

GEOMETRIC DRAWING PROGRAM

Final presentation of Software Engineering Project

Group no. 8

Campagnano Vincenzo, Corcione Marta, D'Onofrio Francesco, Franco Paolo

Objectives

Develop a JavaFX Application for design and realize geometrical drawings

- User Friendly GUI
- Develop valuable code
- Easy to extend
- Follow a test driven development
- Resolve typical problems using design patterns
- Design the application following an architectural pattern

Tools

Technologies used for the development of the project

- GIT as version control system
- TRELLO for developing collaborative projects
- MS Teams for communication in remote
- NetBeans IDE with Java 8
- JUnit4 for testing
- Google Drive for sharing documents

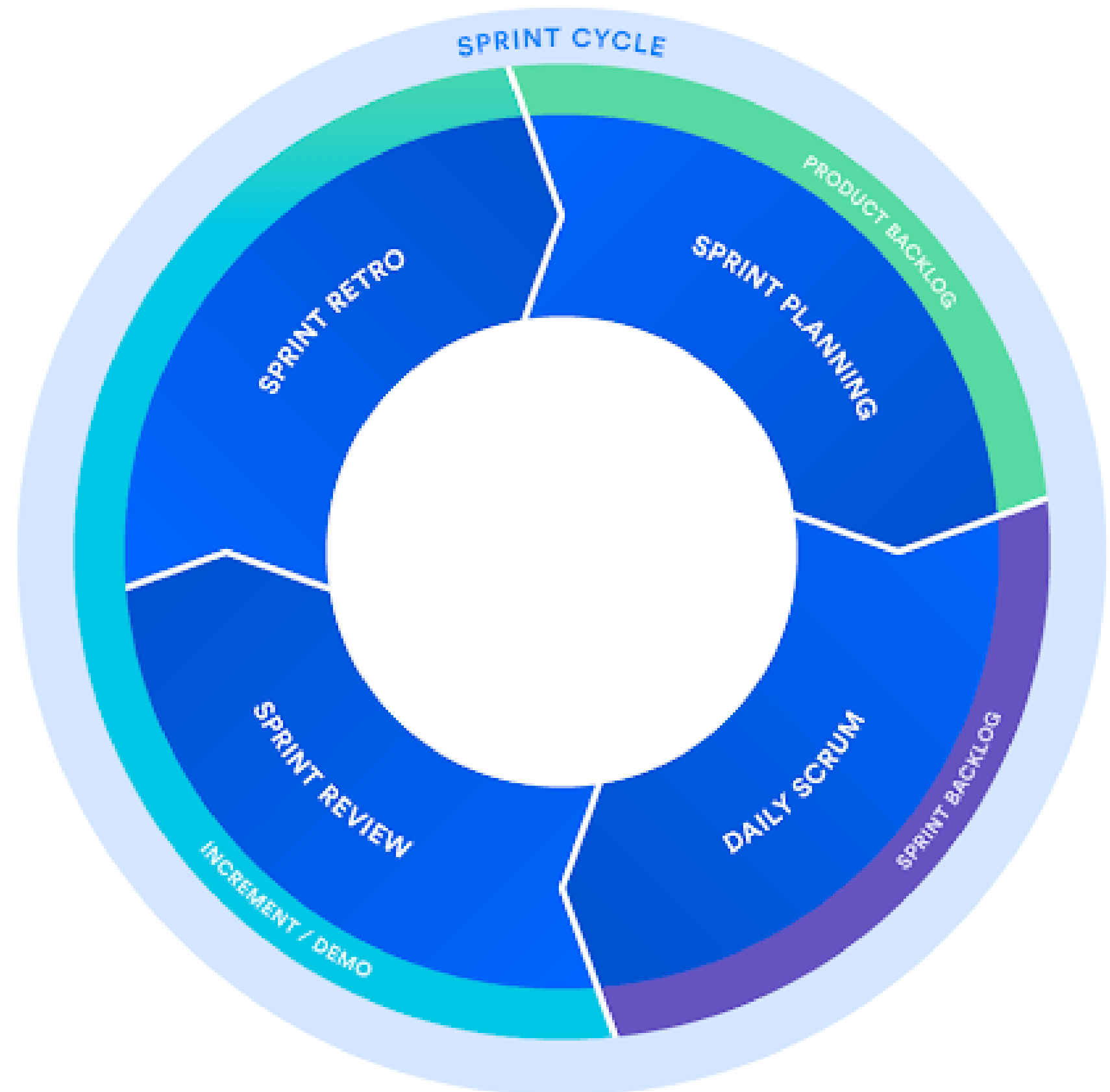




MAIN PRINCIPLES

1. Control over the empirical process
2. Self-organization
3. Collaboration
4. Value-based prioritization
5. Time-boxing
6. Iterative Development

SCRUM



Pivotal points of development



User Stories

Defined by the team through the User Epic related to the Product Owner

Architecture

Model-View-Controller architectural pattern to separate data from view and organize the classes

Test Driven Development

Focuses on creating unit test cases before developing the actual code

User Feedback

Product Owner have a view of the project and meets with the development team on a daily basis, to share customer feedback and insight

First sprint

Design of the project

Architecture

Model view controller pattern, to spilt the business logic from the graphic one

Requirement analysis

Definition and prioritization of the User Stories, reading and interpreting the client's requirements

