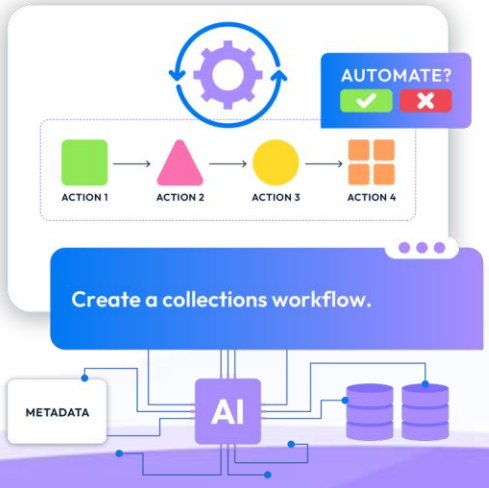
BillingPlatform

FEATURE SHEET

# Workflow Automation: Build & Explain

Run your billing platform by describing what you want.




**Describe the automation to create it. Ask about any automation to understand it. Both in plain language.**

## The Problem

Billing logic lives in workflow rules — the triggers, conditions, and actions that decide when an invoice is generated, when an account is flagged, when a system is called. Over time, building a new rule requires a developer or a ticket, the rules already running become a black box nobody fully understands, and the reasoning behind them leaves with the person who built them. Audits stall on “what does this do?” Handoffs lose context. Every change carries risk because no one is certain what they are changing. BillingPlatform AI closes both ends of that gap.

## Build — automation from a description

Describe the outcome in plain language — for example, “when an invoice ages past 60 days, email the AR owner and flag the account for review.” BillingPlatform AI configures the complete rule directly in the platform: the trigger event, the conditions, the actions, and the formula logic. Because the AI works against a semantic model of your revenue lifecycle rather than a fixed set of pre-built APIs, it builds against the workflow engine’s real constraints and field relationships, not an approximation of them. What used to take a sprint becomes a working session.

## Explain — plain language from any automation

Point at any existing workflow rule — one you built, one you inherited, one nobody remembers — and BillingPlatform AI describes, in clear language, how it is configured:

- When it fires — the trigger event, schedule, or condition
- How it connects to other systems — synchronous or asynchronous callouts
- What it checks before it acts
- Its configured failure handling, including rollback settings
- What it changes, creates, or sends

An opaque expression becomes a sentence anyone on the team can read — turning audit questions, handoffs, and “why is this number what it is?” into a conversation instead of a forensic exercise.

## Trust & governance

- **Permission-faithful** — every action runs as a real user under your existing role-based permissions, and every change is captured in the native audit trail — attributable and reversible. No elevated AI account, no separate access model to govern.
- **Validated before save** — generated workflows pass the same validation as hand-built ones. What will not run cannot be persisted.
- **Governed your way** — layer custom validations and approval workflows on top wherever your process requires explicit sign-off. You decide how much governance each change carries.

## Specifications

### Trigger events
Insert, Update, Delete, Scheduled (interval, run time, time zone)

### Action types
Field Update, Email, HTTP Callout (synchronous and asynchronous), Document Generation, Record Create / Update / Upsert

### Failure handling
Per-rule rollback; asynchronous callout retry with exponential backoff

### Foundation
Built on the Model Context Protocol — open standard, no model lock-in, no parallel data plane

### Tenant-aware
Custom entities, fields, and relationships discovered via native metadata — no per-tenant integration build

## Who it’s for

Finance and RevOps teams who own billing logic but should not need engineering for every change. Controllers and auditors who need to know what automation is doing without a discovery project. Implementation and operations teams inheriting a configured platform they did not build.

See **it on your own workflows. Talk to your BillingPlatform account team to request a demo — or point your AI assistant at your sandbox tenant and try it yourself.**