

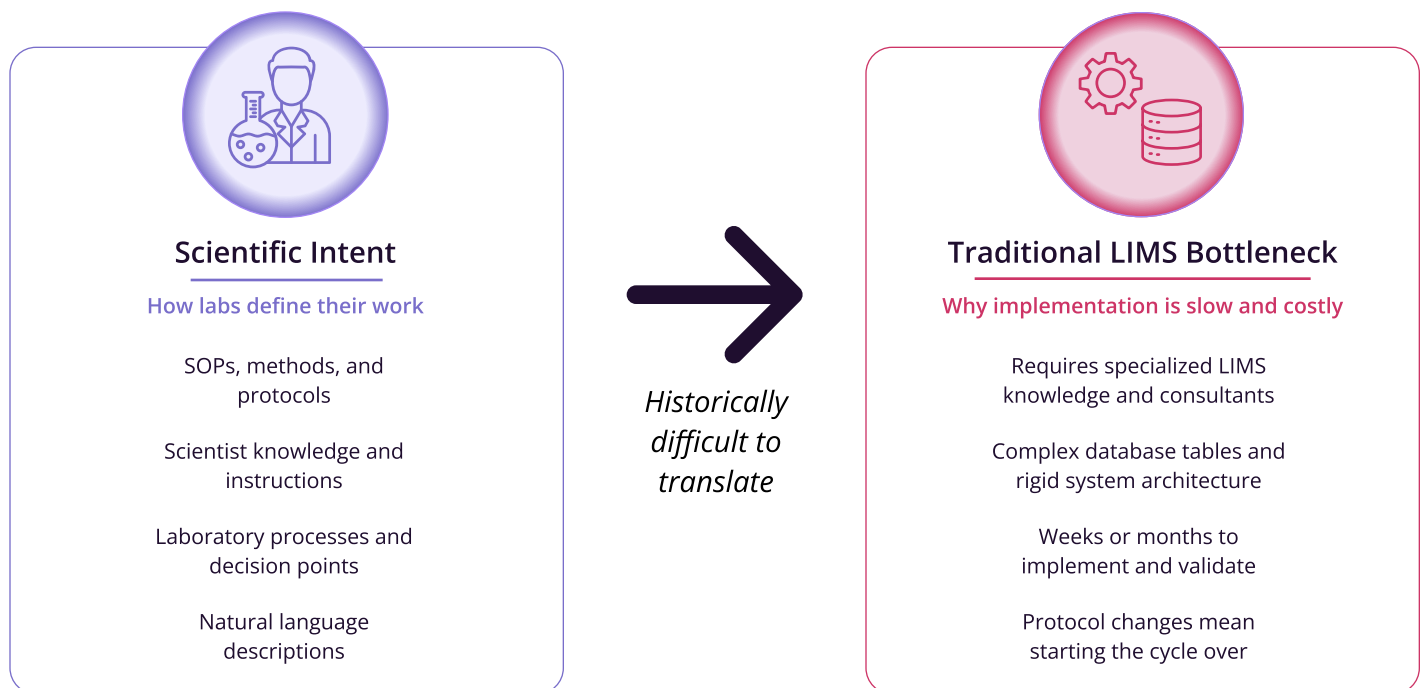


# AI-Powered LIMS Workflows: The Making of Labbit's Configuration Assistant

*How Labbit uses AI agents to turn natural language into deployable LIMS workflows*

Configuring a LIMS can be a complex undertaking. Translating the specific operational needs of a laboratory into a valid workflow has historically required specialized LIMS knowledge, significant time, and deep software expertise. Not to mention the understanding of how to map scientific processes into a rigid system architecture.

For many labs, this complexity creates friction that can add weeks or even months between finalizing a protocol and getting it up and running in the LIMS. When the protocol inevitably changes, the cycle starts again.



