CA Agile Central has specific features that make it a very good fit for managing large, complex program delivery. Agile SAP implementations are no exception.

- Dependency identification and mapping.** Identifying interdependencies and sequencing work accurately is one of the most important aspects of a successful SAP implementation. Agile tooling should be able to track cross-train dependencies in an easy-to-access and easy-to-interpret manner. Teams working in isolation or without appropriate visibility into these interdependencies risk delaying the overall program by inappropriately sequencing work. CA Agile Central supports dependency identification and mapping that can roll up from user stories to features to higher-level portfolio items. This makes it simple for a program manager to identify dependencies at a program level, or for individual teams to understand the status of user stories.
- Coordinated cross-team planning.** Given the highly integrated nature of SAP systems, it is critical that implementation teams execute sprints with synchronized start and end dates. CA Agile Central allows teams to create consistent, synchronized sprint and release cadences across multiple agile teams. This ensures that teams are aligned and shipping software predictably to facilitate integrated solution demos and PI planning, and to implement any course corrections due to unfinished work or previously unidentified interdependencies. Teams can also work with whatever methodology they prefer (SAFe, Scrum, Lean, etc.) and have metrics that are centrally viewable and require no data cleansing.
- Persistent cross-functional teams.** CA Agile Central allows you to create a hierarchy to represent multiple teams that roll into higher-level release trains or programs. This structure ensures that teams can own and manage their own backlogs and keep a consistent flow of work scheduled without losing reportability or metric roll-up for SAP.
- Fixed portfolio structure and ordinality.** Work items in CA Agile Central belong to the portfolio hierarchy. As a part of this hierarchy, work items can be grouped into a strategic category (usually represented by Initiative or Epic) or an execution category (user stories, tasks, etc.). CA Agile Central enforces a 1:1 relationship between parent and child work item types, ensuring a consistent definition of work across all teams and enabling reportability on a very large scale without limiting the way teams manage their own work. In fact, because CA Agile Central decouples the data from its presentation, teams can use whatever methodology they prefer without losing the ability to roll metrics up to the program level.

- **Consistent status mapping.** CA Agile Central allows teams to define their own statuses (called flow state) to accommodate diversity across agile teams. At the same time, it enables all team-defined flow states to be mapped back to a consistent set of scheduled states, which creates a uniform “definition of done” across teams.
- **Sharable dashboards to radiate status across teams.** CA Agile Central provides all members of the program team, from executive to developer, with a real-time view into the status of all work items, from an individual sprint to an overall program status view. Shareable dashboards improve transparency, which leads to better outcomes because awareness between teams impacts decision making and productivity.
- **Managed project risk.** CA Agile Central manages risk as a core capability and can provide dependencies across sprints to get a high-level view into project risk. Managing backlog, value delivery and priorities enables program managers to view changes against risk tolerances.



For more information, please visit ca.com/ca-agile-central

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments. Learn more at ca.com.

Copyright © 2018 CA. All rights reserved. SAP is the trademark or registered trademark of SAP SE in Germany and in several other countries. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. This document is for your informational purposes only. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document “as is” without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or noninfringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, business interruption, goodwill or lost data, even if CA is expressly advised in advance of the possibility of such damages.

200-390900_0918