

USER EXPERIENCE

Enabling Day-to-Day Work in SmartSuite



Executive Summary

The success of a SmartSuite implementation is determined at the point of daily use. No matter how well a solution is planned, modeled, or automated, long-term value depends on whether users can quickly understand their responsibilities, locate the information they need, and take confident, accurate action. Poorly designed interfaces create friction, workarounds, and mistrust in the system, ultimately undermining adoption and operational outcomes.

A thoughtfully designed UX ensures that SmartSuite is more than a data repository, it becomes an operational system that actively guides work, reinforces accountability, and supports decision-making. When done well, UX design improves clarity, consistency, and efficiency, helping users navigate complex workflows, dashboards, and records with minimal friction.



This white paper provides a practical, enterprise-grade approach to shaping user experience in SmartSuite. It focuses on structuring layouts, organizing tabs and sections, applying conditional logic, designing role-based views and dashboards, and creating an intuitive navigation framework. The guidance ensures that every interaction is purposeful and aligned with both process requirements and user needs.

It is intended for solution architects, implementation partners, administrators, and business leaders responsible for adoption and operational success. By following these principles, teams can translate prior planning, modeling, logic, and permissions decisions into a solution that feels intuitive, reduces errors, and drives consistent engagement across the organization.

Why the User Experience Phase Matters

User experience is where strategy meets execution. It is the point at which users decide, often subconsciously, whether SmartSuite feels like a system that supports their work or one that introduces friction.

When UX design is neglected or treated as an afterthought, organizations commonly experience:

- Low adoption outside of power users or administrators
- Incomplete or inconsistent data entry
- Increased reliance on spreadsheets, email, or side systems
- Frequent questions, re-training, and manual intervention
- Reduced trust in dashboards, metrics, and reporting

These symptoms are rarely caused by a lack of features or automation. Instead, they stem from interfaces that do not align with how users think, prioritize, or operate in their daily roles.

Conversely, a well-designed user experience enables users to focus on decisions and actions rather than navigation or interpretation. It guides users naturally through processes, clarifies priorities, and reinforces accountability without requiring constant oversight or instruction.

The UX phase ensures that SmartSuite does not simply store work, but actively supports how work gets done, at scale, across teams, and over time.



The Role of UX in Enterprise SmartSuite Implementations

In enterprise environments, user experience carries additional weight and complexity. SmartSuite solutions often support multiple departments, distinct operating models, and varying levels of system interaction, from frontline contributors executing tasks to executives seeking assurance and strategic insight.

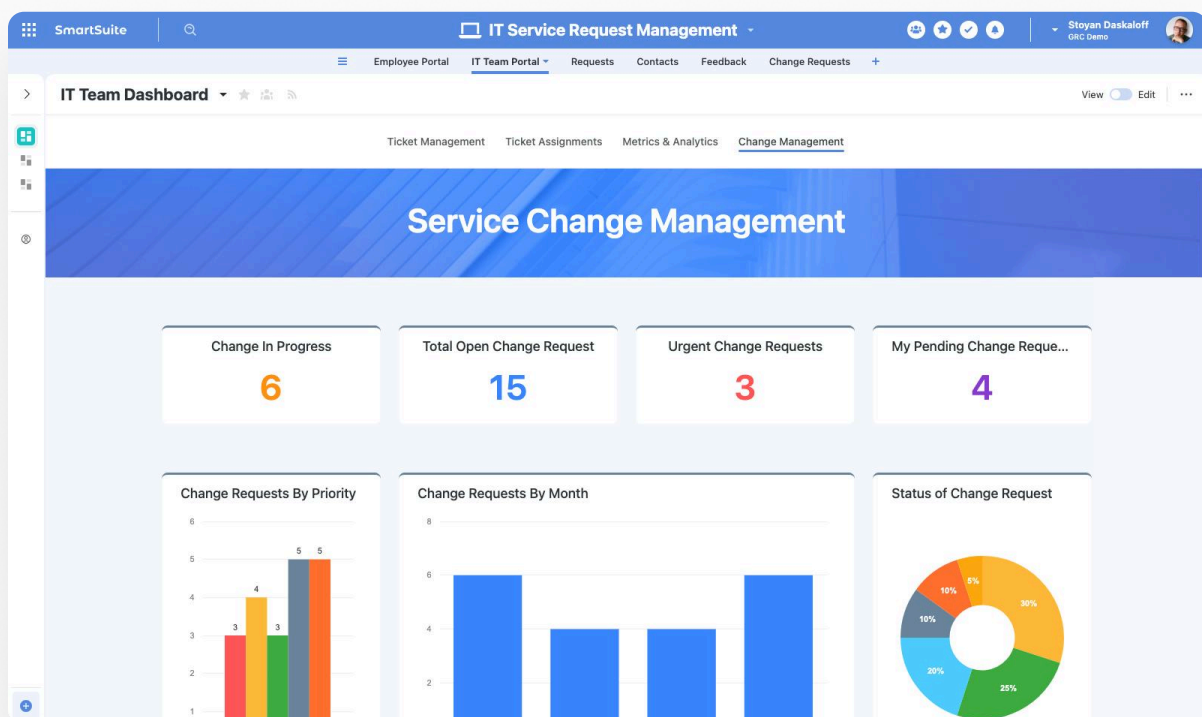
A single SmartSuite workspace may simultaneously serve:

- Contributors completing structured work
- Managers monitoring execution and resolving exceptions
- Compliance or risk teams validating controls and evidence
- Executives reviewing performance, exposure, and outcomes

Without intentional UX design, these audiences are forced into the same interface, leading to friction for contributors and an insufficient signal for leadership.

Enterprise UX design in SmartSuite must therefore balance:

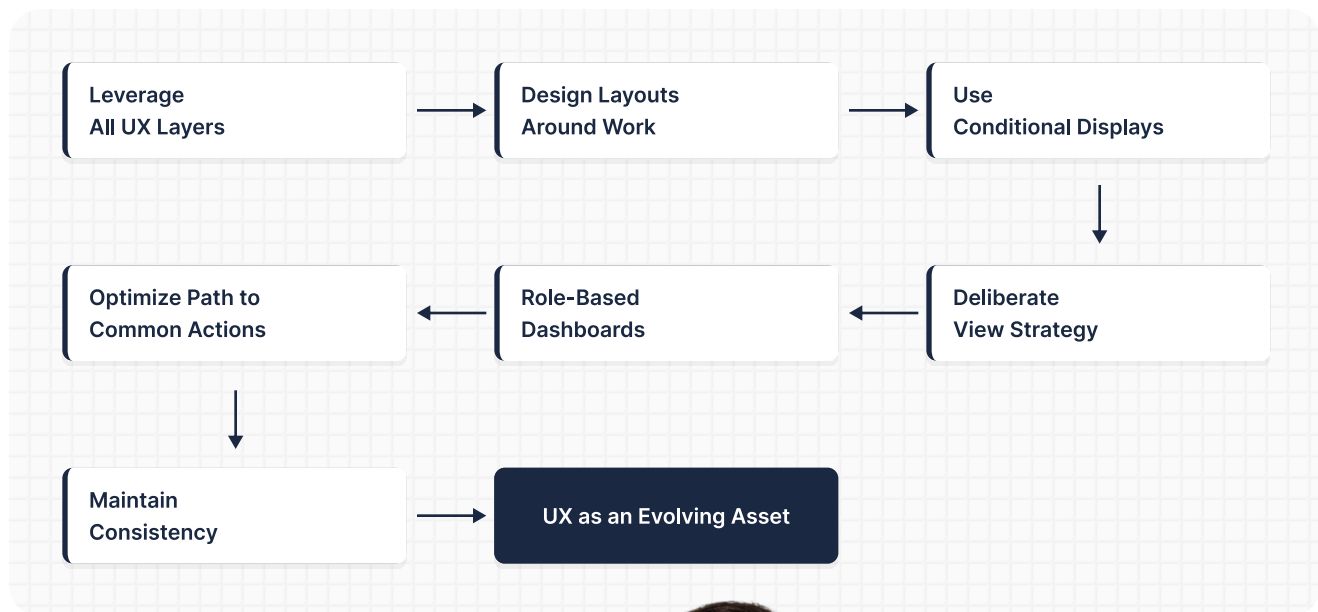
- Simplicity for everyday execution
- Visibility for management and oversight
- Consistency across teams and workflows
- Scalability as adoption grows



User Experience Phase Best Practices

The following best practices are informed by real-world SmartSuite implementations across operational, risk, compliance, project, and enterprise service environments. Organizations that apply these principles consistently experience faster adoption, higher data quality, reduced training overhead, and increased confidence in reporting and outcomes.

These practices are not theoretical design ideals. They reflect patterns that have proven effective across complex, regulated, and high-scale environments.



PHASE 1

Understand and Leverage All Available UX Layers

Effective user experience design requires using every available UX layer thoughtfully, aligning layouts, views, dashboards, and navigation to support how users actually work and make decisions.