Large language models (LLMs) offer a potential solution to this problem. LLMs excel at interpreting natural language, generating structured outputs with clear guardrails and inputs, and executing tasks in a fraction of the time a human would take. If it were possible to train an LLM on LIMS configuration, maybe it would be possible to implement a LIMS with only natural language instructions from a human.

But leveraging LLMs for LIMS implementation is not as simple as uploading a workflow to ChatGPT and pressing send. First off, traditional LIMS configuration is defined in complex database tables, which are not conducive to LLM training. Additionally, LLM output is non-deterministic: prompt the same model the same way ten times and you may get ten different responses. In LIMS configuration, precision is non-negotiable. Because the LIMS is responsible for the integrity and accuracy of laboratory results, outputs cannot be left up to chance. This was the exact challenge the Labbit engineering team set out to solve: taking advantage of LLMs without sacrificing quality and accuracy that LIMS require.

# Why Labbit's Architecture Works Well with LLMs

At its core, Labbit is a modern, cloud-based LIMS built to support the full spectrum of compliant laboratory operations, from sample and inventory management to workflow execution and data capture. Harnessing AI to accelerate LIMS configuration and enable faster workflow iteration, the Labbit team built Labbit's Configuration Assistant, an AI-powered tool that translates natural language into working Labbit configuration.

Even before AI entered mainstream software development, Labbit's foundational design and architecture was well suited for integrating LLMs. Rather than storing configuration in proprietary, complex database tables, Labbit uses a well structured collection of human- and machine-readable documents to define overall workflows and individual workflow steps.

This approach maps directly to how LLMs perform best. Each specific component of the workflow is produced independently, verified individually and then assembled into a complete, deployable workflow.

Together, these architectural choices provide the foundation for how the Configuration Assistant works in Labbit today.

## How the Configuration Assistant Works

The Configuration Assistant is an integrated capability within the Labbit LIMS platform that helps labs rapidly design, configure, and evolve their systems. By combining Labbit's native BPMN workflow modeling engine with an AI-powered assistant, it turns protocols, requirements, files, images, and existing process diagrams into visual workflows and ready-to-use configurations.

What makes the assistant powerful is that much of the underlying complexity is essentially invisible to users. Instead of getting lost in technical details, they simply describe their workflow in plain language and receive a working configuration in minutes. In doing so, the assistant shortens the path from protocol to production and makes workflow design more accessible to scientific teams.

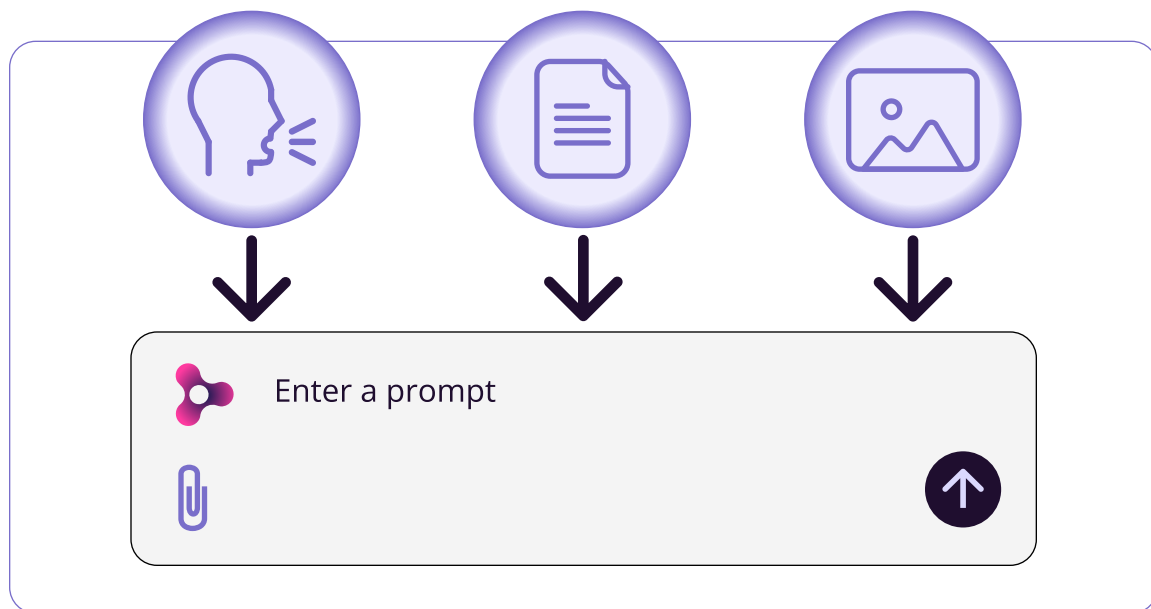
The Configuration Assistant runs on an agentic framework, deploying a collection of specialized agents that each own a defined piece of the workflow and work in concert to execute the prompt. The agentic system, orchestration, and validation processes are unique to Labbit, enabling the Labbit engineering team to ensure that Configuration Assistant will continue to operate optimally as model capabilities evolve.