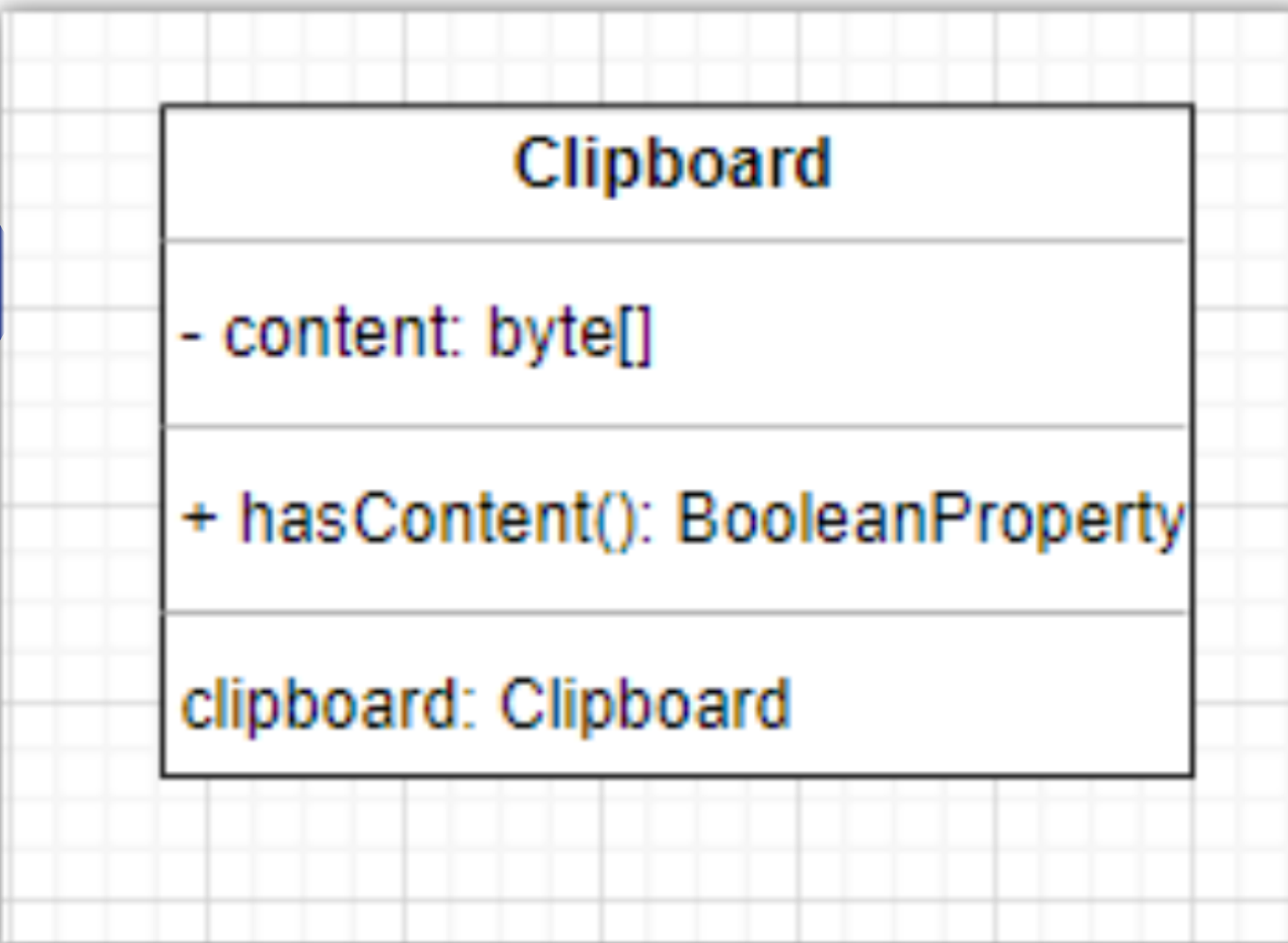
Definition of done

Done means that every task under the User Story has been completed and tested and any work created is attached to the User Story so the Product Owner can review it and make sure it meets his or her expectations.

SINGLETON

Design patterns

The singleton design pattern is used to be sure that only one instance of the clipboard can be instantiated



```
classDiagram
    class Clipboard {
        -content: byte[]
        +hasContent(): BooleanProperty
    }
    clipboard: Clipboard
```

Clipboard

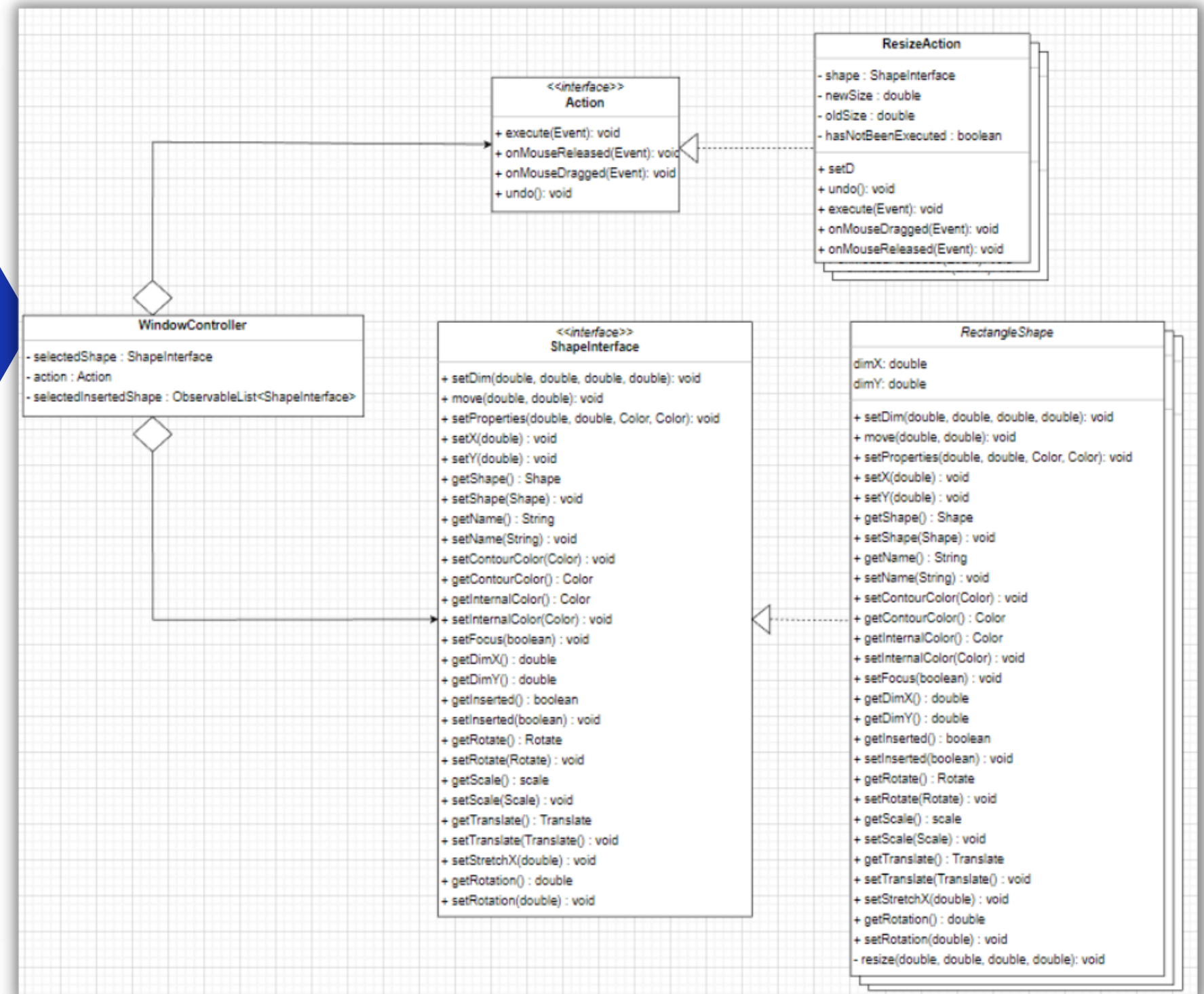
- content: byte[]
- + hasContent(): BooleanProperty