UX Layer 1:

Record Layouts, Structuring the Moment of Work

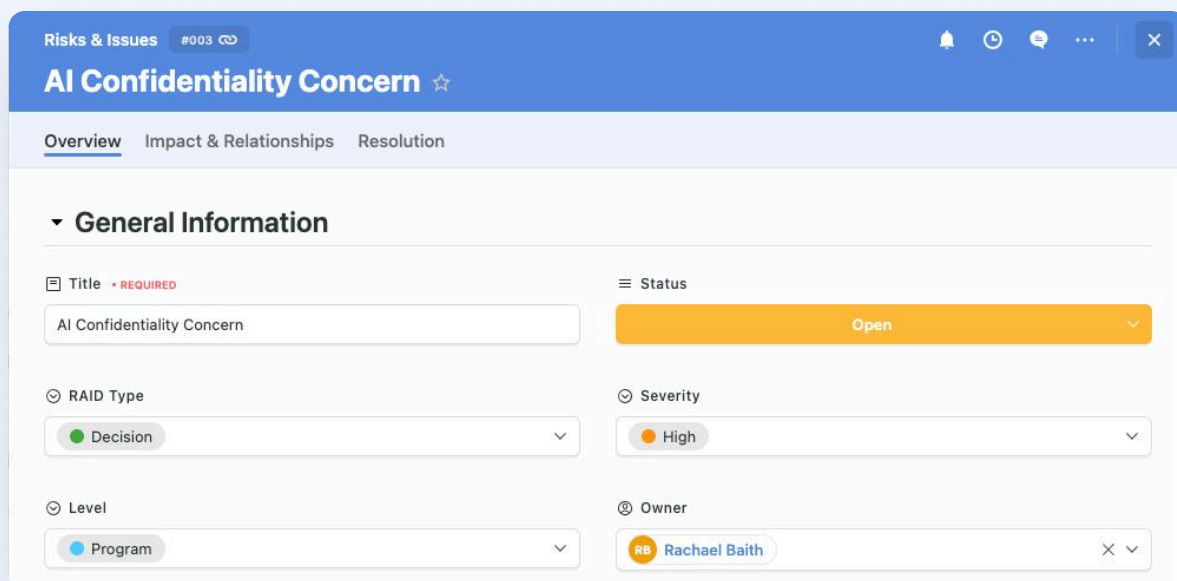
Record layouts represent the primary interaction surface for users. They define what users see, in what order, and with what level of emphasis.

Well-designed record layouts reduce hesitation, eliminate guesswork, and make responsibility unmistakably clear.

Record layouts should reflect the natural flow of work, not the underlying data structure. Users should immediately understand where to focus their attention when opening a record, what information is critical, and what actions are expected of them.

Effective record layouts typically:

- Organize content into logical tabs that mirror the work lifecycle (e.g., Overview, Details, Reviews, Actions, History)
- Place required fields, current status, ownership, and primary actions in the first tab
- Move supporting, reference, or audit information out of the primary interaction path
- Use clear section headers, spacing, and grouping to reduce visual scanning effort



The screenshot shows a record layout for 'AI Confidentiality Concern' in SmartSuite. The header bar is blue and contains the title 'AI Confidentiality Concern' with a star icon. Below the header, there are three tabs: 'Overview' (selected), 'Impact & Relationships', and 'Resolution'. The main content area is titled 'General Information' and contains several fields:

- Title** (REQUIRED): AI Confidentiality Concern
- Status**: Open (orange button)
- RAID Type**: Decision (green button)
- Severity**: High (orange button)
- Level**: Program (blue button)
- Owner**: Rachael Baith (blue button with RB icon)

A well-designed layout answers a single core question for the user: **“What do I need to do or know right now?”**

UX Layer 2:

Conditional Display, Managing Complexity Intelligently

Use Conditional Display to Reduce Cognitive Load

SmartSuite's conditional display rules are a foundational UX capability, not an advanced feature reserved for edge cases. Conditional display allows complexity to be revealed only when it becomes relevant, dramatically improving usability and data quality.

Effective use of conditional display:

- Hides fields and sections until prerequisite conditions are met
- Guides users step-by-step through complex workflows
- Prevents premature or invalid data entry
- Makes forms and records feel simple without sacrificing completeness

In enterprise solutions, conditional display is essential for managing variation across roles, scenarios, and regulatory requirements without fragmenting the solution into separate workflows.