The following section walks through each component of this process: how users input a prompt, how agents get the context they need to build accurately, how agents divide and execute the work, and where human review steps in to validate the outputs.

## Step 1: Prompt Inputs

Users begin by describing what they want to build. The assistant accepts three input types: natural language prompts, protocol documents, or workflow diagrams:

- **Natural language prompt:** The user describes the workflow in plain text. The system interprets this description, asks clarifying questions if needed, and structures the prompt into a workflow design.
- **Document:** The user uploads an existing standard operating procedure in a PDF. The system parses the document, and uses that content as the basis for workflow design.
- **Workflow diagram:** The user uploads an image, such as a Visio drawing or whiteboard photo. The system uses image recognition capabilities to extract the workflow structure from the visual representation: tasks, their relationships, routing logic, and sequence.



## Step 2: Context

To generate Labbit-specific configuration accurately, the AI system and its agents require context. The goal is more than just generating technically valid files; the configuration must also fit the specifications of Labbit's system architecture.

While the required context varies by agent, they generally need to understand Labbit's architecture, fundamental data types, and configuration conventions. The entirety of Labbit's documentation is available to the system; however, individual agents only receive the subset of information needed to complete their job, helping ensure focused, high-quality outputs.

Since accuracy is essential in regulated lab environments, the Configuration Assistant makes use of retrieval augmented generation (RAG) to ensure that any content (entities, data, etc.) pulled from the Labbit environment is anchored to real system data. RAG also keeps agents anchored to the Labbit environment, constraining their outputs to only that which will work within Labbit system architecture.

The Configuration Assistant has additional built-in mechanisms to validate outputs, which are discussed later on in Step 4: Human and Machine Validation.