clipboard: Clipboard

STATE

The state design pattern is used to memorize various choices of the user, like the shape selected for inserting or the one selected from the drawing, and for memorizing the action chosen by the user

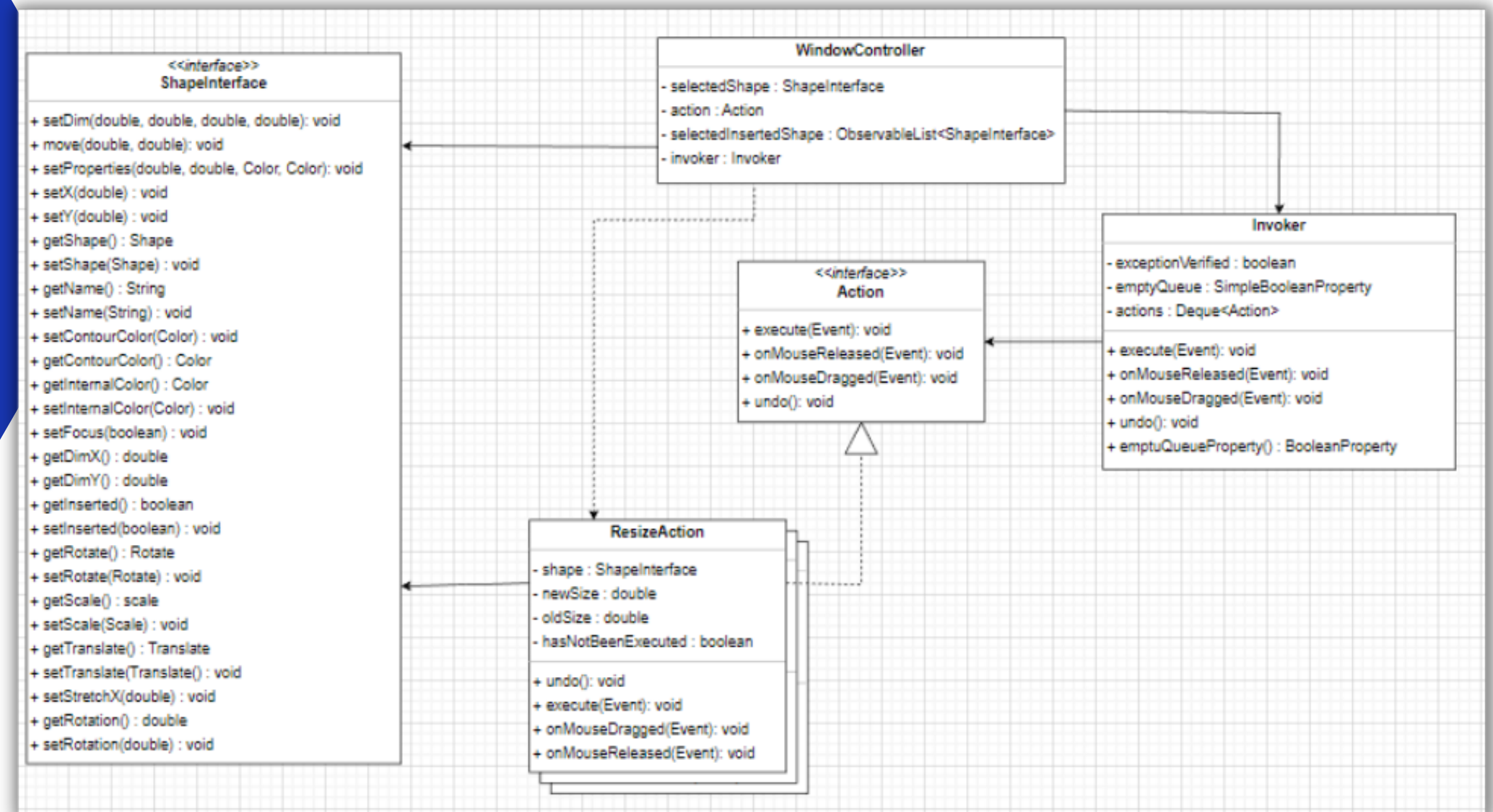
Design patterns



COMMAND

The command design pattern is used to parameterize all the actions to execute and support the undoing of done actions

