SmartSuite

PLAN

Setting Your SmartSuite Implementation Up for Success

Executive Summary

The success of a SmartSuite implementation is largely determined before any configuration work begins. The Plan phase exists to ensure teams clearly understand what they are building, who they are building it for, and what the solution must ultimately deliver. While it is tempting to start configuring solutions, tables and workflows immediately, neglecting to properly plan your implementation often results in rework, inconsistent data, and solutions that fail to support real operational or reporting needs.

A strong planning approach starts by defining what the organization needs to get out of the solution, particularly the reporting, insight, and assurance required by leadership and management. From there, teams can work backwards to determine the data that must be captured, how it should be structured, how records relate to one another, and who needs access to what information. Planning also forces clarity around workflow sequencing, handoffs between roles, and how users are guided through each step of the process.

This white paper focuses on the Plan phase as the foundation for effective SmartSuite implementations. It outlines a practical, best-practice approach to planning a workflow, including defining desired outcomes, identifying stakeholders and roles, distinguishing current- and future-state processes, establishing scope and governance guardrails, and producing clear planning deliverables. The paper also highlights common planning pitfalls and the downstream impacts they can create when this phase is rushed or incomplete. To make these concepts actionable, it includes planning checklists and a mini-case example centered on findings, issues, and action plan management.

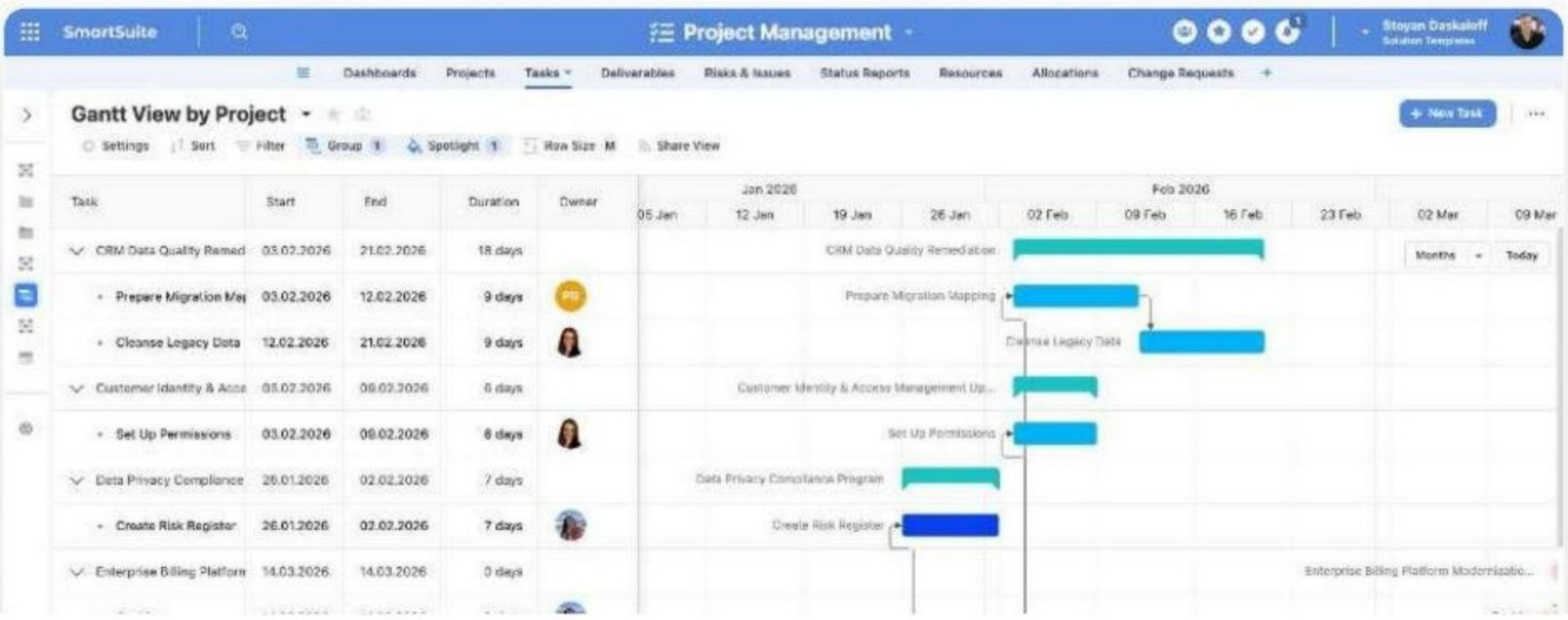
This document is intended for services consultants, implementation partners, and client administrators responsible for designing SmartSuite solutions that are usable, scalable, and trustworthy. For this audience, the Plan phase is critical because it creates alignment on goals, ownership, and expectations before design decisions are locked in. By applying the guidance in this paper, implementation teams can establish a shared blueprint that reduces rework, improves adoption, and increases confidence in the outcomes the solution is meant to deliver.