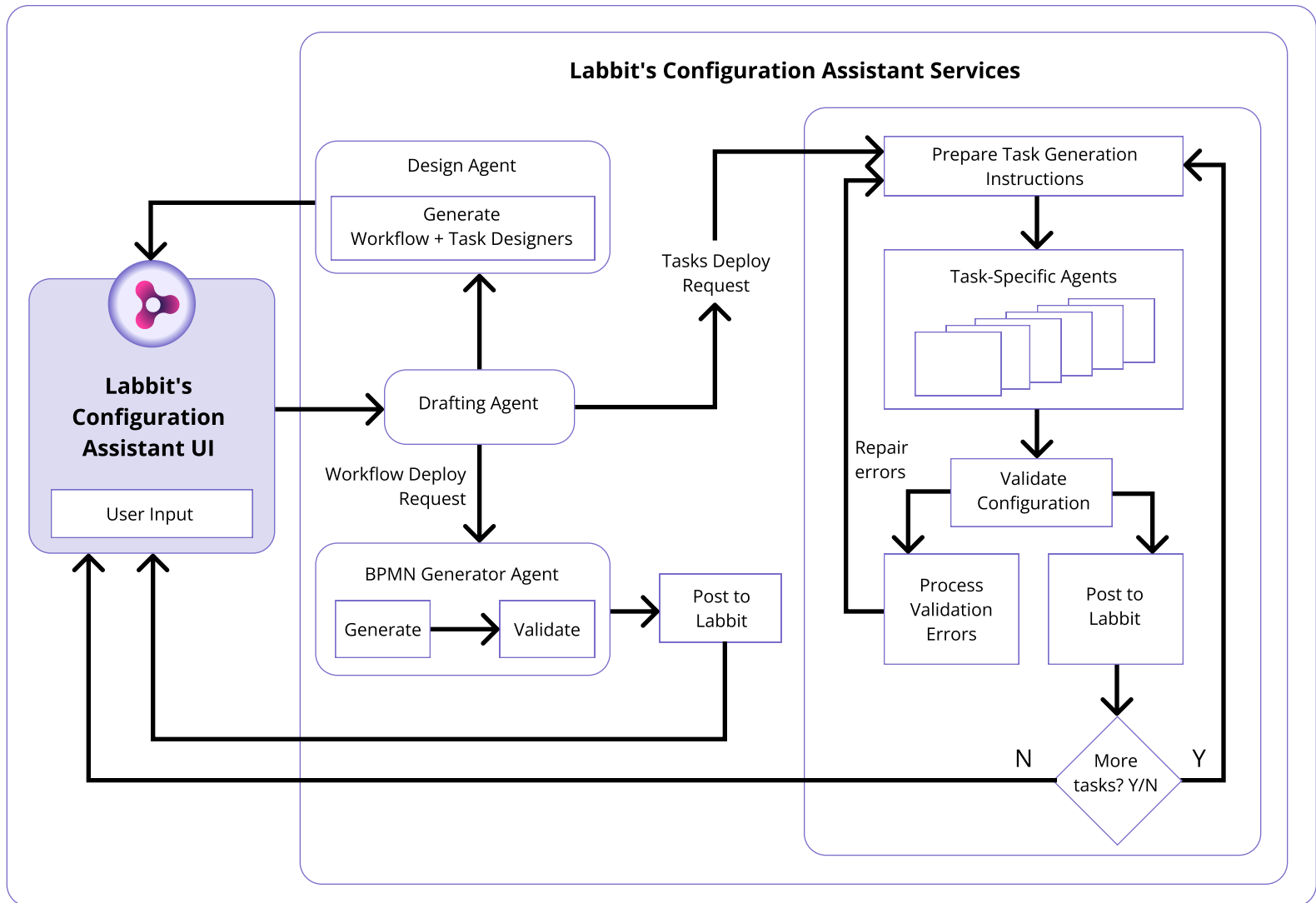
### Step 3: Agent Orchestration, The Divide-and-Conquer Engine

The heart of the Configuration Assistant is a system of specialized agents working together to execute tasks in a seamless, orchestrated way. These agents follow the principle of prompt-chaining: the output of one agent becomes the input for the next. This is a deliberate design choice. Dividing responsibility across focused agents enables LLMs to tackle large, complex implementations. Importantly, before the system generates any configuration documents, it first produces a human-readable design containing workflows, tasks, sequences, and key relationships so users can validate this spec in plain language before any configuration work begins.

Custom-built by Labbit engineers, the framework consists of numerous specialized agents that each handle a narrow piece of workflow design and configuration. These agents include:

- **Design Agent:** This agent works with users to clarify requirements, asks follow-up questions as needed, and produces a structured, human-readable workflow design: a layout of tasks and how entities are routed from task to task in a workflow. It outputs text, not code, so non-technical users can easily understand the Configuration Assistant's recommendations. The design agent has a deep understanding of Labbit system architecture to ensure that any generated protocol will function in Labbit. Before the Configuration Assistant builds anything, the user reviews and approves the Design Agent's output.
- **Drafting Agent:** Once the layout is approved, the Drafting Agent takes over. It reads the Designer Agent's output and converts it into explicit build instructions for all downstream agents. This includes specifying workflows, the order they must be deployed, and the tasks each workflow contains.