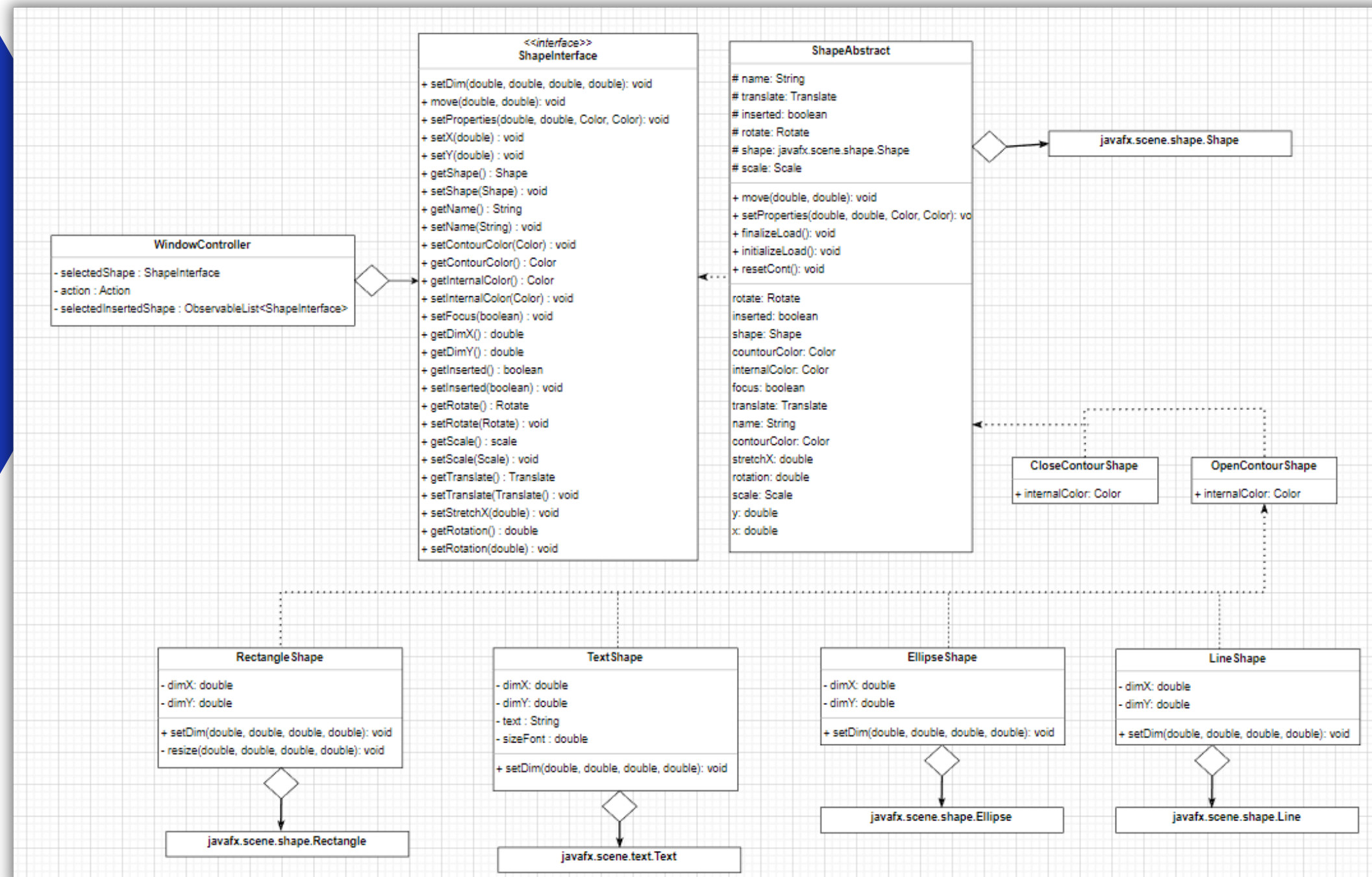
Design patterns



ADAPTER (Object)

The object design pattern is used to adapt the public methods of the default JavaFX shapes to an interface defined by the developers