UX Layer 3:

Views, Supporting Different Work Modes

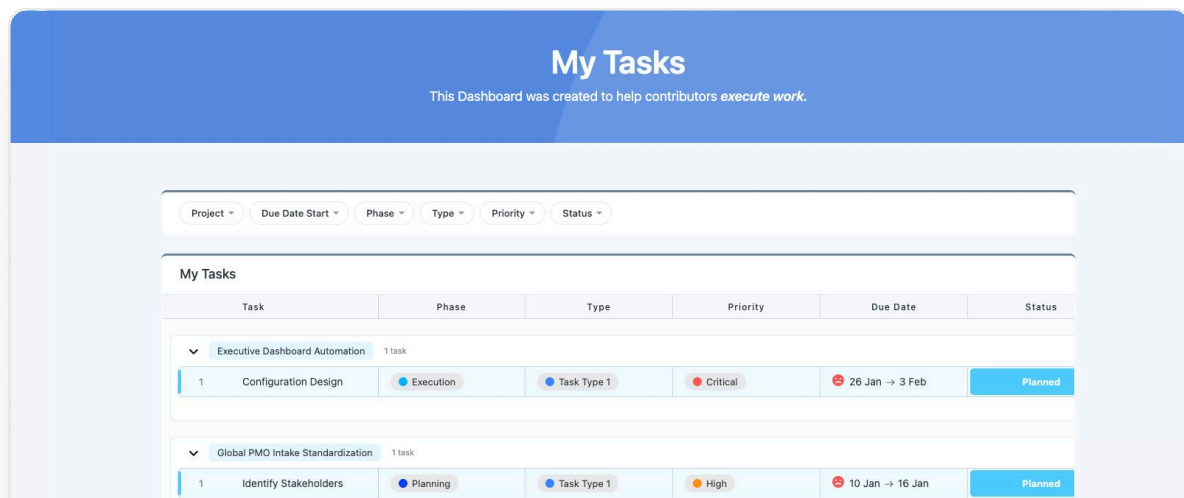
Apply a Deliberate View Strategy for Different Work Modes

Views are not interchangeable. Each view should be designed with a specific purpose tied to a user question or action.

Common work modes include:

- Execution: What do I need to work on right now?
- Exception Management: What is overdue, blocked, or at risk?
- Oversight: How is the process performing overall?
- Investigation: Why did this happen?

Queue-style grid views, filtered kanban boards, calendar views, and drilldown analysis views should each be created intentionally rather than accumulated organically. Every view should justify its existence by answering a clear operational question.



My Tasks

This Dashboard was created to help contributors *execute work*.

Project ▾ Due Date Start ▾ Phase ▾ Type ▾ Priority ▾ Status ▾

Task	Phase	Type	Priority	Due Date	Status
Executive Dashboard Automation	1 task				
1 Configuration Design	Execution	Task Type 1	Critical	26 Jan -> 3 Feb	Planned
Global PMO Intake Standardization	1 task				
1 Identify Stakeholders	Planning	Task Type 1	High	10 Jan -> 16 Jan	Planned

UX Layer 4:

Dashboards, Driving Decisions, Not Just Visibility

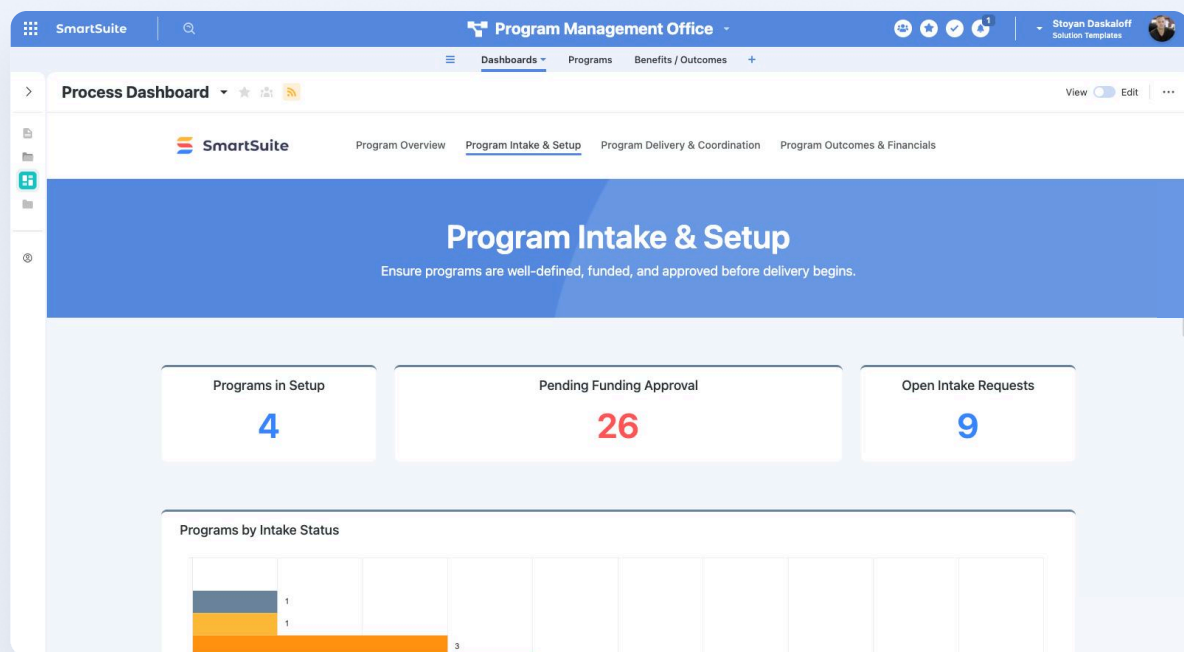
Build Role-Based Dashboards That Drive Action