- **Task-specific Agents:** There are multiple sections in a Labbit configuration file and each is handled by a dedicated agent with targeted instructions scoped only to that section. This is the divide-and-conquer principle in practice: smaller, more focused agents produce more reliable outputs than a single agent attempting to generate the entire configuration at once. And when errors arise in a specific part of the file, the agent can quickly fix the mistake instead of regenerating the entire file.
- **BPMN Generator.** This agent produces the XML process definition for each workflow, including the tasks, their sequence, and how they connect.



The agentic framework does more than break complex work into coordinated tasks. It also allows the Configuration Assistant to evolve in step with the platform itself. As the Labbit platform's capabilities are continuously expanded, the assistant incorporates these changes by updating or extending the relevant agent behavior within the workflow. This modularity is a significant operational advantage, allowing Labbit engineers to quickly and continuously upgrade the Configuration Assistant.

## Step 4: Human and Machine Validation

The Configuration Assistant employs multiple methods to ensure that outputs align with both Labbit technical requirements and user expectations.

### ***Machine validation: the self-repair loop***

After the Configuration Assistant generates a task configuration file, the output is submitted to Labbit's validation endpoint. If it passes, it moves forward. If it fails, the validation errors are sent to the responsible agent, which fixes the errors and resubmits. Users only ever see configuration that has passed validation; the self-repair process handles issues before they surface.

### ***Human review checkpoint #1: Workflow review and confirmation***

Before the Configuration Assistant generates a workflow configuration, the system displays the proposed workflow layout to the user, who can review it, request changes, or provide more context. Getting the layout right first helps the assistant avoid generating detailed configuration for a workflow that needs structural changes.

### ***Human review checkpoint #2: Detailed task review***

After the Configuration Assistant generates task configuration for each workflow, the system shares the results to the user. The user reviews the task configuration in Labbit, including the data fields and the user interface components. They can edit tasks manually, provide feedback in natural language for the assistant to incorporate, or deploy the configuration as designed. This step is essential for aligning Labbit configuration to the user's specific use case. While Labbit agents can validate that the workflow fits within the structural confines of the Labbit environment, human review is essential for confirming that the workflow is accurate from a scientific and operational perspective. This dynamic showcases the importance of incorporating AI and human-review checkpoints, and keeping users focused on reviewing only areas requiring their expertise.

# A Note on Security and Privacy

The Labbit team built the Configuration Assistant with the same security and privacy standards that govern Labbit overall.

- **Who can access it?**

**Access and deployment controls.** At its core, the Configuration Assistant operates with clear human review checkpoints, as outlined above. Before anything is pushed to production, users always receive a draft configuration that they must review and approve. This is the same process that governs all Labbit configuration changes. Role-based access control adds a further gate, controlling who can view or approve within any given environment.

- **How can I test configuration before it's deployed?**

**Safe testing before production.** Labbit's environment feature gives teams a designated space to evaluate AI-generated configuration before they reach production. Because each environment has its own configuration, permissions and data, teams can separate development, testing, and production work, validate changes in a test environment, and move them forward when ready.

- **How is my data governed?**

**Versioning and traceability.** Every configuration change, AI-generated or otherwise, is versioned and fully traceable, with a complete auditable history of what changed, when, and who deployed it. In regulated environments where configuration history is subject to inspection, that audibility is required and available within Labbit, regardless of how it was created.

- **Is my data secure?**

**Data security.** All data in Labbit is encrypted, stored securely, and accessible only to authorized users. Labbit is SOC 2 Type II certified.

- **Does Labbit comply with industry regulations?**

**Regulatory compliance.** Labbit is built for the compliance requirements of regulated laboratory environments (GxP, CAP/CLIA, FDA, and ISO). Compliance controls are embedded directly into workflows, including 21 CFR Part 11-aligned eSignatures and automated audit trails that capture every action and result. AI-generated configuration operates within these same controls.

