

# Release notes ATM sprint 2

## 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>

## Sprint goals

- As a trader I want to be able to login on the platform so I can watch and test my own strategies
- 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 to test different variable ranges in my strategies so the system can determine all possible outcomes

## Sprint achievements

- Metatrader config generation
- Dynamic strategy variable parser
- Strategy variable adjustability
- Continuously running Metatrader a docker with kubernetes
- High availability kubernetes clusters
- Progress on research
- GRPC server (stratservice)

## 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 retrospective

### Keep doing:

- Effective standups every day
- Good research done
- Good collaboration
- Being on time

### Should Improve:

- Slightly less functionality than previous sprint
- Research took a little longer than expected



## Sprint planning

For sprint 3 we will be working on completing the flow. This involves working on the test service and connecting this with the strategy service. After that we will be looking into getting advice from the test that are running. This has priority after the flow is done.

The user stories we are going to work on are:

- As a trader I want 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 to test different variable ranges in my strategies so the system can determine all outcomes
- 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