

## BACKGROUND

Modern software development teams depend on multiple solutions and workflows to support their evolving needs and technology. Unfortunately, each product that's implemented introduces an additional layer of complexity, risk, and inefficiency—no matter how valuable or necessary the product may be.

Coordinating tasks, syncing information, and transferring work products across a collection of systems and workflows invariably takes time and increases the opportunity for error. At best, teams depend on partial integrations that are highly complex, expensive, and difficult to maintain. Other common symptoms include siloed knowledge, limited collaboration, lower development velocities, and system vulnerabilities.



**The challenge of coordinating systems and workflows is not the same for every team, and it can be uniquely difficult for teams employing approaches like distributed, non-linear software development.**

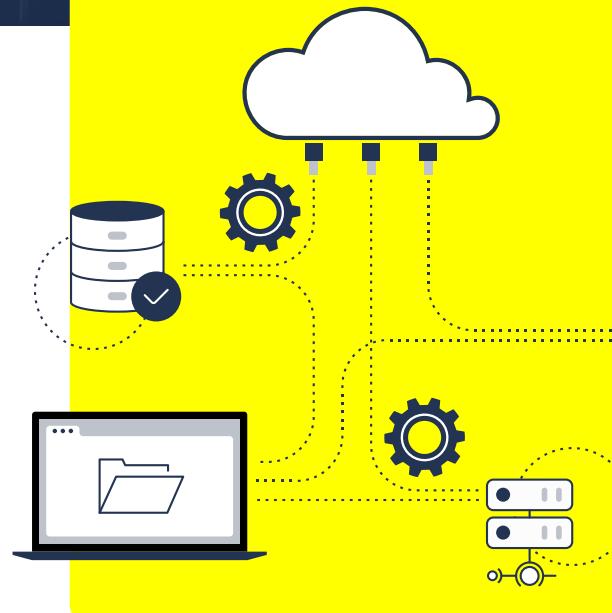
Recognizing these challenging dynamics, the OMNI Arsenal team leveraged its deep expertise in DevOps and advanced Agile software development methodologies to come up with a solution. Arsenal is an all-in-one platform designed to improve agility, collaboration, and communication at every level of development. The platform puts DevOps teams in command of a seamless, end-to-end solution that integrates Software Development, Cyber Security, and Ops systems to help teams stay in sync and deliver Better. Software. Faster.

However, the challenge of coordinating systems and workflows is not the same for every team, and it can be uniquely difficult for teams employing approaches like distributed, non-linear software development. In these cases, work is performed simultaneously on different elements of a project and at different stages of the software development life cycle (SDLC). Timelines for the completion and order of work are estimated but often incorrect, requiring frequent adjustments and coordination. While there are many advantages to this kind of development, there are also significant areas for improvement.

For example, an engineer uses one particular application to develop code and has teammates using other applications to work on quality assurance (QA) and deployment at the same time. To see the progress and details of each other's work, they all have to reference a separate project management platform where information has to be manually entered and updated. In actuality, teams with this approach often use separate applications for the majority of project management, code development, code integration, testing, QA, and deployment. Spread across an entire team and shifting development tasks, the opportunities for error and delay are manifold.

## Use Case

**OMNI CodeOps** streamlines management of distributed, non-linear software development.



Arsenal is an all-in-one platform designed to improve agility, collaboration, and communication at every level of development.

The platform puts DevOps teams in command of a seamless, end-to-end solution that integrates Software Development, Cyber Security, and Ops systems to help teams stay in sync and deliver **Better. Software. Faster.**



## Use Case

**OMNI CodeOps** streamlines management of distributed, non-linear software development.

### BACKGROUND CONTINUED

Development teams are not the only ones affected by deployment pipeline issues like these. Information and insights from these same systems and workflows regularly inform other non-technical decision-makers. Executives like a company's COO will use this data in determining burn rates, ROI, resource allocation, hiring needs, process improvements, and even product release dates. The freshness, accuracy, and availability of this information directly impacts how quickly and how well such decisions are made for a business.