# The User Experience

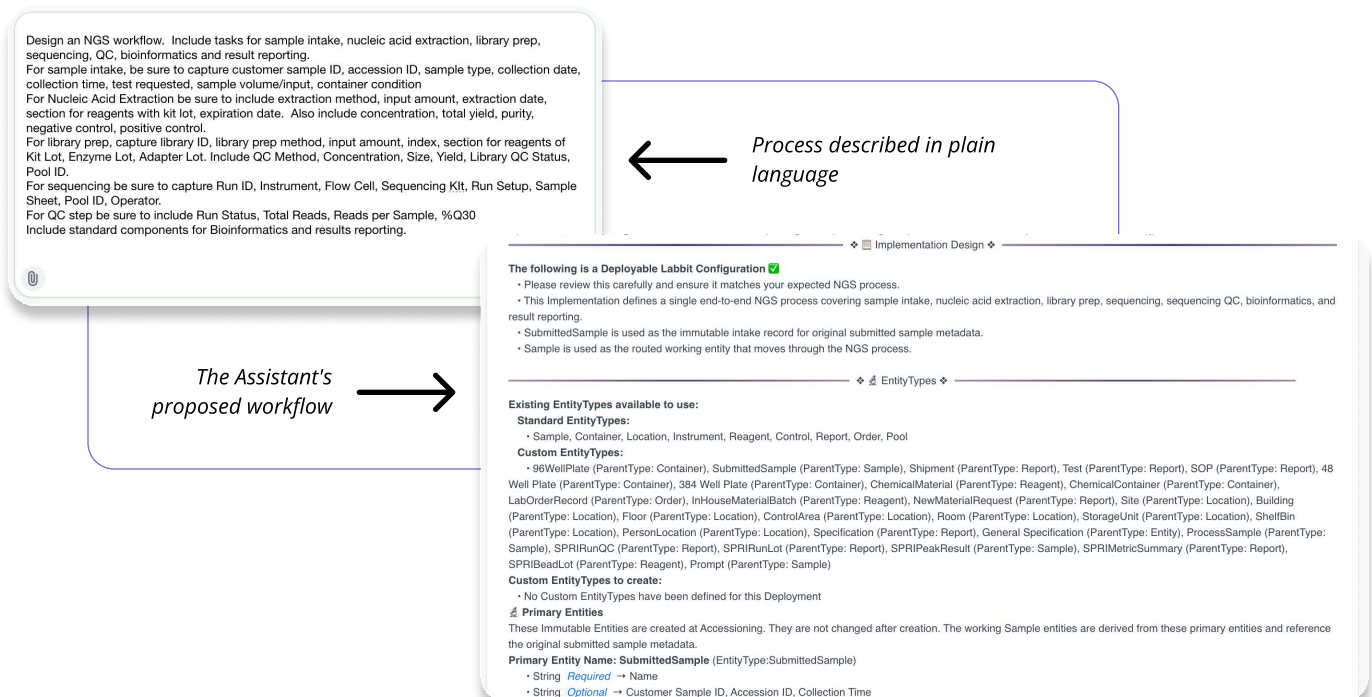
The architecture described in the preceding section includes a lot of moving parts: specialized agents, retrieval, prompt chaining, schema validation, security controls. But the power of the Configuration Assistant comes from the fact that all these complex details are essentially “invisible” to the users. What users see is a chat interface that, within minutes, ingests their prompt and returns a high-quality, working configuration.

