

d-fine



analytical.
quantitative.
tech.



Front Arena DevOps

A modernised toolchain for efficient and robust
application development and maintenance

Modern DevOps tools and processes facilitate efficient software development for Front Arena¹

Radically changing customer behaviours, volatile markets and cost pressures are the new reality for trading floors. Front Arena IT teams have to adapt with efficient organisation and DevOps processes to deliver features faster and through a fine-tuned value chain. With some important differences between classical workflows and development in Front Arena, setting up an efficient development and integration toolchain is a challenge. This brochure describes how best practice approaches can be successfully adapted. Even small discrete steps can help to increase velocity and improve quality.

Change Management

A trading system requires complex stakeholder management. A clearly defined change management process (like ITIL) is critical to ensure transparent change methodology throughout an institution. Moreover, in such a trading system context, categorization of changes, including fast tracking of low risk standard changes, increases user satisfaction and relieves change management and developers to focus on important changes.

Toolchain

The change management process is facilitated by (and to some extent only possible through) a set of tools for automation. The different facets of such a toolchain work together to ensure a consistent process, and with less manual tasks, more tests can be performed quicker, improving both quality and speed.

Code Versioning

Successful Front Arena teams have custom Front Arena configuration and code objects under version control – from extension modules to server registry configurations. Not every object needs to be under version control but a clear object scope strategy is required. A single repository simplifies release and deployment management, whereas sub-repositories allow several release cycles and partial rollbacks. Every commit on the release or master branch should reference a development ticket.

Exemplary Toolchain

We have successfully used and extended the CI toolchains by Atlassian, Microsoft (Azure DevOps) and GitLab on various Front Arena DevOps projects. The following toolchain is based on Atlassian software.

◆ Jira Software

Issue tracking software that allows bug tracking and agile project management to

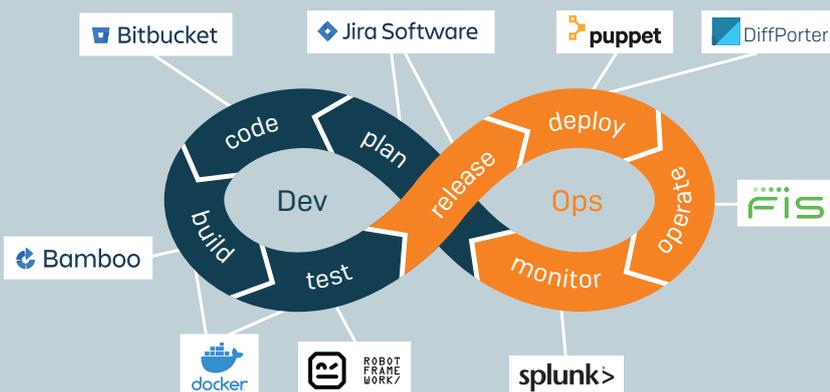
- » plan and review the sprints,
- » see pull requests and code changes per ticket,
- » assist the development and test process with the ticket workflow.

▣ Bitbucket

Source code hosting service with dedicated features for

- » coordination of parallel development branches (pull requests),
- » code reviews,
- » online editor for quick & secure business configuration changes.

¹ Note that Front Arena has been rebranded to "FIS Cross-Asset Trading and Risk Platform" in February 2021.



What is DevOps?

DevOps is a modern software engineering paradigm, which combines development and operation of a software application in a unified and continuous process. It enhances synergies and encourages the design of maintainable software. A well-developed toolchain and a high level of automation to handle recurring tasks in a standardized way underpins the DevOps process.



Open and extendable framework for acceptance test automation that separates

- » test case design (requires business expertise, no programming skills),
- » test code implementation (requires programming skills, no business know-how).



Service to define and run release pipelines including

- » automatic unit tests,
- » automatic code quality checks (Pylint / SonarQube),
- » environment management for test and production servers,
- » deployment audits,
- » emergency rollbacks.



Server configuration tool with a declarative language for state modelling of

- » custom component registries,
- » AFG, TAB, AMAS custom mapping or Python files,
- » server batch files,
- » Windows tasks.



Our extension of the Front Arena Transporter tool to avoid unnecessary code commits and deployments by

- » removing non-essential information (e.g. time stamps),
- » enhanced change comparison for various ADS objects.

Test Automation

The stability of a Front Arena system crucially depends on the rapid impact assessment of changes. Each change in functionality should therefore come with a new unit test and each change in business logic with a new acceptance test. Testing gaps shrink gradually with so-called characterization tests, which do not test what the code is supposed to do, but what the code currently does. These tests should run automatically on a freshly built Docker environment, triggered by the deployment pipeline on every push to the main branch.

Release Management

A continuous release management process requires a dedicated environment branch for each Front Arena environment, which is deployed to the respective environment on a daily basis. Every change is tracked in adherence to the change management guidelines (e.g. in JIRA) and the release to a particular environment is performed via a pull request to the environment branch. Continuous integration is the goal, but it is risky with insufficient test coverage. Clients should start with a regular release cycle and increase the frequency along with higher test coverage.

Deployment Automation

Deployment tools push the code and configuration artefacts to the target environments. They require a certain degree of sophistication to be performant. “Infrastructure as code” tools manage the deployment of all server changes such as registries, tasks or code. The Front Arena Transporter deploys ADS objects, but an extension may be required to commit changed objects only. Finally, the deployment process also manages the restart of the required components.

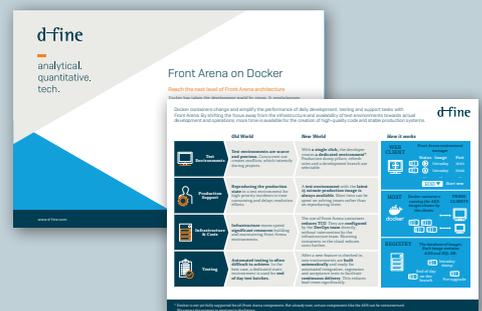
Environment Management

In order to leverage a continuous integration and delivery toolchain, it is necessary to provide a suitable environment infrastructure. A minimal set of environments consists of production, acceptance testing and development environments. Our clients often complement this setup with a so-called pre-production environment, which they continuously synchronize with production and monitor with similar attention. All other non-productive environments are regularly refreshed with production database snapshots. For non-integrated development environments, Docker allows an on-demand creation and minimizes development conflicts.

Environment Management with Docker

Docker is a powerful tool, which revolutionizes how an application is built and shipped. It helps to overcome conventional development limits and infrastructure silos and is therefore a valuable building block for the DevOps toolchain. Even though it will take time for Front Arena production environments to run on Docker, there are plenty of non-productive use cases already available today. You can flexibly create new environments consisting of a chosen ADS image and code branch. This helps to speed up development and test automation, and it can further improve the efficiency of production support.

Request our flyer “Front Arena on Docker” or ask us for a demo.





www.d-fine.com

Do you want to improve your Front Arena infrastructure and benefit from our toolchain and project experience?

Contact us:

Dr Matthias Braun, Senior Consultant
Matthias.Braun@d-fine.de

Dr Jakob Ditchen, Manager
Jakob.Ditchen@d-fine.ch

Dr Kai Eckert, Partner
Kai.Eckert@d-fine.ch

Dr Ruth Joachimi, Manager
Ruth.Joachimi@d-fine.de

Marc-Stephan Maenner, MBA, Manager
Marc-Stephan.Maenner@d-fine.ch

d-fine Berlin, Dusseldorf, Frankfurt, London, Munich, Vienna, Zurich

www.d-fine.com