Design patterns



Second sprint

Product updates

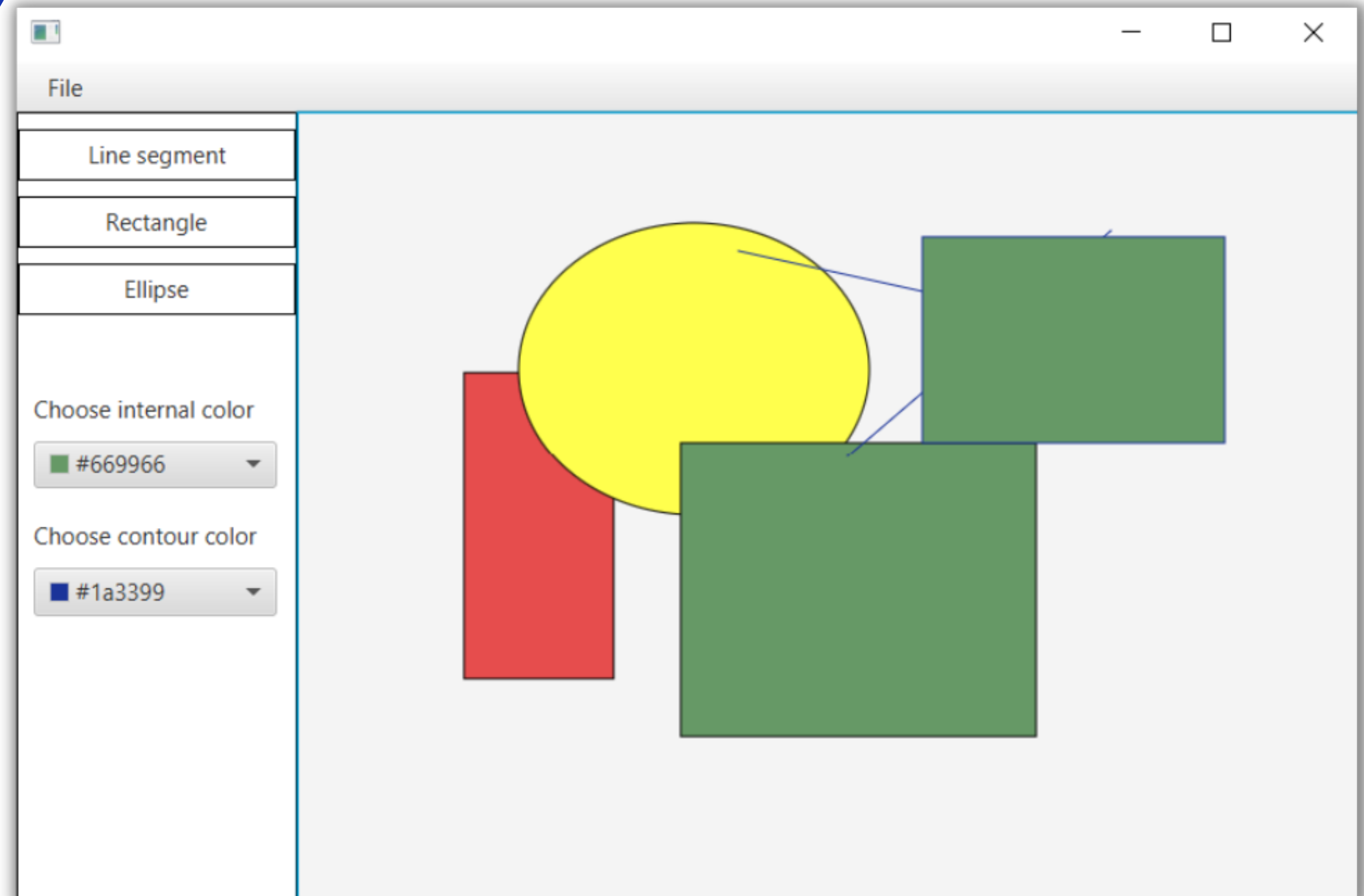
Basic operations on the drawing window

Basic GUI with a few operations

- Very simple GUI made of a drawing window and a shapes section from which choose a shape to insert

Save and load capabilities

- Possibility to save on a binary file the content of the drawing window and load a drawing from it, so it becomes possible to continue drawing among different execution of the program



Second Sprint

Sprint Review and Retrospective

Process Analysis

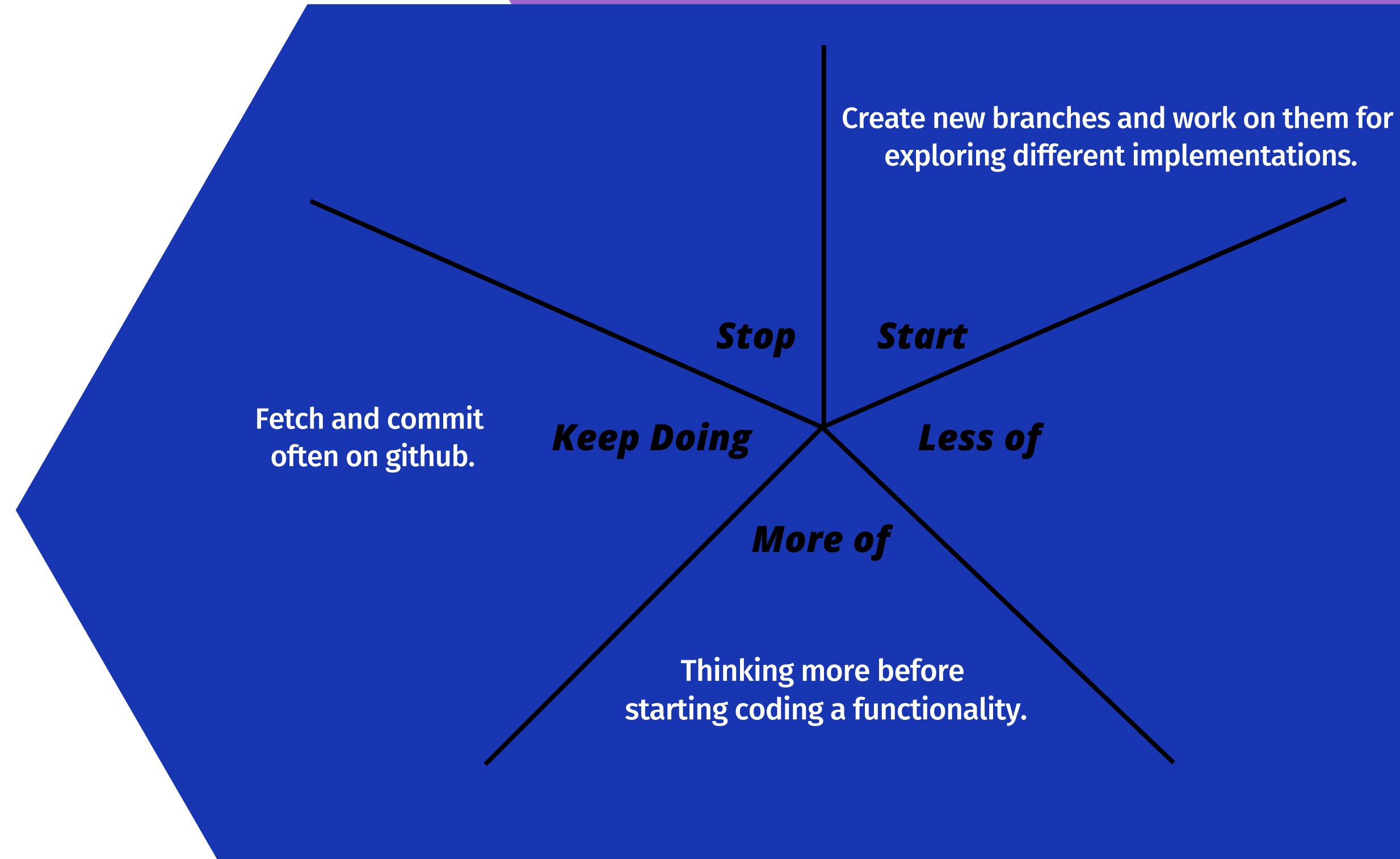
User Stories planned: 13
Corresponding story point: 34