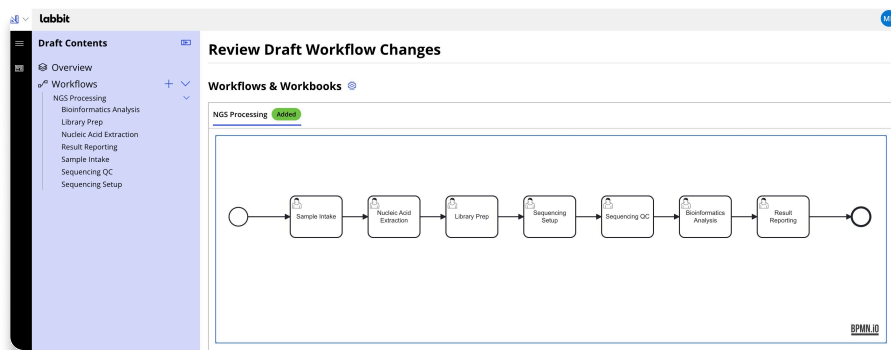
That gap—between a complex system under the hood and a simple experience at the surface—is the core design achievement of the Configuration Assistant. Users describe what they want in natural language. The system seamlessly translates these instructions into the structured configuration Labbit needs to execute the workflow.

## Example: standing up an NGS workflow

Here’s an example in practice. Let’s say a scientist wants to stand up an NGS workflow. They describe the workflow in plain language: sample intake nucleic acid extraction, library prep, sequencing, QC, and result reporting. Then, they specify the relevant scientific details, such as sample metadata, QC thresholds, and which results need to be captured at each step.

Within minutes, the assistant returns a proposed workflow layout: the tasks, the sequence, and the key routing logic. The scientist reviews the layout to confirm it matches how the protocol actually runs in their lab.





← The proposed workflow opened in Labbit's Designer for review and further iteration.

## Common Use Cases

The Configuration Assistant's most fundamental value is closing the gap between scientific knowledge and platform expertise. Scientists and lab teams often know exactly how a protocol should run, what data needs to be captured, and where decision points belong. The Configuration Assistant enables the laboratory expert to compress the build, test, debut cycle so that any user, regardless of technical background, can create a lab workflow.

In practice, the Configuration Assistant excels at supporting the following use cases:

- **New workflow creation.** A LIMS user uploads a protocol document, describes a process, or shares a workflow diagram. The Assistant proposes a workflow layout, the user reviews and approves it, and task configuration is generated. In a single brief session, a user with minimal training has configured the LIMS to their specification.
- **Duplicate and modify.** When a team needs to adapt an existing workflow for a new process or variant, the assistant handles the heavy lifting. For example, let's say a team wants to modify a DNA extraction workflow for an RNA variant. The assistant pulls the existing configuration, applies the requested modifications, and generates the adapted version, preserving what should be preserved and changing what needs to change, without rebuilding the common structure from scratch.
- **Iterate on existing workflows and tasks.** A workflow is live and needs a new task, a modified data field, or a UI change. The user describes the change in natural language; the assistant generates the update as a draft changeset, which the user tests in a sandbox and deploys through the standard review process.

## Early Results

The Configuration Assistant is currently in active pilot with Labbit's services team, who are using it to build and modify workflows. Experience it firsthand through Labbit's interactive online demo (link below) or request a trial to see it in action.

Even before the Configuration Assistant was developed, Labbit was reducing configuration timelines relative to legacy LIMS. The Configuration Assistant builds on this foundation by helping teams generate a strong starting point even faster. Teams can get a working configuration in minutes, then validate, refine, and iterate from there. Teams can explore multiple workflow structures, compare approaches, and iterate more frequently, all without configuration headaches.

## What's Next

The translation gap between scientific knowledge and system configuration has constrained lab teams for as long as LIMS platforms have existed. But advancements in LLMs and AI are now closing this divide, transforming days of work into minutes

And this is just the beginning. The Labbit engineering team is continuing to build new functionality into the Configuration Assistant, including more advanced system validation and incremental workflow updates.

Try the Configuration Assistant for yourself: **[www.labbit.com/ai-configuration-assistant](http://www.labbit.com/ai-configuration-assistant)**.

Interested in learning more about Labbit? Visit **[www.labbit.com](http://www.labbit.com)**.

**labbbit**

**Contact Us**

[sales@labbbit.com](mailto:sales@labbbit.com)

[www.labbbit.com](http://www.labbbit.com)