Dashboards should not function as passive reporting pages. In enterprise implementations, dashboards are decision-making tools.

Effective dashboards:

- Are tailored to specific roles (contributors, managers, executives)
- Surface prioritized work, alerts, and key indicators
- Provide direct navigation into records requiring action
- Avoid overwhelming users with excessive or redundant metrics

A strong dashboard answers: **“What should I do next?”** rather than **“What happened?”**



UX Layer 5:

Navigation and Interaction Efficiency

Optimize for the Fastest Path to Common Actions

High-frequency actions should be easy to perform with minimal navigation. UX design should actively remove friction from the most common tasks.

This includes:

- Reducing unnecessary clicks and page changes
- Using buttons and quick links intentionally
- Placing actions near the information they relate to
- Designing layouts that minimize scrolling and context switching

Optimizing for speed improves adoption, reduces errors, and minimizes reliance on training or documentation.

UX Layer 6:

Consistency and Standards

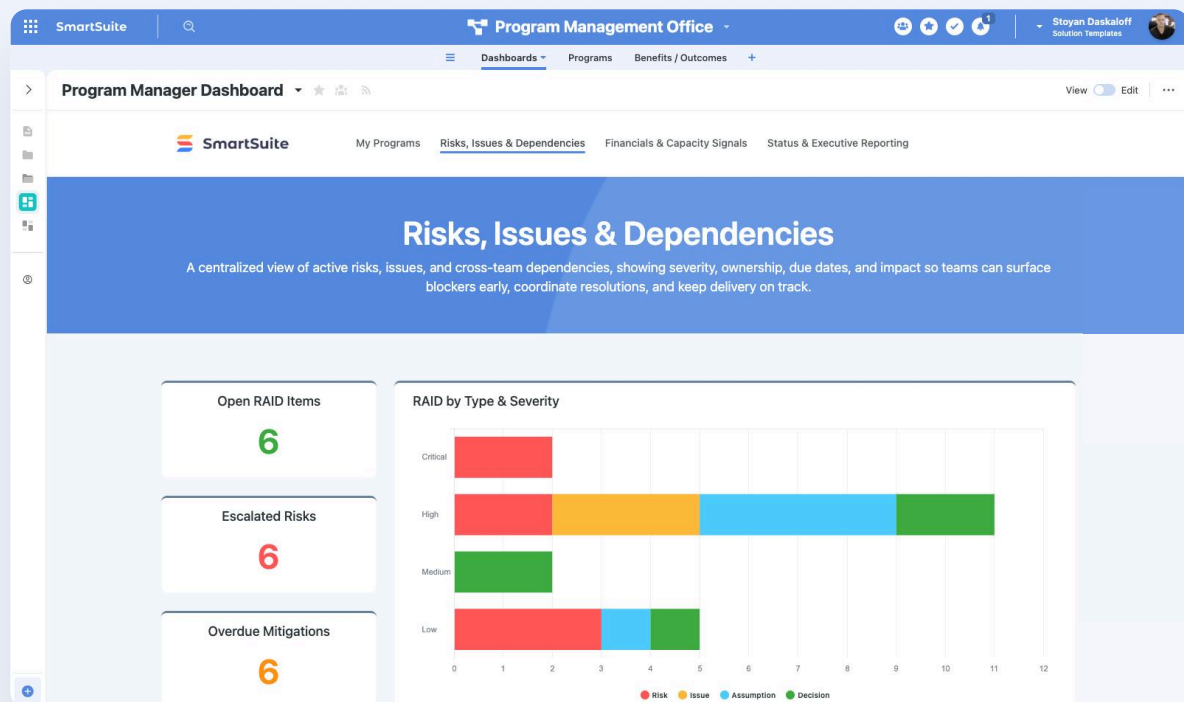
Maintain UX Consistency Across the Solution

Consistency builds trust. When users encounter predictable patterns across tables and workflows, they spend less time learning the system and more time using it effectively.