Why the Plan Phase Matters

The Plan phase is where implementation teams create alignment before committing to design and configuration decisions that are difficult and costly to reverse. This phase brings the right people together to agree on goals, responsibilities, terminology, and measures of success, and to establish a shared understanding of how the solution is expected to work. It encompasses stakeholder identification, outcome definition, current- versus future-state analysis, scope setting, and agreement on ownership and stewardship before build activities begin.

The importance of the Plan phase lies in its ability to establish clarity and confidence early. By aligning on what the process is meant to achieve, how success will be measured, and what information must be captured, teams can make informed decisions during later phases around data structures, workflows, permissions, and user experience. This upfront alignment directly improves solution quality, usability, and trust in the outputs the system produces.

When planning is rushed or treated as a formality, the consequences are often immediate and cumulative. Teams begin building on incomplete assumptions, critical stakeholder input arrives too late, and foundational questions around data, workflow behavior, and ownership surface only after configuration is underway. This typically results in rework, inconsistent data capture, misaligned reporting, and frustration for both users and leadership, shifting effort downstream where changes are more disruptive and expensive.

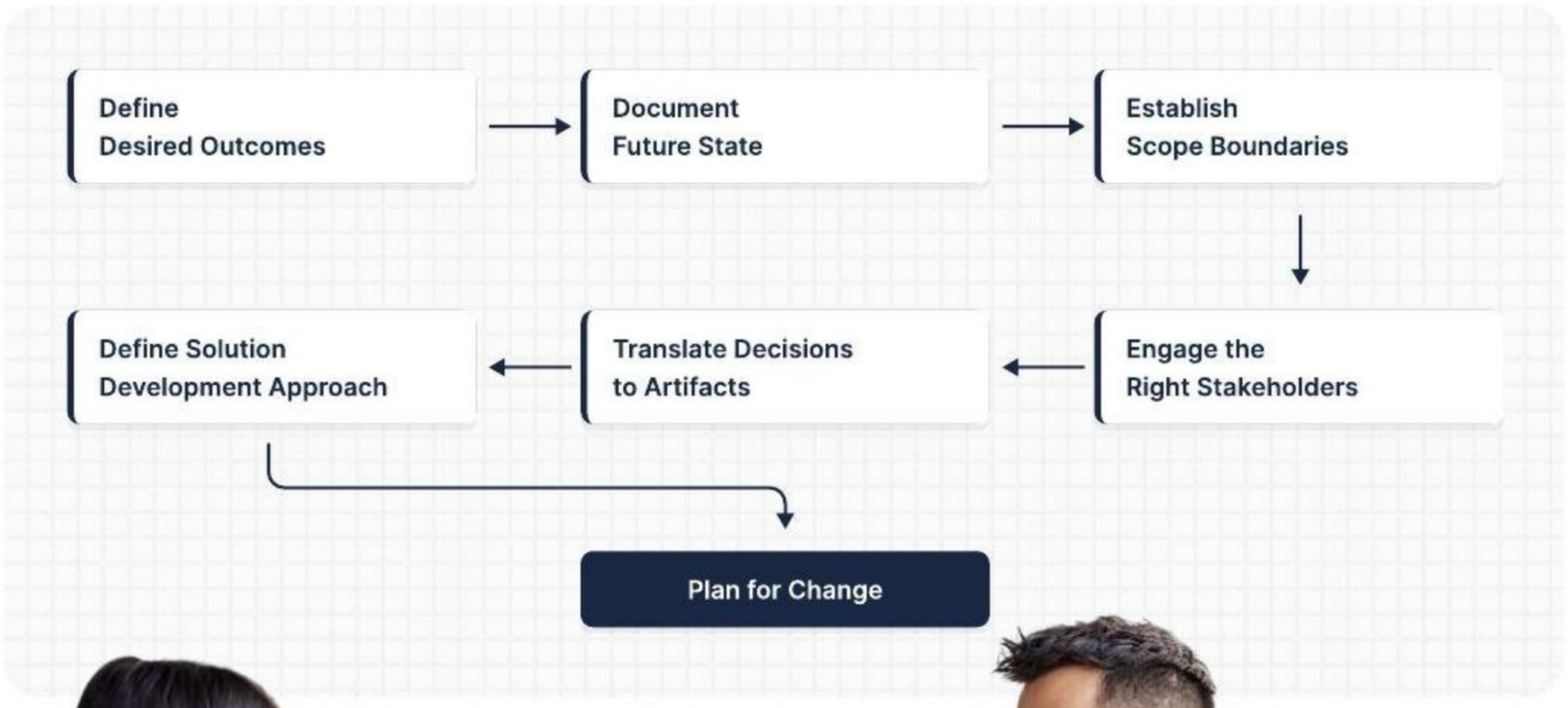


In addition, the Plan phase provides a critical opportunity to challenge assumptions and intentionally design for improvement. By explicitly distinguishing between current-state practices that should be preserved and those that should evolve, teams avoid carrying forward inefficiencies simply because they exist today. The result is a shared, well-understood blueprint that reduces risk, accelerates implementation, and positions the solution to scale and adapt over time.

Planning Phase Best Practices

The Plan phase best practices outlined below are informed by extensive implementation experience across a wide range of SmartSuite use cases and customer environments. Organizations that invest time in these planning activities consistently experience smoother builds, faster adoption, stronger reporting outcomes, and fewer downstream corrections during deployment and support.

