**Sommario**

[PRE GAME 2](#_Toc153102370)

[Conventions and tecnical set-up 2](#_Toc153102371)

[Architecture 2](#_Toc153102372)

[Interface 2](#_Toc153102373)

[UML diagram 3](#_Toc153102374)

[WHOLE UML 3](#_Toc153102375)

[UML SEPARATED PART-RULE 3](#_Toc153102376)

[UML SEPARATED PART-TRIGGER 4](#_Toc153102377)

[UML SEPARATED PART-ACTION 4](#_Toc153102378)

[Product backlog 5](#_Toc153102379)

[Definition of done 7](#_Toc153102380)

[BurnDown chart 7](#_Toc153102381)

[SPRINT 1 10](#_Toc153102382)

[sprint backlog 10](#_Toc153102383)

[sprint review 11](#_Toc153102384)

[sprint RETROSPECTIVE 12](#_Toc153102385)

[SPRINT 2 13](#_Toc153102386)

[sprint backlog 13](#_Toc153102387)

[sprint review 15](#_Toc153102388)

[sprint RETROSPECTIVE 16](#_Toc153102389)

[SPRINT 3 17](#_Toc153102390)

[sprint backlog 17](#_Toc153102391)

[sprint review 18](#_Toc153102392)

[sprint RETROSPECTIVE 19](#_Toc153102393)

# **PRE GAME**

## Conventions and tecnical set-up

* Programming language: Java;
* IDE: NetBeans;
* Graphic interface: JavaFX;
* Testing: JUnit 4;
* SCRUM design: Trello;
* Coding conventions: Oracle Java coding conventions
* Methods and variables: lowerCamelCase;
* Classes, interfaces, exceptions: CamelCase;
* Constants: CAPITAL LETTERS.

## Architecture

For this project was choosen the **Model-View-Control (MVC)** architectural pattern. In which there are three types of components:

* Model: Set of classes, specifically a subsystem that handles the application logic independently of the user interface;
* View: Takes care of displaying the information received from the model;
* Controller: Translates the input (e.g. pressing a button) provided by the user into requests that are made to the Model and that can lead to an update of the View.