### SOLUTION

Fortunately, Arsenal's core platform is complemented by CodeOps, an out-of-the-box solution that includes all of the necessary tools to successfully Code. Build. Test. Deploy. CodeOps streamlines teams' management of distributed, non-linear software development and significantly improves deployment pipelines through automation and collaboration tools.

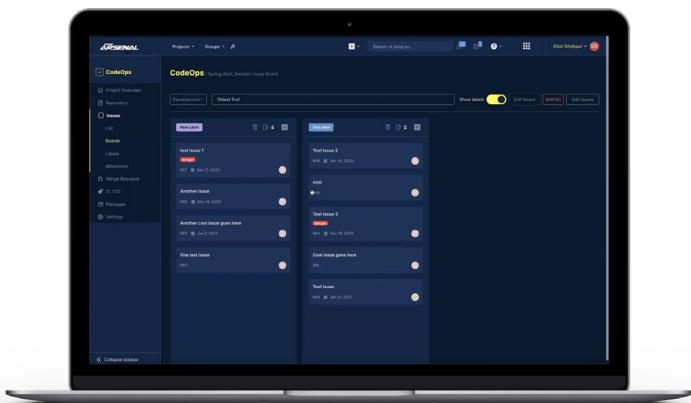
Automated pipelines incorporate testing and validation every step of the way before merging code changes into the main branch. Automated code reviews further improve quality by assessing development projects against industry standards to identify key areas for improvement. These tools remove manual errors, provide standardized feedback loops, and enable rapid iteration with advanced Agile methodologies such as Continuous Integration and Delivery (CI/CD) and Scaled Agile Framework (SAFe). CodeOps even provides users with templates for configuring their own CI/CD pipelines within the platform, saving significant time, money, and stress.

Furthermore, team members can easily collaborate on source code development with CodeOps' distributed version-control system, while simultaneously tracking tasks and monitoring a project's issues, enhancements, and bugs in real time.

Operations teams are ultimately able to align more effectively with development teams, as well. CodeOps enables all users to maintain a coherent perspective on what's happening in their development project at any given point in time. Centralized project tracking, development, testing, and deployment allows Managers and Team Leads to precisely evaluate the state of their projects' code repositories and ongoing work. Executives can quickly visualize and generate reports on development progress and gain real-time insight into resource usage and performance—all of which enables quicker, more intelligent business decisions.



Arsenal's core platform is complemented by CodeOps, an out-of-the-box solution that includes all of the necessary tools to successfully **Code. Build. Test. Deploy.**



#### Automated pipelines

incorporate testing and validation every step of the way before merging code changes into the main branch.

#### Automated code reviews

further improve quality by assessing development projects against industry standards to identify key areas for improvement.

# Use Case

**OMNI CodeOps** streamlines management of distributed, non-linear software development.

## RESULTS



### Seamless Integration

CodeOps connects with all of the tools needed for development build, test, and deployment, which allows for every functionality to be operated and tracked from within the same platform.



### End-to-End Visibility

Development and operations teams alike can see project updates and changes as they happen. This allows for quick and accurate evaluations of what's happening in their development project at any given point in time.



### Better. Software. Faster.

Automated pipelines test, validate, and compare code to reduce errors, provide standardized feedback loops, identify areas for improvement, and enable rapid iteration.



### Time + Money Saved

In addition to improvements in efficiency and code quality, CodeOps lets software development teams avoid spending excessive time and money developing, implementing, and maintaining a CI/CD solution for themselves.



### Enhanced Collaboration

Shared task and release tracking keeps everyone effortlessly up-to-date and on the same page. CodeOps makes knowledge transfer and documentation easy by enabling users to contribute, review, and modify essential project information in real time.



### Improved Decision-Making

Both technical and non-technical decision-makers gain insight through precise DevOps real-time tracking and visualization of task completion, deployments, resource usage, performance, and overall progress.



CodeOps connects with all of the tools needed for development build, testing, and deployment.