While these best practices provide a proven foundation, they are not intended to be prescriptive in every scenario. Each organization must fully understand its own objectives, constraints, and operating context when planning a solution. The most successful implementations apply these practices thoughtfully, adapting them to specific needs while maintaining the underlying discipline that planning requires



PHASE 1

Define Desired Outcomes and Decision-Driven Reporting Upfront

Planning should begin by clarifying what the organization ultimately needs to get out of the solution. This includes the questions leaders need answered, the insights required to manage risk or performance, and how success will be measured. When outcomes and reporting needs are defined first, teams can work backward to determine the data, workflows, and controls required to support those objectives, rather than simply capturing activity for its own sake.

For example, if executives need confidence that issues are being identified, prioritized, and remediated on time, planning should explicitly define what metrics, statuses, and evidence are required to support that assurance.

PHASE 2

Document the Future-State Process, Not Just the Current State

While understanding the current process is important, planning should focus on the intended future-state workflow. Teams should explicitly identify which aspects of the existing process should be preserved, which should be improved, and which should be eliminated altogether. This prevents inefficient or manual practices from being unintentionally carried forward into the new solution. Effective planning treats SmartSuite as an opportunity to improve how work gets done, not simply digitize what exists today.

PHASE 3

Establish Clear Scope Boundaries and an MVP Definition

A well-defined scope is critical to maintaining momentum and focus. Planning should clearly articulate what will be included in the initial release versus what will be deferred to later phases. Defining a minimum viable product (MVP) allows teams to deliver value quickly while preserving a clear roadmap for future enhancements. Without clear scope boundaries, solutions tend to grow uncontrollably during build, increasing complexity, delaying launch, and diluting quality.

PHASE 4

Engage the Right Stakeholders with Clear Decision Ownership

Strong planning depends on involving the right stakeholders at the right time, and being explicit about decision authority. Teams should identify who provides input, who is accountable for outcomes, and who has final decision-making responsibility. This clarity avoids stalled progress, conflicting direction, and late-stage redesign. Planning is not about achieving consensus on every detail; it is about enabling timely, informed decisions with shared accountability.