## Interface

**Immagine che contiene testo, schermata, linea, numero

Descrizione generata automaticamente**

## UML diagram

### WHOLE UML

Immagine che contiene diagramma, Disegno tecnico, Piano, schematico

Descrizione generata automaticamente

### UML SEPARATED PART-RULE

Immagine che contiene testo, diagramma, Piano, Parallelo

Descrizione generata automaticamente

### UML SEPARATED PART-TRIGGER

Immagine che contiene testo, diagramma, Piano, linea

Descrizione generata automaticamente

### UML SEPARATED PART-ACTION

Immagine che contiene testo, diagramma, Piano, linea

Descrizione generata automaticamente

## Product backlog

Priority: High, Medium-high, Medium, Medium-low, Low

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#ID** | **DESCRIPTION** | **CATEGORY** | **DEPENDENCIES** | **STORY POINT** | **ACCEPTANCE CRITERIA** | **PRIORITY** | **STATUS** |
| **000** | The application is a single-user desktop program that allows the creation of rules with triggers based on a specific time. When the current time reaches the trigger, the program performs an action, such as playing an audio or displaying a message in a dialogue box. This control occurs every few seconds. | User epic | - | - | - | - | - |
| **001** | **As** user,  **I want** to create a set of rules,  **So that** I can automate a certain action. | User story | - | 8 | • **When** the user selects the alarm action, **I want** the fields to appear for choosing the time and the audio file.  • **When** the user selects the reminder action, **I want** the fields to appear to choose the time and to write the text. | High | Done |
| **002** | **As** user**,**  **I want** to decide the time at which the action is to be executed,  **So that** the system can automatically check this time and activate the associated rule. | User story | #001 | 2 | • **When** time is reached, **I want** the action to be executed once.  • **When** an invalid time is entered, **I want** the rule not to be set. | High | Done |
| **003** | **As** user,  **I want** messages to be displayed in a dialogue box that requires an explicit action to be closed,  **So that** I cannot miss reminders or alerts. | User story | #001, #002 | 1 | • **When** the time is reached, **I want** the preset dialog box to be displayed.  • **As long as** the user does not click on the specific button, **I want** the window to remain open. | Medium-high | Done |
| **004** | **As** a user,  **I want** to be able to choose an audio file to be played when a rule is activated,  **So that** I can have a customized sound reminder. | User story | #001, #002 | 3 | • **When** the user chooses to set an alarm, **I want** them to be able to choose an audio file as ringtone.  • **When** the time is reached, **I want** the sound file chosen by the user to be played. | Medium-high | Done |
| **005** | **As** user,  **I want** to be able to view past rules at program start-up,  **So that** I can keep track of them. | User story | - | 5 | • **When** the program is started, **I want** the active/terminated rules to be visible on the application interface.  • **When** I insert a new rule, **I want** it to be visible in the table and added to the file. | Medium-high | Done |
| **006** | **As** user,  **I want** to be able to delete previously entered rules,  **So that** I can manage the rules | User story | #005 | 3 | • **When** I have deleted the rule, **I want** it to no longer be displayed in the interface table.  • **When** I have deleted the rule, **I want** the action associated with the deleted rule not to be triggered when the time is reached.  • **When** I have deleted the rule, **I want** that rule to no longer appear in the file. | Medium-high | Done |
| **007** | **As** user,  **I want** to be able to deactivate and reactivate a previously entered rule,  **So that** I can manage the rules | User story | #005 | 5 | • **When** I have deactivated the rule, **I want** it to be visible in the interface table with inactive status.  • **When** I have deactivated the rule, **I want** checks on that rule to not occur until I reactivate it.  • **When** I have reactivated the rule, **I want** it to be visible in the interface table with active status.  • **When** I deactivated or reactivated the rule, **I want** the status to be updated on the file | Medium-high | Done |
| **008** | **As** user,  **I want** to be able to choose whether to repeat a rule several times by specifying a waiting time (in days, hours, minutes),  **So that** I can manage when to repeat the rules. | User story | #001 | 5 | • **When** I have set the rule for once, **I want** the action to happen at that instant and then be deactivated.  • **When** I have set the rule for several times, **I want** the action to be re-executed after the waiting time has elapsed. | Medium | Done |
| **009** | **As** user,  **I want** to be able to write a specific string to the end of a specified file when the rule is activated,  **So that** I can update my files with new information. | User story | - | 5 | • **When** I have specified the string to be written and the file, **I want** the string to be written at the end of that file once the trigger is verified. | Medium | Done |
| **010** | **As** user,  **I want** to be able to copy or move a file from one directory to another when the rule is activated,  **so that** I can manage files according to my needs. | User story | - | 5 | • **When** I have specified the file and destination directory to which it is to be copied, **I want** the file to be present on both the source and destination directories once the trigger has been verified.  • **When** I have specified the file and destination directory to which it is to be moved, **I want** the file to be present only in the destination directory once the trigger has been verified. | Medium | Done |
| **011** | **As** user,  **I want** to be able to delete a file from a directory when the rule is activated,  **So that** I can maintain the efficiency of my storage space. | User story | - | 3 | • **When** I choose the file to be deleted in a specific directory, **I want** the file to no longer be present in that directory once the trigger has occurred. | Medium | Done |
| **012** | **As** user,  **I want** to be able to execute an external program with specified command line arguments,  **So that** I can automate external processes. | User story | - | 13 | • **When** I choose an external program and specify the arguments to be passed, **I want** the program to be executed with these arguments when the trigger is triggered. | Medium-low | Done |
| **013** | **As** user,  **I want** to be able to choose a specific day of the week or month or a specific date on which the action is to be performed,  **So that** the system can automatically check the day or date and activate the associated rule. | User story | - | 3 | • **When** I specify the day of the week or month, **I want** the associated action to be executed when that day arrives.  • **When** I specify the date, **I want** the associated action to be executed when that date arrives.  • **When** I specify an invalid day or date, **I want** the associated action not to be executed. | Medium-low | Done |
| **014** | **As** user,  **I want** to be able to perform a specified action when I choose a file with a specified name in a specified directory,  **So that** I can automate file management | User story | - | 5 | • **When** I specify a file in a directory, **I want** the system to perform the action only if that file exists. | Low | Done |
| **015** | **As** user,  **I want** to be able to specify a file and a maximum size,  **So that** I can manage the storage space. | User story | - | 8 | • **When** I select a file, **I want** the associated action to be executed if it exceeds a size limit. | Low | Done |
| **016** | **As** user,  **I want** certain actions to be automatically executed after the execution of an external program with specific command-line arguments,  **So that** I ensure that these actions are only triggered after the external program has finished with a particular output state. | User story | - | 8 | • **When** an external program with specific arguments terminates with a particular output state, **I want** a specific action to be executed. | Low | To do |
| **TD1** | Repetition of code within the SetOfRules class. Specifically:  *FileManagement.saveRulesToFile(rules);* | Technical debt | - | 1 | - | Medium | Done |
| **TD2** | Using the Observer pattern to optimize rule state change on file. | Technical debt | - | 3 | - | Medium-high | Done |
| **TD3** | Re-implement the trigger and action classes using the pattern Factory method. | Technical debt | - | 8 | - | Medium-high | Done |
| **TD4** | Replace the Service with the Thread class in Java | Technical debt | - | 8 | - | Medium-high | Done |
| **TD5** | Replace the ObservableList implementing the pattern Observer | Technical debt | - | 13 | - | High | Done |
| **B1** | Bug in the rule repetition in user story 008. | Bug | - | 1 | - | Medium-high | Done |
| **B2** | Bug in the ActionAppendFile test class | Bug | - | 2 | - | High | Done |
| **B3** | Bug when adding a repeated action before a non-repeated action | Bug | - | 2 | - | Medium-high | Done |

