

Delegate: Eliminating administrative busywork in software testing



The biggest time thief in testing **isn't testing**

Imagine you are analyzing your daily regression test run. The run finishes, but the real work is just starting.

First, you check the build notification, confirm that the build completed, validate the deployment, open the linked test run, verify the test run context, and save the useful links and IDs. Then you move into the failed automated tests: opening failure details, reviewing logs, comparing failures with previous runs, identifying new versus repeated failures, and classifying the likely root cause.

Some failures point to real product defects. Those need to be reproduced, assessed for severity, documented with evidence, and linked to Jira. Other failures come from environment or automation issues, so you check environment health, review data availability, examine script or framework behavior, rerun tests, record likely root causes, and escalate blockers when needed.

Finally, you prepare the execution summary: delivery risk, failed tests, rerun outcomes, tested requirements, product defects, environment issues, automation issues, follow-up actions, and updates for the team, QA lead, project manager, or release manager.

And this is just one example. Tester time is consumed by many surrounding activities that keep delivery moving but are not hands-on testing. In fact, only about 30% of software tester work is spent on hands-on testing. The remaining 70% goes into administrative and operational work such as analysis, reporting, meetings, follow-ups, maintenance, and coordination.

This is how one recurring testing workflow can consume 4–8 hours per week, span 6–8 applications, and involve 6–7 roles across testing, development, management, environment, and infrastructure teams. Delegate is your ally for ditching testing busywork. It turns repetitive work into agent-managed action, giving testers more time to focus on finding real product issues.

What Delegate **can do**

Delegate is your testing team's digital sidekick to ditch busywork. It learns from users, executes tasks across apps, and helps teams move beyond suggestions to completed work. Built for the enterprise, Delegate combines computer use, automation, skills, governance, and security controls so testers can hand off more of the busywork around testing.

Cross-application task execution — Delegate drives your desktop and web apps, reads files, calls APIs, and moves data between systems without manual copy-paste. It works across Windows and macOS, handling UI and API actions in a single workflow.

Human-in-the-loop execution modes — Choose how much control you keep. Guided mode asks before risky actions. Autonomous mode proceeds unless blocked. Background mode runs without stealing your focus, notifying you when done.

Reusable routines and scheduling — Save any successful session as a Routine that runs on demand, on a trigger, or on a schedule. Teach Delegate once, then let it handle the repetition.

What Delegate **can do** (continued)

Delegate by text, voice, or demonstration — Text prompts, voice commands, screen recordings, and templates give you flexible ways to delegate work.

Enterprise security and governance — Control which tools Delegate can use, set permissions, require human confirmation for sensitive actions, and maintain full audit trails. Your IT team governs access centrally through the UiPath Platform.

Pre-built connectors — Connect to Confluence, GitHub, Gmail, Jira, Slack, Salesforce, and hundreds more business systems through UiPath Integration Service — using your existing credentials and permissions.

MCP server connectivity — Extend reach by connecting to external MCP (Model Context Protocol) servers, adding new capabilities and integrations as your toolchain evolves.

Pre-built skills library — Access hundreds of ready-made skills for common workflows: file system operations, Git version control, PowerPoint creation, Word document editing, PDF processing, Excel analysis, and more.

Custom skill building — Create your own skills to teach Delegate how your team works — build them with Delegate's help or independently, then share them across your organization.

Persistent learning and session memory — Delegate learns from your conversations over time, remembers your preferences.



Testing use cases **for Delegate**

- **Build and test run context review:** Checking notifications, confirming build and deployment status, opening linked test runs, verifying test run context, saving useful links and IDs. Delegate handles the full setup before a tester opens a single result.
- **Failed test triage:** Opening failure details, reviewing logs, comparing failures with previous runs, classifying failures, capturing triage outcomes, documenting results. Delegate works the queue so nothing falls through.
- **Defect follow-up:** Checking for existing Jira issues, updating open defects, creating new ones, adding evidence, linking test results to Jira issues. Delegate closes the loop on confirmed defects without manual handoff.
- **Environment and automation issue handling:** Checking environment health, reviewing data availability, examining script or framework behavior, re-running failed tests, recording root causes, creating follow-up tasks, escalating blockers. Delegate keeps triage moving when the noise spikes.

