

# Release notes ATM sprint 4

## Inhoudsopgave

|   |          |
|---|----------|
| <b>Sprint goals .....</b>                         | <b>3</b> |
| <b>Sprint achievements .....</b>                  | <b>3</b> |
| <b>Project management tool .....</b>              | <b>3</b> |
| <b>CI/CD pipelines and repository links .....</b> | <b>3</b> |
| <i>Platform .....</i>                             | <i>3</i> |
| <i>Strategy service .....</i>                     | <i>3</i> |
| <i>Test service .....</i>                         | <i>4</i> |
| <b>Sprint retrospective .....</b>                 | <b>4</b> |
| <b>Sprint planning .....</b>                      | <b>4</b> |

## Project management tool

For the project management tool we used [Jira](#).

## CI/CD pipelines and repository links

### Platform (frontend)

- Log in functionality (creating account or google login)
- Added a screen to import EA's and then run a test.
- <https://github.com/S-A-RB05/platform>

### Strategy service

- Initialized repository
- Added EA storing functionality
- <https://github.com/S-A-RB05/StratService>

### Test Manager service

- Initialized repository
- Can run tests by starting a docker container containing MetaTrader
- <https://github.com/S-A-RB05/TestManager>

### Docker image pipelines

- When pushing to the main branch of the repo, a docker image gets built and is put onto docker hub, making it easily accessible for the stakeholder. This is made for the Platform, StratService and TestManager Service.
- <https://github.com/S-A-RB05/StratService/actions/workflows/docker-image.yml>
- <https://github.com/S-A-RB05/TestManager/actions/workflows/docker-image.yml>
- <https://github.com/S-A-RB05/platform/blob/main/.github/workflows/docker-image.yml>

### Go build and test pipelines

- The pipeline tries to build the project when pushing, making sure the build doesn't have any build errors. It then runs any tests that the service has.
- <https://github.com/S-A-RB05/StratService/actions/workflows/go.yml>
- <https://github.com/S-A-RB05/TestManager/actions/workflows/go.yml>

With these pipelines in place, assuming the code is covered by tests, we can automate the process of checking functionality, avoiding build errors and the deployment of the application.

## Sprint goals

As a trader I want to be able to upload my trading strategies so I can pass them down to the test program

- As a trader I want to be able to run a strategy test so I can test strategies
- As the system I want to scale the number of tests that run so a user can easily run a lot of tests
- As a trader I want the test results to be saved in a database so I can always look them back up again.
- As a trader I want to select certain strategies so I can choose which strategies to test
- As a trader I want to see test results on the platform

## Sprint achievements

- As a trader I want to be able to run a strategy test so I can test strategies
- As the system I want to scale the number of tests that run so a user can easily run a lot of tests
- As a trader I want the test results to be saved in a database so I can always look them back up again.
- As a trader I want to select certain strategies so I can choose which strategies to test

## Sprint retrospective

Keep doing:

- Flow goed afgemaakt
- Transformeren van report naar data
- Eindstand product

Should Improve:

- Niet aan advice toegekomen
- Niet alle user stories helemaal kunnen implementeren

## Sprint planning

For sprint 4 we will be working on completing the flow.

- As a trader I want to be able to run a strategy test so I can test strategies
- As the system I want to scale the amount of tests that run so a user can easily run a lot of tests
- As a trader I want to see test results on the platform
- As a trader I want the test results to be saved in a database so I can always look them back up again.
- ~~As a trader I want to receive advice based on the saved test results so I can determine the best strategy~~