

Step-By-Step: How DataArt Processes Application Backlogs, Converting Legacy Apps with DevOps

By [David Geer](#)

DataArt designs and custom builds software for industries such as travel & hospitality, media, IoT, financial services, and healthcare. “We build new products and services, improving existing solutions, re-engineering legacy systems, and providing expertise in technological and industrial domains,” says Anton Krasikov, Project Manager, DataArt.

How DataArt Uses DevOps

DataArt quickly delivers high quality software in support of operations performance. DataArt’s DevOps process enlists version control systems like Git and build systems such as Ant, Maven, and Gradle. “Almost every project relies on using these build systems inside CI servers like Jenkins, TeamCity, or TFS,” says Krasikov.

DataArt’s development process also requires a lot of unit and integration testing using frameworks like TestNG and Junit. “There are also projects where we prefer TDD or BDD approaches, with the use of specific frameworks such as Selenium for UI testing,” says Krasikov. By using Puppet, Vagrant, and Chef, DataArt can swiftly deploy the development environment in the CD pipeline. DataArt applies these select DevOps development tools using methodologies, which could include Kanban or Scrum (Scrum was most frequently mentioned by DataArt for this story).

Previously, DataArt relied on manual processes or scripting tools for build systems, source code, and unit tests. “It took a lot of management effort to tie everything including development, quality assurance, and operations together,” says Krasikov.

Challenges to Adding Backlogged Applications to Those You Develop Using DevOps Approaches

In the instance where DataArt returns to work on existing systems that include some legacy applications, it is possible to use agile development in maintaining these as they are, but excessive software dependencies could make it hard to align the software releases / release cycles.

“Legacy software may not have good test coverage or a proper build system, so we usually take steps in a DevOps direction, if not to achieve continuous delivery, at least to implement proper continuous integration,” says Krasikov.

There are challenges to adding backlogged apps to a DevOps development process. “These kinds of non-DevOps applications usually suffer from several issues, or in worst case scenarios from all of them,” says Krasikov. Backlogged apps waiting in a pre-DevOps state and hungry for transformation may have any or all of the following symptoms, according to Krasikov:

- *many dependencies between system modules
- *no automated environments

- *no continuous integration
- *no test automation
- *no build system or inefficient usage

Transforming an Application, Step-By-Step

To add apps to DevOps approaches, says Krasikov, start by analyzing the existing source code, system dependencies, and system environments. The steps that follow come from an example with a recent Java project that DataArt did.

A) Select a source code control system such as Git.

B) Do refactoring and perform unit tests. “These go hand-in-hand, so the main business logic was covered with tests. It does not make sense to rush to 100% test coverage, but we tested the main bits. This also allowed us to reduce system complexity and decouple system components,” says Krasikov.

C) Build system – “The original application used huge and confusing Ant scripts to produce deployable packages. After we completed the refactoring, we split the application into modules with Maven as the build and dependency management system,” says Krasikov.

D) Integration tests – “We introduced these as a separate test automation suite to ensure that application modules work correctly as a whole,” says Krasikov.

E) Continuous Integration – DataArt set up a Jenkins server for multiple build jobs, building with each commit and targeting deployments to their intended environments with overnight integration testing. “CI allowed us to identify non-working builds early to save time on investigations,” says Krasikov.

F) Environment setup – Finally, DataArt implemented infrastructure as code. The new development environment uses Puppet with Vagrant to check out source code, spin up a VM box / template, and deploy fast, according to Krasikov.

DataArt runs some steps in parallel for additional time savings, for example moving to virtualized provisioned environments along with source code migration, according to Krasikov.

DevOps Helps DataArt Speed Projects to Their Customers’ Delight

By increasingly converting backlogged apps to DevOps development approaches, the enterprise benefits by achieving project goals and delivering a working software iteration with each pass through the DevOps processes. “The high level of automation reduces the risk of human errors and helps us to fail early in the development cycle. With continuous delivery, it is much easier for our clients to see the immediate impact of their ideas,” says Krasikov.

Original article — <http://devops.com/2015/05/06/step-step-dataart-processes-application-backlogs-converting-legacy-apps-devops-approaches/>