PHASE 5

Translate Process Decisions into Concrete Planning Artifacts

Planning should produce tangible, durable artifacts that directly inform later phases. These may include workflow diagrams, role and access definitions, success metrics, data capture requirements, and governance rules. Well-documented artifacts serve as a shared reference point for the modeling, logic, permissions, and UX design phases. If a decision cannot be clearly documented, it is often a signal that it has not been fully thought through.

PHASE 6

Define the Solution Development Approach Early

SmartSuite offers multiple ways to approach solution development, and planning is the right time to determine which path best fits the organization's needs. Options may include leveraging preconfigured Solution Suites designed around industry best practices, using Solution Accelerators as a structured starting point, or building a solution from scratch to meet highly specific requirements.

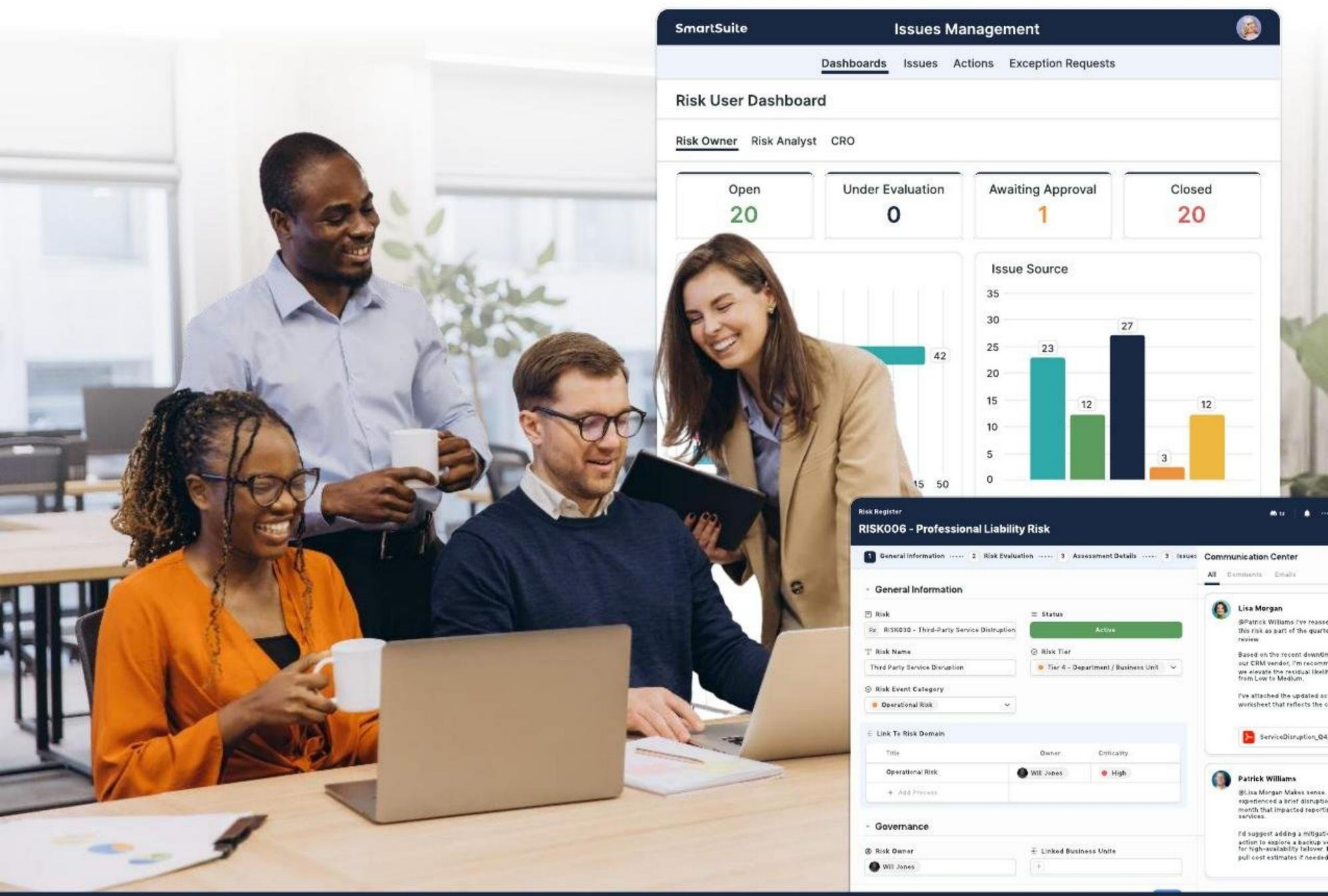
Defining this approach upfront helps set expectations, align effort, and ensure the build strategy supports both near-term goals and long-term flexibility. Planning should also consider where SmartSuite's native AI capabilities can enhance efficiency, analysis, or review activities without bypassing governance or controls.