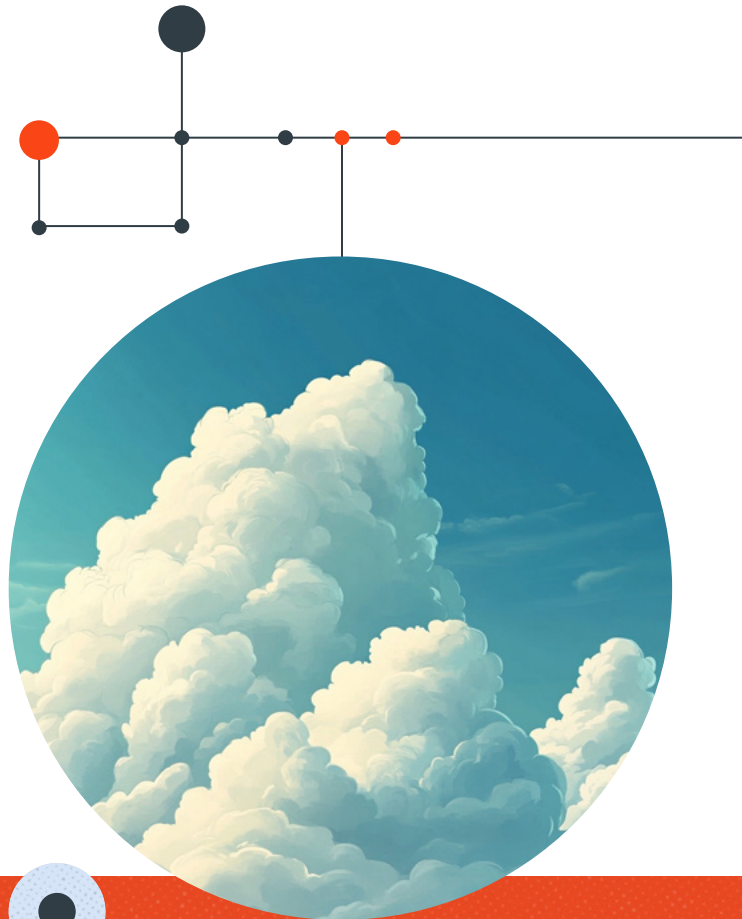
Testing use cases for Delegate (continued)

- **Execution summary preparation:** Delivery risk, failed tests, re-run outcomes, tested requirements, product defects, environment issues, automation issues, follow-up actions. Delegate prepares and shares the full execution summary, ready to send.
- **Additional testing busywork:** Test case maintenance, requirement analysis, test data administration, environment coordination, defect lifecycle administration, reporting and meeting overhead — if it's repetitive, Delegate can take it.

How Delegate compares to other agents

Many AI agents can help with tasks, but Delegate is built for enterprise-grade action across the real tools testing teams use every day. It runs natively on Windows and macOS, works across desktop apps, browsers, files, legacy systems, SAP GUI, Citrix, and other enterprise tools, and connects to 100+ enterprise systems through UiPath Integration Service.

Delegate also combines agentic reasoning with reusable automations, skills, and routines. That means it can help with repeatable testing workflows like failed-test triage, defect follow-up, execution reporting, and cross-team coordination, while still supporting enterprise controls like model flexibility, private-cloud deployment, audit logs, RBAC, data residency, and security governance.



Where Delegate goes further

Think of Autopilot as embedded assistance and Delegate as cross-tool action. Autopilot helps inside UiPath experiences like Test Manager and Studio. Delegate goes further by acting across tools, technologies, and platforms on the user's behalf.

That is what makes Delegate different for software testing teams. It can learn repetitive work like failed-result triage, execution summaries, Jira follow-up, and coordination steps, then turn trusted skills into routines that run on a schedule or trigger.

The biggest time thief in testing isn't testing. It is the repetitive work around testing. Delegate gives testers a way to hand that work to an agent, reclaim time, and focus on finding the issues that matter.

Ready to learn more?

UiPath