

Deployment, delivered better

Automating deployments and streamlining build pipelines with automated CI/CD



Summary

Sonata partnered with a leading global provider of cloud-based human capital management (HCM) and workforce management (WFM) software to modernize its software release and build processes. The engagement spanned two complementary workstreams: automating deployments across development, QA, UAT, and production environments, and optimizing the developer build chain to shorten feedback loops. By introducing modular, CI/CD-integrated deployment automation along with build chain enhancements such as tool upgrades and parallel execution, Sonata helped the client achieve significantly faster, more consistent releases while reducing build chain time by close to 35 minutes per cycle, freeing development teams to focus on innovation rather than manual processes.

Customer overview

A leading global technology company that provides cloud-based HCM and WFM software solutions.

Headquarters Massachusetts	Revenue \$5Bn	Industry Technology
--------------------------------------	-------------------------	-------------------------------

Pressure points

The client's software release process spanned multiple environments, each with its own configuration, and relied heavily on manual steps that were time-consuming and prone to error. At the same time, long build chain durations and an extended developer feedback loop were slowing down testing, deployment, and overall development velocity.

Release optimization

Inconsistent, manual deployment steps across development, QA, UAT, and production increased the risk of version mismatches and rollbacks

Environment-specific configurations added maintenance complexity without a standardized approach

Sensitive credentials and access data lacked a secure, centralized management approach

Limited rollback capabilities meant failures could lead to extended downtime

Build optimization

Developer feedback loops took between 14 and 29 hours, even under ideal conditions

The core build chain alone took 6 to 10 hours, delaying testing and deployment

Redundancy across more than 40 build configurations slowed the CI pipeline

Sequential execution of build and test stages created unnecessary idle time

Solution highlights

Sonata implemented a two-pronged modernization approach as below:

Release optimization

End-to-end automated deployments across all environments using modular playbooks for installation, upgrade, and rollback

CI/CD integration with GitHub using version-tag-based deployment triggers for traceable releases

Environment-aware logic for dynamic script execution tailored to each server group

GCP Secret Manager and centralized logging integration for secure operations and audit trails

Parameterized, extensible architecture designed to support future components and environments

Build optimization

Upgraded MSBuild from v14 to v17 to improve build performance

Split and refined backup steps, and merged similar builds to remove redundancy

Enabled parallel execution of code generation with Apply Server Updater, and of MSBuild with NUnit tests

Reused compiled DLLs to skip unnecessary rebuilds and reordered long-running steps earlier in the pipeline

Executed a proof of concept to migrate Pro Build Pipelines to GitHub Actions using GARM runners, successfully generating a single MSI artifact

Results that speak volumes

Release optimization

Significant reduction in deployment time, leading to faster time-to-market

Higher reliability and consistency in installations across all environments

Improved rollback mechanisms, minimizing impact from failed releases

Better visibility into release activities through centralized logging and monitoring

Enhanced security with secrets and access management via GCP

Build optimization

Eliminated redundant build steps and optimized execution order

Improved developer experience via faster feedback loops, shorter cycle times, and greater efficiency

By the numbers

~**24** minutes saved by optimizing the core build chain

~**10** additional minutes saved through backup and apply server updater improvements