

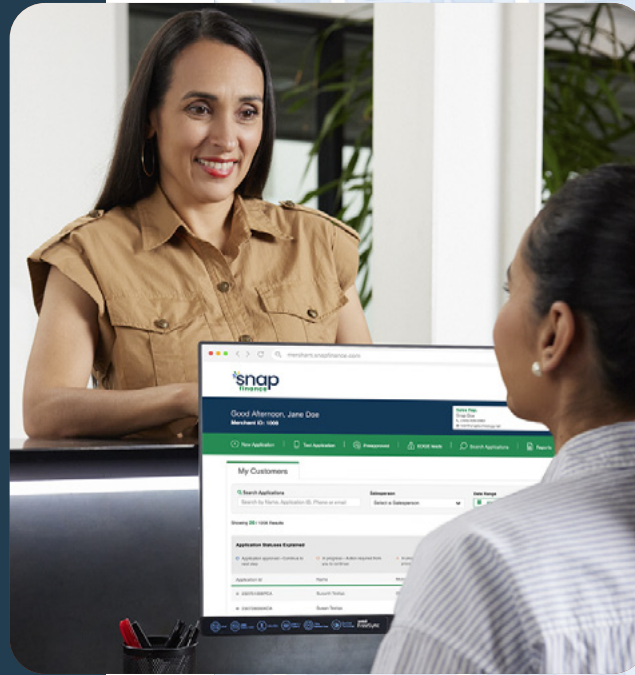


CUSTOMER SPOTLIGHT

Building robust, modular dashboards with internal packages and Posit

SUMMARY :

Snap uses Posit's Git-backed publishing feature to seamlessly modify dashboards across our team and publish to Posit Connect. Snap has two Posit Connect servers: production and development, which allows testing features or coding methods without messing with what stakeholders are seeing. Posit Package Manager helps share internally created packages (like {graveler}) and ensure that everyone uses the same packages/functions for work.



ABOUT :

Snap Finance is a digital financial services company that offers lease-to-own and installment loan solutions, specializing in consumer financing for those with bad or no credit.

INDUSTRY :

Finance

SIZE :

1500+ employees

TECHNOLOGY USED :

Posit Workbench, Posit Package Manger, Posit Connect

The Challenge:

The Maintenance “Labyrinth”

When every developer has a different “dialect” or coding style, legacy dashboards become impossible to update.

- **UNSUSTAINABLE MAINTENANCE BURDEN:** With no shared structure or standards, updating a legacy dashboard meant hours spent reverse-engineering another developer’s logic before a single line of new code could be written, pulling the team away from building new features and toward firefighting existing ones.
- **COMPOUNDING TECHNICAL DEBT:** Inconsistent code made refactoring so time-consuming that developers routinely chose to scrap and rebuild dashboards from scratch. This reset the cycle of fragmentation with every new project.



The Solution:

A standardized, reproducible development workflow

- **THE FOUNDATION: {GOLEM}**
The team adopted the open-source package {golem}, which forces Shiny dashboards to be built like R packages. This ensures every project has a consistent structure, built-in documentation, and rigorous testing.

- **GOVERNED DEVELOPMENT STANDARDS:**
The {golem} framework enforced a consistent, package-based structure for every Shiny dashboard, while the internal {graveler} package (shared via Posit Package Manager) ensured the entire team used the same functions and dependency versions across all projects.
- **GIT-BACKED PRODUCTION PIPELINE:**
GIT-BACKED PRODUCTION PIPELINE: Posit Connect’s Git-backed publishing created a seamless, auditable path from a developer’s machine to the production environment, with a separate development server allowing new features to be tested safely before stakeholders see any changes.

Learn more at posit.co > | [Read the full story](#) >