User Stories completed: 15
Corresponding story point: 41

Estimated initial Project Velocity: 30
Measured Project Velocity: 41

Added 2 user stories

Found 2 bugs and 2 technical debts



Third sprint

Product updates

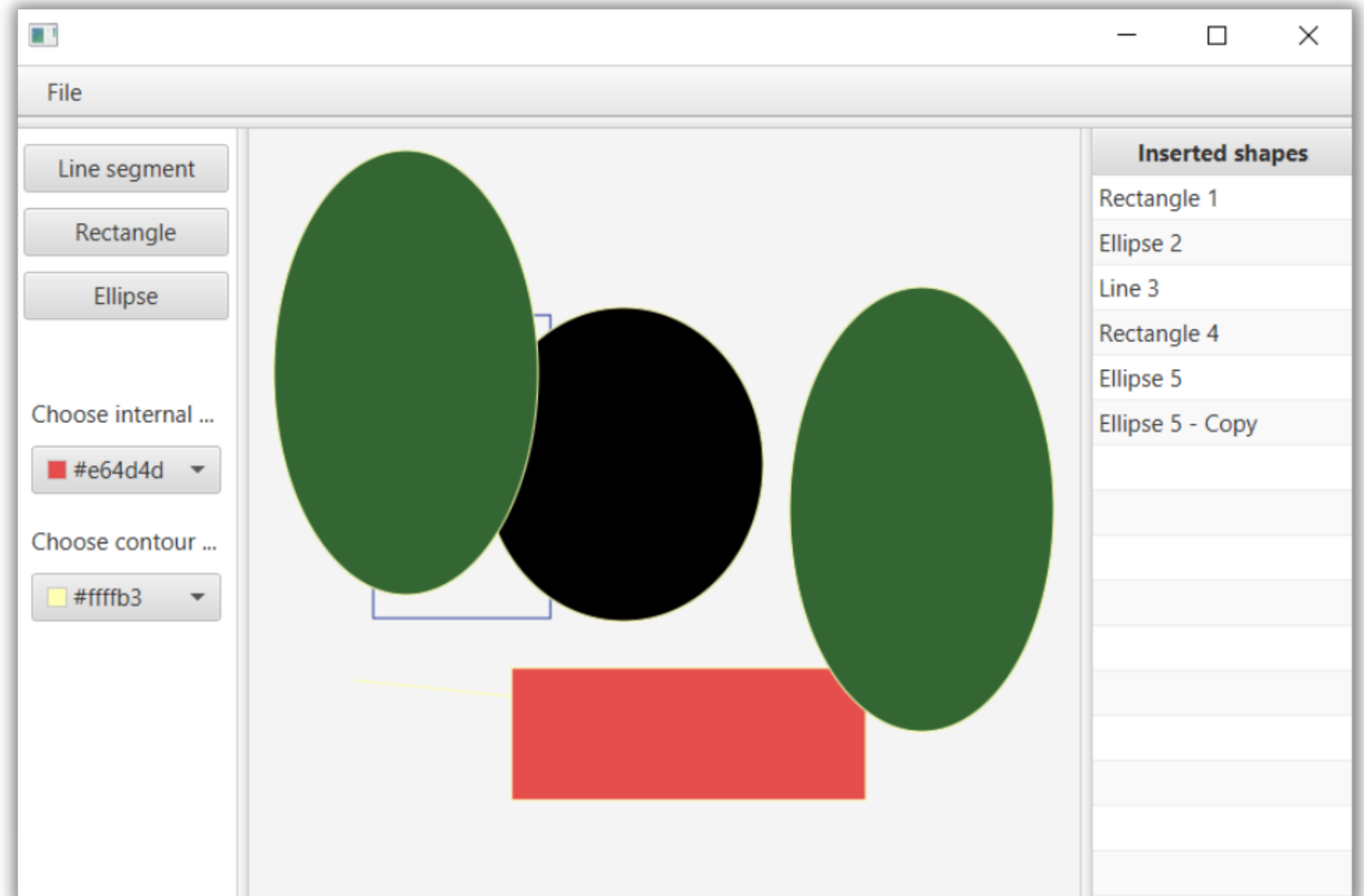
Shape Editing

New operations

- **Added move, focus, change internal and contour color, resize, copy and paste operations**

List of shapes

- On the right side of the program now there is a table filled with the shapes currently inserted in the drawing