## Definition of done

The definition of done below are valid for all the user stories:

* Tests were performed and successful;
* The product backlog has been updated;
* The code has been reviewed by all other team members;
* The user story implementation meets ALL acceptance criteria.

## BurnDown chart

Within the Burndown chart, only the story points of the user stories were included, to see the actual progress of the application. However, days when technical debts and bugs were resolved were specified in a special column, so there was no advancement in the user stories.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TIME |  | STORY POINT |  |  |
| **DAY** | **DATE** | **PLANNED** | **ACTUAL** | **NOTE** |
| Monday | 20/11/23 | 90 | 90 |  |
| Tuesday | 21/11/23 | 87 | 86 |  |
| Wednesday | 22/11/23 | 84 | 82 |  |
| Thursday | 23/11/23 | 81 | 81 |  |
| Friday | 24/11/23 | 78 | 76 |  |
| Saturday | 25/11/23 | 75 | 71 |  |
| Sunday | 26/11/23 | 72 | 71 |  |
| Monday | 27/11/23 | 69 | 71 |  |
| Tuesday | 28/11/23 | 66 | 71 | Technical debts and bugs were solved |
| Wednesday | 29/11/23 | 63 | 63 |  |
| Thursday | 30/11/23 | 60 | 55 |  |
| Friday | 01/12/23 | 57 | 50 |  |
| Saturday | 02/12/23 | 54 | 45 |  |
| Sunday | 03/12/23 | 51 | 45 |  |
| Monday | 04/12/23 | 48 | 45 | Technical debts and bugs were solved |
| Tuesday | 05/12/23 | 45 | 45 | Technical debts and bugs were solved |
| Wednesday | 06/12/23 | 42 | 31 |  |
| Thursday | 07/12/23 | 39 | 24 |  |
| Friday | 08/12/23 | 36 | 22 |  |
| Saturday | 09/12/23 | 33 | 19 |  |
| Sunday | 10/12/23 | 30 | 19 |  |
| Monday | 11/12/23 | 27 | 19 |  |
| Tuesday | 12/12/23 | 24 | 19 |  |
| Wednesday | 13/12/23 | 21 | 19 |  |
| Thursday | 14/12/23 | 18 | 19 |  |
| Friday | 15/12/23 | 15 | 19 |  |
| Saturday | 16/12/23 | 12 | 19 |  |
| Sunday | 17/12/23 | 9 | 19 |  |
| Monday | 18/12/23 | 6 | 19 |  |
| Tuesday | 19/12/23 | 3 | 19 |  |
| Wednesday | 20/12/23 | 0 | 19 |  |