PHASE 7

Plan with Scale, Change, and Stewardship in Mind

Effective planning anticipates growth, whether in users, data volume, processes, or reporting demands. Teams should consider how the solution will be owned, governed, and evolved over time, including who is responsible for maintenance, enhancements, and ongoing data quality.

Planning for scale and change upfront reduces the need for disruptive structural changes later and ensures the solution remains usable, trusted, and sustainable as adoption expands.



Common Planning Pitfalls

Even experienced teams can fall into planning traps when momentum, familiarity, or urgency takes over. Common pitfalls include over-scoping the initial release, relying on informal decision-making, and treating planning as a lightweight or off-platform exercise. In practice, these missteps introduce ambiguity that carries forward into design, build, and deployment, where corrections become far more disruptive and expensive.

The Plan phase is not just about discussion; it is about making decisions visible, durable, and actionable. SmartSuite itself can be used during planning to capture scope boundaries, assumptions, roles, and governance rules as structured records. Doing so creates traceability between planning intent and configuration outcomes, and helps teams avoid the pitfalls outlined below.

Skipping Outcome Definition and Success Metrics

When desired outcomes and success criteria are not clearly defined, planning conversations lack a true north. Decisions become subjective, priorities shift, and teams risk delivering a solution that appears complete but fails to produce meaningful insight or confidence for leadership.

How to Avoid It:
Start planning by defining the decisions the solution must support and the metrics that indicate success, then validate them with key stakeholders before moving forward.

Failing to Engage the Right Stakeholders Early

Excluding key stakeholders or involving them too late in the process leads to missed requirements, conflicting expectations, and late-stage redesign. Planning without broad and appropriate input almost guarantees friction once built activities are underway.

How to Avoid It:
Identify process owners, contributors, reviewers, and decision-makers upfront, and clearly distinguish who provides input versus who has final decision authority.

Confusing Current State with Future State

Simply recreating today's process in a new system locks in inefficiencies and limits the value of the implementation. Without explicitly defining what should change, teams default to familiarity rather than improvement.

How to Avoid It:

Document both current-state and future-state workflows and explicitly call out which steps will be preserved, improved, or eliminated.

Overlooking Scope Boundaries and MVP Definition

Trying to deliver everything in the initial release results in bloated solutions, extended timelines, and diluted focus. Without clear scope boundaries, planning gives way to unchecked expansion during build.

How to Avoid It:

Define a clear MVP that delivers core value and establish a roadmap for future phases before configuration begins.

Neglecting Governance Guardrails

Without agreed-upon rules for ownership, change management, and escalation, configuration decisions become ad hoc. Over time, this leads to inconsistent behavior, fragile logic, and solutions that are difficult to maintain or scale.

How to Avoid It:

Establish governance expectations early, including who owns the solution, how changes are approved, and how exceptions are handled.

Failing to Document Planning Decisions and Artifacts

Relying on informal notes or tribal knowledge causes intent to be lost as teams grow or change. Poor documentation undermines alignment and makes it difficult to validate, support, or extend the solution later.

How to Avoid It:

Capture planning outputs as structured, shareable artifacts, ideally within SmartSuite, so decisions remain visible and durable.

Underestimating the Cost of Rushed Planning

Jumping straight into configuration creates the illusion of progress while quietly accumulating risk. Every skipped planning step compounds downstream, increasing rework, delays, and frustration for both users and administrators.