Consistency should apply to:

- Naming conventions
- Status definitions
- Layout structures
- Dashboard design patterns

Establishing UX standards early prevents confusion as solutions expand across teams and use cases.



UX Layer 7:**UX as an Ongoing Capability**

Treat UX as an Evolving Asset

User experience is not static. As adoption grows and workflows mature, UX should be reviewed and refined.

Strong UX governance includes:

Monitoring usage patterns and adoption metrics

- Monitoring usage patterns and adoption metrics
- Gathering structured user feedback
- Periodically reviewing views, dashboards, and layouts
- Adjusting designs based on real-world behavior

UX maturity increases over time when refinement is intentional rather than reactive.

The following best practices are informed by real-world SmartSuite implementations across operational, risk, compliance, project, and enterprise service environments. Organizations that apply these principles consistently achieve faster adoption, higher data quality, and more reliable reporting outcomes.

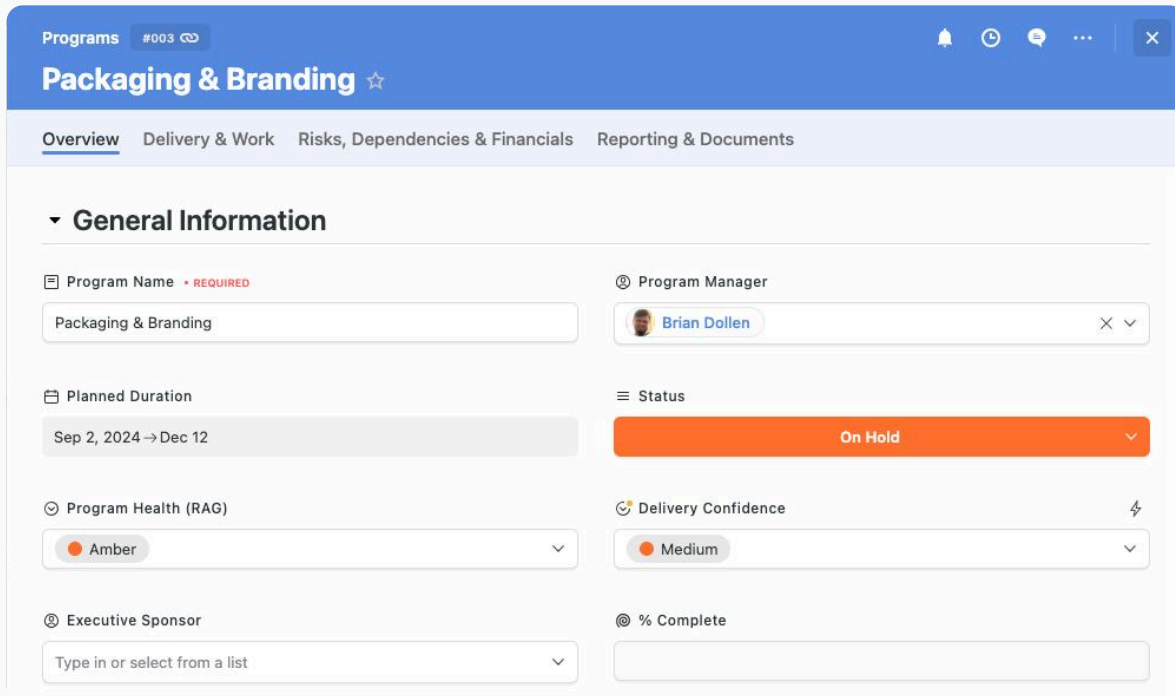
**PHASE 2****Design Record Layouts Around How Users Actually Work**

Record layouts should reflect the natural flow of work, not the underlying data structure. Users should immediately understand where to focus their attention when opening a record, what information is critical, and what actions are expected of them.