# **SPRINT 1**

## sprint backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **USER STORY/ACCEPTANCE CRITERIA** | **TASK** | **STORY POINTS** | **OWNER** | **PLANNED HOURS** |
| **001** | **As** user,  **I want** to create a set of rules,  **So that** I can automate a certain action. |  | 8 |  |  |
| • **When** the user selects the alarm action, **I want** the fields to appear for choosing the time and the audio file.  • **When** the user selects the reminder action, **I want** the fields to appear to choose the time and to write the text. | Create an initial UI visible to the user in which there is a table summarizing the rules and a menu for choosing the rule. |  | Viktor, Pasquale | 2 |
| Implement code to show only the fields pertaining to the alarm setting, if selected. | Sara | 3 |
| Implement the code to display the fields pertaining to the reminder setting, if selected. | Pasquale | 3 |
| Create the interfaces: Rule, Trigger e Action | Elena | 2 |
| Class implementations TriggerTime, ActionMemo, ActionAlarm, SingleRule e SetOfRules | Sara | 2 |
| **002** | **As** user**,**  **I want** to decide the time at which the action is to be executed,  **So that** the system can automatically check this time and activate the associated rule. |  | 2 |  |  |
| • **When** time is reached, **I want** the action to be executed once.  • **When** an invalid time is entered, **I want** the rule not to be set. | Implement in a Service the function that checks, every 10 seconds, whether we have arrived at the time to execute the rule. |  | Viktor | 1,5 |
| Implement in a Service the function that executes the rule, only once when the time is reached. | Pasquale | 2 |
| Implement the check on the time entered (example of invalid time: 25:25, AB:00, 00:AB, 18:65, etc.). | Viktor | 0,5 |
| Changing the status (active/terminated) of the rule in the table once finished | Elena | 1 |
| Integration of the various tasks | Sara | 2 |
| Creation of test classes for each task | Each member tests what they write | 2 |
| **003** | **As** user,  **I want** messages to be displayed in a dialogue box that requires an explicit action to be closed,  **So that** I cannot miss reminders or alerts. |  | 1 |  |  |
| • **When** the time is reached, **I want** the preset dialog box to be displayed.  • **As long as** the user does not click on the specific button, **I want** the window to remain open. | Implementing the dialogue box UI (example: pop-up) |  | Pasquale | 1,5 |
| Implement the function that opens and closes the dialogue box once we have arrived at the right time, using the function created in user story 002. | Elena | 2 |
| Integration of the various tasks | Viktor | 1 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **004** | **As** a user,  **I want** to be able to choose an audio file to be played when a rule is activated,  **So that** I can have a customized sound reminder. |  | 3 |  |  |
| • **When** the user chooses to set an alarm, **I want** them to be able to choose an audio file as ringtone.  • **When** the time is reached, **I want** the sound file chosen by the user to be played. | Implement the functionality of choosing the audio file from the user's computer. |  | Sara | 3 |
| Implement the function to play the audio file. | Elena | 3 |
| Implement control over the type of file selected by the user, not allowing formats such as .pdf, .txt, .csv, etc. | Viktor | 0,5 |
| Integration of the various tasks | Pasquale | 1 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **005** | **As** user,  **I want** to be able to view past rules at program start-up,  **So that** I can keep track of them. |  | 5 |  |  |
| • **When** the program is started, **I want** the active/terminated rules to be visible on the application interface.  • **When** I insert a new rule, **I want** it to be visible in the table and added to the file. | Implement the event that inserts the set rule into the table of active/terminated rules. |  | Viktor | 1,5 |
| Implement the retrieval of past rules from file, so that they can be inserted into the table once the program is reopened |  | Pasquale | 1,5 |
| Integration of the various tasks |  | Elena | 1 |
| Creation of test classes for each task |  | Each member tests what they write | 1 |

## sprint review

**User stories done**

All planned user stories have been completed: 001, 002, 003, 004, 005.

**User stories added**

None

**User stories deleted**

None

**Technical debts**

• Initially, much of the code was placed in the interface controller, which led to coupling problems and problems in testing the code. Therefore it was decided to divide the methods implemented in the controller, where possible, into special java classes (e.g. for trigger control and action execution).