How to Avoid It:

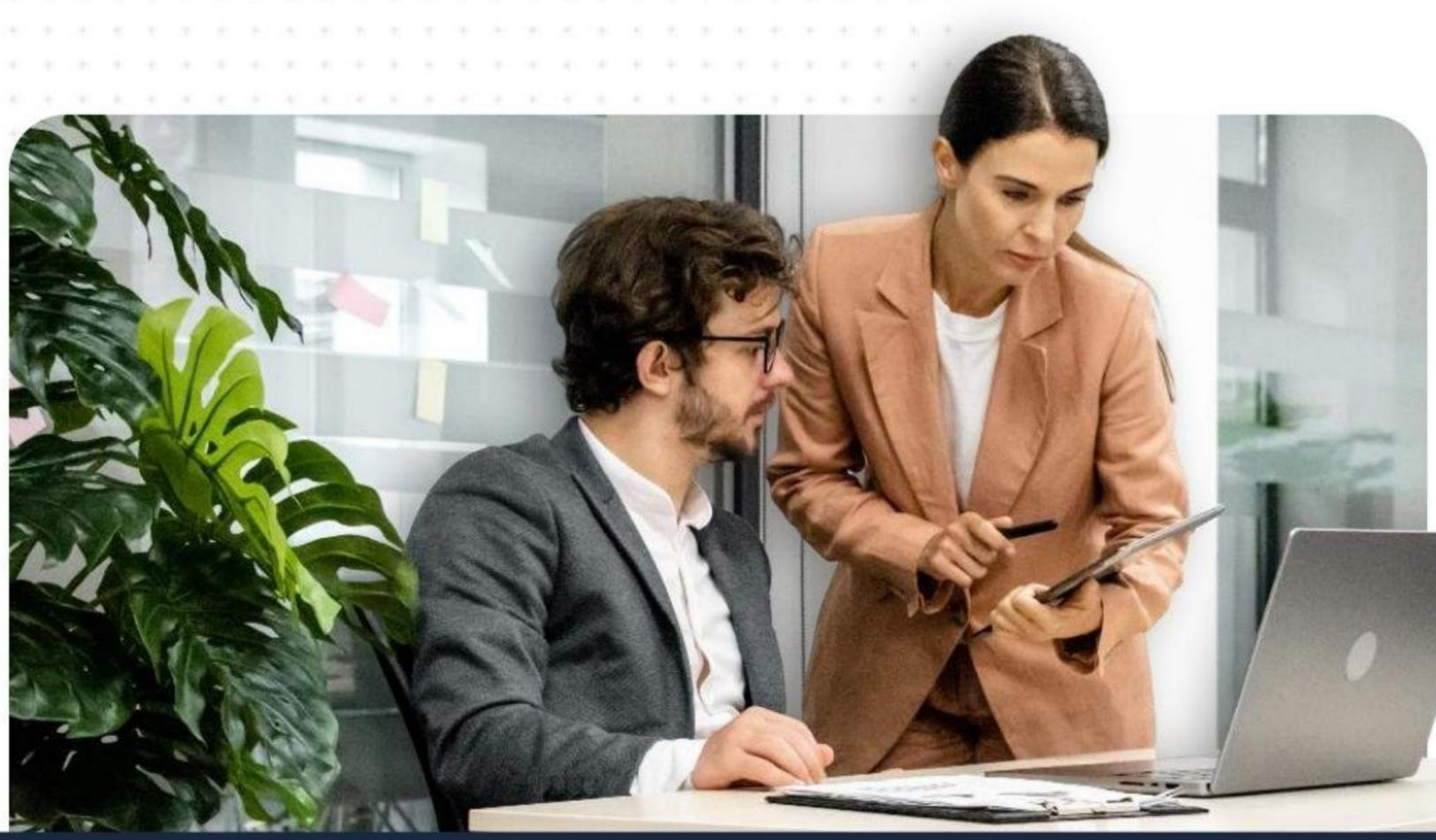
Treat planning as an investment, not a delay. Time spent aligning upfront consistently reduces total implementation effort and risk.

High-Level Planning Checklist

The checklist below captures the minimum planning artifacts required before moving into modeling and configuration. These items may be captured and maintained directly in SmartSuite to ensure visibility and continuity throughout the planning phase.