Third Sprint

Sprint Review and Retrospective

Process Analysis

User Stories planned: 12
Corresponding story point: 50

User Stories completed: 10
Corresponding story point: 40

Estimated initial Project Velocity: 41
Measured Project Velocity: 40

Added 2 user stories

Found 3 bugs and 1 technical debt
Solved 2 bugs and 4 technical debts



Fourth sprint

Product updates

More Shapes and Editing Operations

Text shape

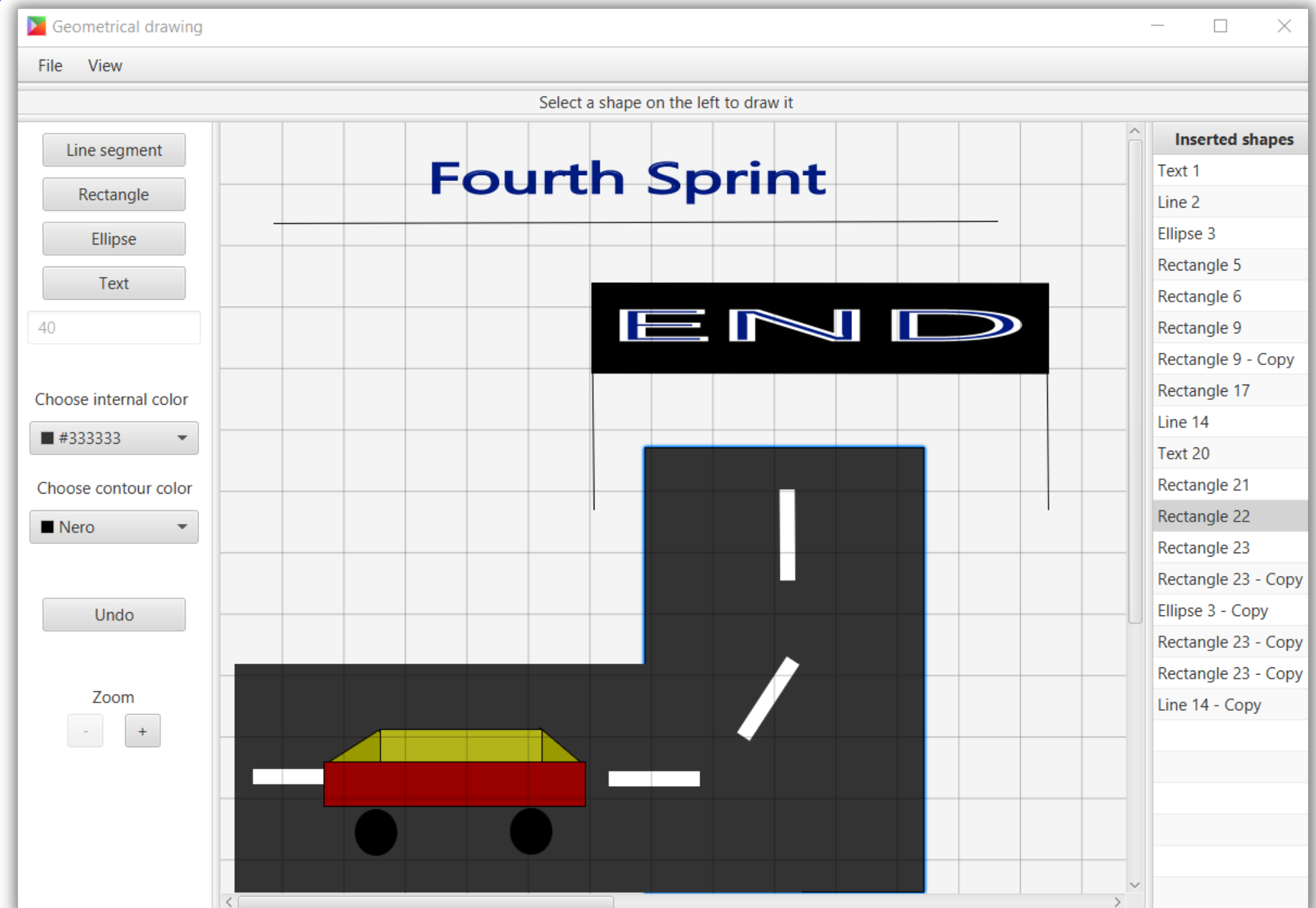
- Now a text shape is available, in which the user can freely write anything

New operations and undo

- Added cut, rotate, mirror, stretch and set font size operations with the possibility to undo executed operations