•There is a repetition of code within the SetOfRules class. Specifically:

*FileManagement.saveRulesToFile(rules);*

**Bugs**

None

**Sprint velocity**

Sprint velocity: 19

## sprint RETROSPECTIVE

**Immagine che contiene testo, Carattere, diagramma, schermata

Descrizione generata automaticamente**

# **SPRINT 2**

## sprint backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **USER STORY/ACCEPTANCE CRITERIA** | **TASK** | **STORY POINTS** | **OWNER** | **PLANNED HOURS** |
| **006** | **As** user,  **I want** to be able to delete previously entered rules,  **So that** I can manage the rules |  | 3 |  |  |
| • **When** I have deleted the rule, **I want** it to no longer be displayed in the interface table.  • **When** I have deleted the rule, **I want** the action associated with the deleted rule not to be triggered when the time is reached.  • **When** I have deleted the rule, **I want** that rule to no longer appear in the file. | Implement the onBtnDelete() method of the Controller class (delete the rule from the list/table/file) |  | Pasquale | 2 |
| Implement cancellation confirmation | Pasquale | 2 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **007** | **As** user,  **I want** to be able to deactivate and reactivate a previously entered rule,  **So that** I can manage the rules |  | 5 |  |  |
| • **When** I have deactivated the rule, **I want** it to be visible in the interface table with inactive status.  • **When** I have deactivated the rule, **I want** checks on that rule to not occur until I reactivate it.  • **When** I have reactivated the rule, **I want** it to be visible in the interface table with active status.  • **When** I deactivated or reactivated the rule, **I want** the status to be updated on the file | Implement the onBtnOnOff method of the Controller class (based on the click of the on/off button, the rule shall become active/deactive and change the state of the rule in the file). |  | Sara | 3 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **008** | **As** user,  **I want** to be able to choose whether to repeat a rule several times by specifying a waiting time (in days, hours, minutes),  **So that** I can manage when to repeat the rules. |  | 5 |  |  |
| • **When** I have set the rule for once, **I want** the action to happen at that instant and then be deactivated.  • **When** I have set the rule for several times, **I want** the action to be re-executed after the waiting time has elapsed. | Add to the UI a checkbox to decide whether to repeat the rule, plus a graphic component to choose after how much to repeat it. |  | Elena | 2 |
| Update the attributes of the SingleRule class and its methods with: a boolean attribute for the checkbox and another to define when it should be repeated | Elena | 1 |
| Update Service to ensure that after the rule has been triggered, if it is to be repeated, it is re-initialized at the next time decided by the user. | Elena | 3 |
| Integration of the various tasks | Elena | 1 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **009** | **As** user,  **I want** to be able to write a specific string to the end of a specified file when the rule is activated,  **So that** I can update my files with new information. |  | 5 |  |  |
| • **When** I have specified the string to be written and the file, **I want** the string to be written at the end of that file once the trigger is verified. | Add to the UI the possibility of choosing the action of append to a file, with fields inherent to that action. |  | Viktor | 2 |
| Implement the controller methods related to the graphic components described above. | Viktor | 2 |
| Create the class ActionAppendFile and implement the executeAction() method. | Viktor | 2 |
| Create the method for writing to files in the class ActionAppendFile. | Viktor | 1 |
| Integration of the various tasks | Viktor | 1 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **011** | **As** user,  **I want** to be able to delete a file from a directory when the rule is activated,  **So that** I can maintain the efficiency of my storage space. |  | 3 |  |  |
| • **When** I choose the file to be deleted in a specific directory, **I want** the file to no longer be present in that directory once the trigger has occurred. | Add within the UI, in the drop-down menu of the actions, a field for choosing to delete a file. |  | Sara | 0,5 |
| Add the possibility of choosing all file types (not only mp3, wav, etc.) | Sara | 0,5 |
| Implement the onBtnDeleteFile method to select the file to be deleted. | Sara | 1 |
| Creation of a new class ActionDeleteFile with the necessary methods | Pasquale | 0,5 |
| Implementation of the ActionDeleteFile class with methods that check the selected file and display various messages according to the status of the file (e.g. deleted successfully, file already deleted, etc.). | Pasquale | 2 |
| Integration of the various tasks | Pasquale | 1 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **010** | **As** user,  **I want** to be able to copy or move a file from one directory to another when the rule is activated,  **so that** I can manage files according to my needs. |  | 5 |  |  |
| • **When** I have specified the file and destination directory to which it is to be copied, **I want** the file to be present on both the source and destination directories once the trigger has been verified.  • **When** I have specified the file and destination directory to which it is to be moved, **I want** the file to be present only in the destination directory once the trigger has been verified. | Adding within the UI the possibility of choosing whether to move a file from one directory to another |  | Pasquale | 1 |
| Adding within the UI the possibility of choosing whether to copy a file from one directory to another | Sara | 1 |
| Create the class ActionMoveFile and implement the executeAction() method, which allows a file to be moved from one directory to another | Elena | 2 |
| Create the ActionCopyFile class and implement the executeAction() method, which allows a file to be copied from one directory to another | Viktor | 2 |
| Integration of the various tasks | Sara | 1 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **TD1** | Repetition of code within the SetOfRules class. Specifically:  *FileManagement.saveRulesToFile(rules);* | - | 1 | Pasquale | 0,5 |
| **TD2** | Using the Observer pattern to optimise rule state change on file. | - | 3 | Sara | 3 |
| **B1** | Bug in the rule repetition in user story 008. | - | 1 | Elena | 1 |
| **B2** | Bug in the ActionAppendFile test class | - | 2 | Pasquale | 1 |