Checklist Item	Description	Benefits
Define Outcomes and Success Criteria	Clearly articulate what the solution must deliver, including key decisions it should support, required reporting, and how success will be measured. Align on these outcomes with business and executive stakeholders before discussing design.	Ensures the solution is built to produce meaningful insight and confidence, not just capture activity.
Identify Stakeholders, Roles, and Ownership	Identify all users and stakeholders involved in the process, including contributors, reviewers, approvers, and decision-makers. Define ownership, stewardship, and decision authority for the solution.	Prevents misalignment, late-stage rework, and unclear accountability during build and beyond.
Document Current-State and Future-State Processes	Capture how the process works today and explicitly define how it should work in the future. Call out steps to retain, improve, or eliminate as part of the SmartSuite implementation.	Avoids recreating inefficient processes and provides a clear blueprint for workflow design.
Define Scope and MVP Boundaries	Establish what will be included in the initial release versus what will be deferred to later phases. Document assumptions, exclusions, and dependencies tied to scope decisions.	Keeps the initial implementation focused, achievable, and aligned to delivery timelines.
Determine Data and Reporting Requirements	Identify the critical data elements that must be captured to support workflows, reporting, and oversight. Consider how data should relate across records and what information different audiences need to see.	Enables accurate reporting, clean data models, and informed design decisions in later phases.

Checklist Item	Description	Benefits
Establish Governance and Change Guardrails	Define how decisions will be made, how changes will be requested and approved, and how exceptions or escalations will be handled once the solution is live.	Protects the solution from uncontrolled changes and ensures long-term consistency and scalability.
Select the Solution Development Approach	Determine whether to leverage SmartSuite Solution Suites, Solution Accelerators, or a custom-built approach based on requirements, maturity, and timeline. Consider where AI capabilities may add value.	Aligns implementation strategy with organizational needs and avoids misaligned build approaches.
Capture and Validate Planning Artifacts	Consolidate planning outputs, process flows, role definitions, scope decisions, governance rules, into clear, shareable artifacts and validate them with stakeholders.	Creates a shared, approved foundation that directly informs modeling, logic, UX, and permissions.



Plan Use Case Example: Issue & Action Tracking

In planning an Issue & Action Tracking workflow, the organization first aligned on success metrics such as issue cycle time, ownership clarity, and executive visibility. By defining a future-state workflow and limiting the MVP to core issue and action records, the team avoided unnecessary complexity and established a clear foundation for later enhancements.

Planning Consideration	Benefit
Clarified what leadership needs from the Issues & Action Plan process, including visibility into open issues, risk severity, overdue actions, and remediation effectiveness.	Ensures the solution supports executive assurance and decision-making, not just issue tracking.
Identified issue owners, action owners, reviewers, approvers, and executive stakeholders, and defined responsibilities across the issue life cycle.	Prevents role confusion and establishes clear accountability.
Defined how issues move from identification through remediation, validation, and closure, including required reviews and hand offs.	Provides a clear workflow blueprint and avoids inconsistent or manual practices.
Determined the data required for issues and actions, such as severity, root cause, ownership, due dates, and evidence, to support reporting and oversight.	Enables accurate reporting, prioritization, and confidence in remediation status.
Agreed on which issue types, workflows, and reports would be included in the initial release versus deferred to later phases.	Keeps the first release focused and achievable while preserving a roadmap for expansion.
Established solution ownership, escalation rules for overdue or high-risk issues, and expectations for ongoing stewardship and change management.	Ensures the solution remains controlled, consistent, and sustainable over time.

Conclusion: Planning with Intent

The Plan phase sets the direction for everything that follows in a SmartSuite implementation. By clearly defining outcomes, engaging the right stakeholders, and establishing shared expectations before configuration begins, teams create the conditions for a solution that is usable, scalable, and trusted. Effective planning transforms ambiguity into alignment and ensures that design decisions are driven by purpose rather than assumption.

In practice, investing time in planning reduces downstream risk, limits rework, and accelerates overall delivery. Teams that approach the Plan phase with discipline, using structured best practices, avoiding common pitfalls, and documenting decisions as durable artifacts, enter subsequent phases with clarity and confidence. The result is not just a smoother implementation, but a stronger foundation for long-term ownership, adoption, and continuous improvement.