Grid on the drawing

- The user can now place a custom size grid to help him drawing new shapes



Fourth Sprint

Sprint Review and Retrospective

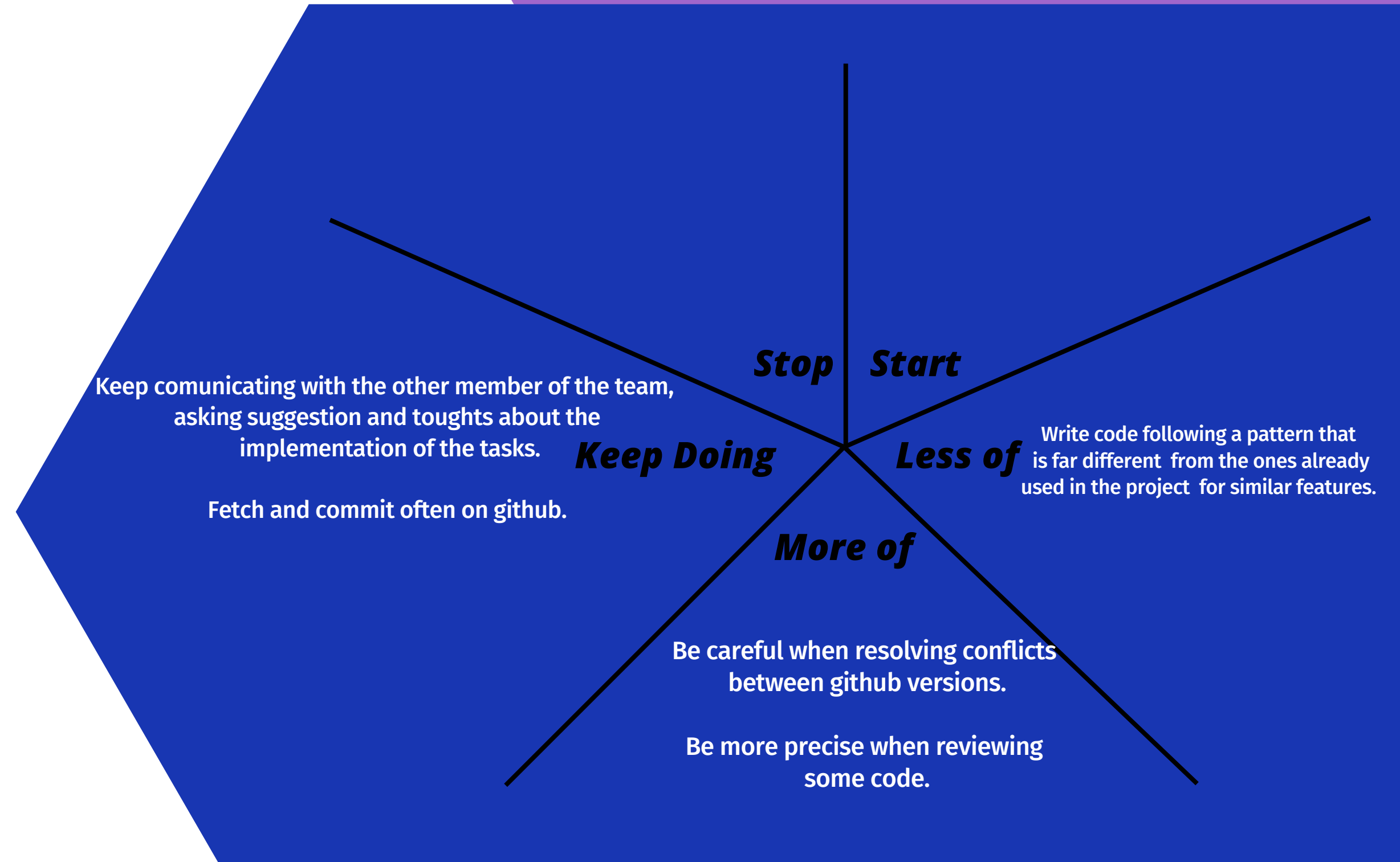
Process Analysis

User Stories planned: 16
Corresponding story point: 42

User Stories completed: 15
Corresponding story point: 41

Estimated initial Project Velocity: 42
Measured Project Velocity: 41

Found 2 bugs and 1 technical debt
Solved 4 bugs and 1 technical debts



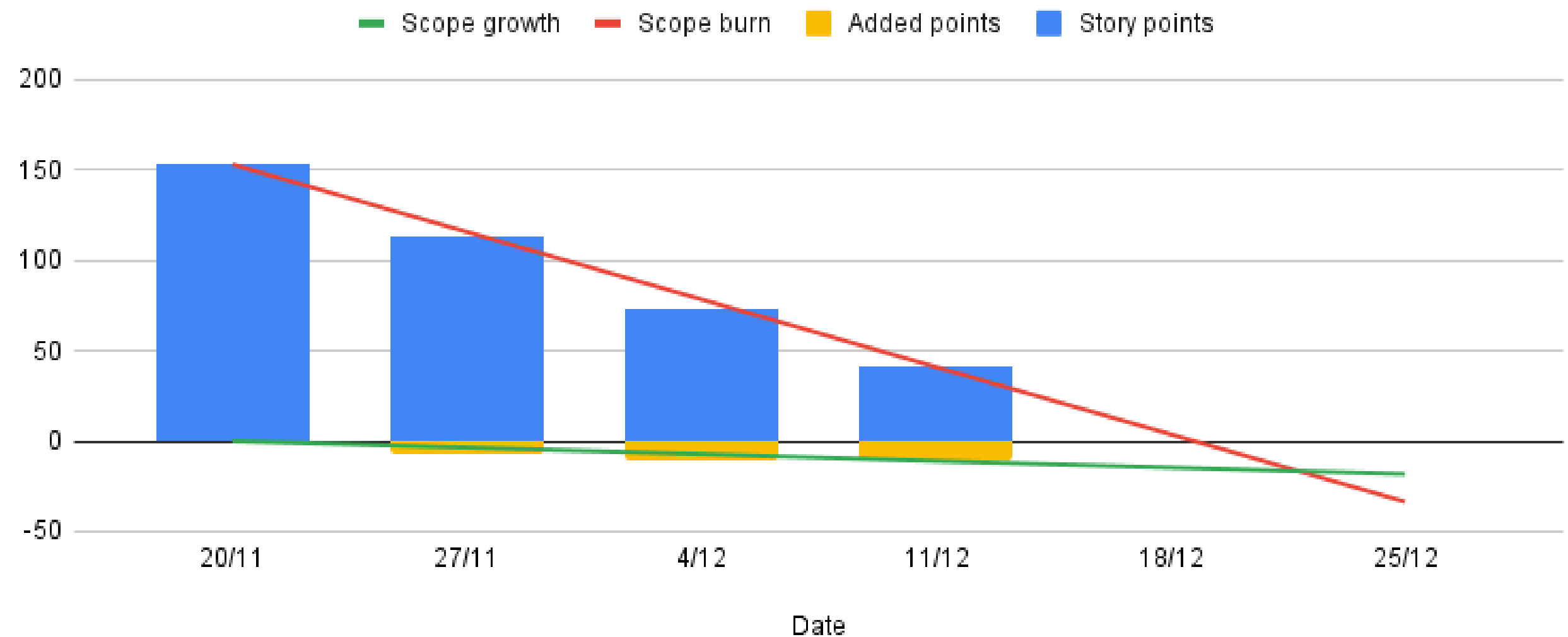
Burndown chart

Paolo version

Likely release date

Through analyzing the burndown chart it's possible to estimate a likely release date it's foreseen to be around the 20th of December.

Burndown chart



The image features a white background with several large, overlapping hexagonal shapes in the bottom-left corner. These shapes are colored in shades of purple, dark blue, and light blue. The text 'Thank you!' is positioned on the right side of the image in a bold, dark blue font.

Thank you!