Effective record layouts typically:

- Organize content into logical tabs that mirror the work lifecycle (e.g., Overview, Details, Reviews, Actions, History)
- Place required fields, current status, and primary actions in the first tab
- Move supporting, reference, or audit information out of the primary interaction path
- Use clear section headers, spacing, and grouping to reduce visual scanning effort

A well-designed layout answers a single core question for the user: **“What do I need to do or know right now?”**



PHASE 3

Use Conditional Display to Reduce Cognitive Load

SmartSuite's conditional display rules are a foundational UX capability, not an advanced feature reserved for edge cases. Conditional display allows complexity to be revealed only when it becomes relevant, dramatically improving usability and data quality.

Effective use of conditional display:

- Hides fields and sections until prerequisite conditions are met
- Guides users step-by-step through complex workflows
- Prevents premature or invalid data entry
- Makes forms and records feel simple without sacrificing completeness

Progressive disclosure enables solutions to scale in sophistication while maintaining clarity for everyday users.

PHASE 4

Apply a Deliberate View Strategy for Different Work Modes

Views are not interchangeable. Each view should be designed with a specific purpose tied to a user question or action.

Common work modes include:

- Execution: What do I need to work on right now?
- Exception Management: What is overdue, blocked, or at risk?
- Oversight: How is the process performing overall?
- Investigation: Why did this happen?

Queue-style grid views, filtered kanban boards, calendar views, and drilldown analysis views should each be created intentionally rather than accumulated organically.

Every view should justify its existence by answering a clear operational question.

PHASE 5

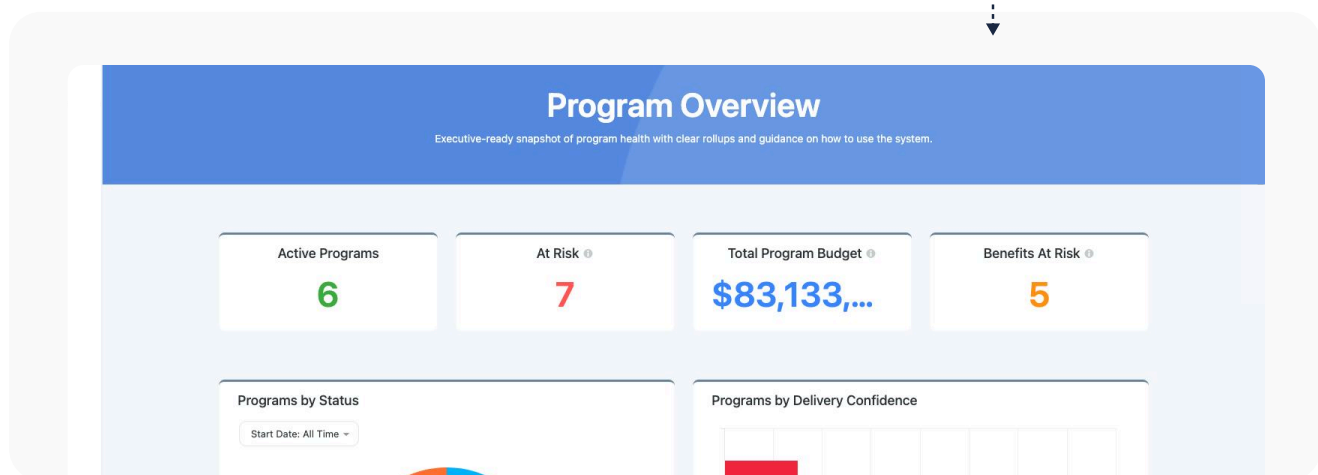
Build Role-Based Dashboards That Drive Action

Dashboards should not function as passive reporting pages. In enterprise implementations, dashboards are decision-making tools.

Effective dashboards:

- Are tailored to specific roles (contributors, managers, executives)
- Surface prioritized work, alerts, and key indicators
- Provide direct navigation into records requiring action
- Avoid overwhelming users with excessive or redundant metrics

A strong dashboard answers: **“What should I do next?”** rather than **“What happened?”**



PHASE 6

Optimize for the Fastest Path to Common Actions

High-frequency actions should be easy to perform with minimal navigation. UX design should actively remove friction from the most common tasks.

This includes:

- Reducing unnecessary clicks and page changes
- Using buttons and quick links intentionally
- Placing actions near the information they relate to
- Designing layouts that minimize scrolling and context switching

Optimizing for speed improves adoption, reduces errors, and minimizes reliance on training or documentation.

PHASE 7

Maintain UX Consistency Across the Solution

Consistency builds trust. When users encounter predictable patterns across tables and workflows, they spend less time learning the system and more time using it effectively.

Consistency should apply to:

- Naming conventions
- Status definitions
- Layout structures
- Dashboard design patterns
- Visual cues and terminology

Establishing UX standards early prevents confusion as solutions expand across teams and use cases.

PHASE 8

Treat UX as an Evolving Asset

User experience is not static. As adoption grows and workflows mature, UX should be reviewed and refined.

Strong UX governance includes:

- Monitoring usage patterns and adoption metrics
- Gathering structured user feedback
- Periodically reviewing views, dashboards, and layouts
- Adjusting designs based on real-world behavior

UX maturity increases over time when refinement is intentional rather than reactive.

Common User Experience Pitfalls

Even experienced teams encounter UX challenges when focus shifts too heavily toward configuration or automation.

Designing for Power Users Instead of Everyday Users

Focusing UX on administrators or power users often overwhelms the majority of contributors, making it harder for them to complete common tasks efficiently.

How to Avoid It:

Build layouts, views, and dashboards around everyday tasks and the most frequent user actions.

Exposing Too Much Information by Default

Displaying all fields, tables, or views at once creates cognitive overload and increases the risk of mistakes.

How to Avoid It:

Curate content by role and provide only the information users need to act effectively.

Missing Role-Based Work Queues

Without clear “what do I do next” views, users struggle to prioritize and sequence work, which can slow progress and reduce accountability.

How to Avoid It:

Implement role-specific queues that surface pending tasks and highlight priority items.



Inconsistent UX Standards

When views, forms, and dashboards vary across tables or teams, users must relearn how to interact with the system, reducing efficiency and confidence.

How to Avoid It:

Apply consistent formatting, navigation, and interaction patterns across all parts of the solution.

Overloaded Dashboards

Dashboards that try to show everything overwhelm users and dilute focus on critical information.

How to Avoid It:

Design dashboards to highlight key metrics and actions relevant to each role, using filters and drill-downs for additional detail.

Ignoring Conditional Display

Failing to hide irrelevant fields or sections can clutter interfaces, confuse users, and increase errors.

How to Avoid It:

Use progressive disclosure and conditional logic to show only relevant information based on role, status, or context.

Treating UX as a One-Time Activity

Designing UX once and never revisiting it leads to outdated interfaces as processes, roles, and priorities change.

How to Avoid It:

Continuously gather user feedback and iterate UX layouts and dashboards to reflect evolving workflows and business needs.

High-Level User Experience Checklist

The checklist below highlights the essential UX considerations to address before moving into modeling and configuration. These items can be documented and tracked directly in SmartSuite to ensure visibility, consistency, and a user-centered approach throughout the design process.

Checklist Item	Description	Benefits
Role-Based Layouts Defined	Record layouts tailored to user roles and responsibilities	Improves clarity and task focus
Conditional Display Applied	Fields and sections shown only when relevant	Reduces cognitive load
Views Mapped to Work Modes	Each view supports a specific action or question	Prevents clutter and confusion
Dashboards Built by Role	Dashboards drive decisions and next steps	Increases adoption and accountability
UX Standards Established	Naming, layouts, and patterns are consistent	Builds trust and scalability
Feedback Loop Defined	UX reviewed and refined post-launch	Ensures long-term usability



User Experience Use Case Example: Issue & Action Tracking

In an Issue & Action Tracking solution, UX was intentionally designed around three primary user roles: contributors, owners, and executives.

Contributors interacted with simplified record layouts and queue-based views showing only assigned actions and required fields.

Owners used dashboards highlighting overdue items, risk severity, and validation status, with direct links to records requiring attention.

Executives accessed high-level dashboards focused on trends, aging, and remediation effectiveness without exposure to operational detail.

Conditional display rules ensured that root cause analysis, evidence uploads, and validation steps appeared only when relevant. Standardized layouts and status definitions maintained consistency across issues and actions.

The result was high adoption, improved data completeness, and increased leadership confidence in reporting, all driven by intentional UX design.



Conclusion: Designing Experiences that Work

The User Experience phase ensures that SmartSuite solutions are not just functional, but intuitive, actionable, and aligned with how people actually work. Thoughtful UX design reduces friction, guides users through responsibilities, and reinforces accountability, transforming the system from a static repository into an operational tool that actively supports execution and decision-making. Consistent interfaces, clear navigation, and prioritized dashboards help users quickly find information, focus on critical tasks, and engage confidently with the system, improving adoption and trust in the data and processes.

In addition, strong UX establishes a foundation for ongoing improvement. As processes, roles, and data requirements evolve, maintaining and refining interfaces ensures the solution remains efficient, relevant, and scalable. Prioritizing UX from the start allows teams to deliver a system that grows with the organization while keeping users productive, minimizing errors, and maximizing confidence in the solution.