## sprint review

**User stories done**

The following user stories have been completed: 006, 007, 008, 009, 010, 011

**User stories added**

The user story 011 was added to the sprint

**User stories deleted**

None

**Technical debts**

* A code repetition within the SetOfRules class was resolved. Specifically:

*FileManagement.saveRulesToFile(rules);*

* The technical debt was solved to optimize the change of state of the rule on file, using the Observer pattern.

**Bugs**

The bug in the rule repetition in user story 008 has been fixed. As once the rule was reactivated, it displayed the dialogue box every 10 seconds.

**Sprint velocity**

Sprint velocity: 26 + 7 for bug and technical debts

## sprint RETROSPECTIVE

Immagine che contiene testo, schermata, Carattere, diagramma

Descrizione generata automaticamente

# **SPRINT 3**

## sprint backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **USER STORY/ACCEPTANCE CRITERIA** | **TASK** | **STORY POINTS** | **OWNER** | **PLANNED HOURS** |
| **012** | **As** user,  **I want** to be able to execute an external program with specified command line arguments,  **So that** I can automate external processes. |  | 13 |  |  |
| • **When** I choose an external program and specify the arguments to be passed, **I want** the program to be executed with these arguments when the trigger is triggered. | Updating the GUI with the necessary components |  | Sara | 2 |
| Create the ActionOpenExternalProgram class and implement the methods to open an external program with specified line arguments | Sara | 2 |
| Create the ActionOpenExternalProgramFactory class and implement the createAction method | Sara | 0,5 |
| Update the switch case in the createActionFactory method in the Controller class. | Sara | 0,5 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **013** | **As** user,  **I want** to be able to choose a specific day of the week or month or a specific date on which the action is to be performed,  **So that** the system can automatically check the day or date and activate the associated rule. |  | 3 |  |  |
| • **When** I specify the day of the week or month, **I want** the associated action to be executed when that day arrives.  • **When** I specify the date, **I want** the associated action to be executed when that date arrives.  • **When** I specify an invalid day or date, **I want** the associated action not to be executed. | Update the GUI with TriggerDate input fields and logic. |  | Elena | 1 |
| Create the TriggerDate class | Elena | 2 |
| Create the TriggerDateFactory class | Elena | 0,5 |
| Update the switch case in the createTriggerFactory method in the Controller class. | Elena | 0,5 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **014** | **As** user,  **I want** to be able to perform a specified action when I choose a file with a specified name in a specified directory,  **So that** I can automate file management |  | 5 |  |  |
| • **When** I specify a file in a directory, **I want** the system to perform the action only if that file exists. | Updating the GUI with the necessary trigger components |  | Viktor | 1 |
| Create the triggerExistingFile class (create the checkTrigger and getTrigger functions) | Viktor | 2 |
| Create the triggerExistingFileFactory class | Viktor | 0,5 |
| Update the switch case in the createTriggerFactory method in the Controller class. | Viktor | 0,5 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **015** | **As** user,  **I want** to be able to specify a file and a maximum size,  **So that** I can manage the storage space. |  | 5 |  |  |
| • **When** I select a file, **I want** the associated action to be executed if it exceeds a size limit. | Updating the GUI with the necessary trigger components |  | Viktor | 1 |
| Create the triggerSizeFile class | Viktor | 2 |
| Create the triggerSizeFileFactory class | Viktor | 0,5 |
| Update the switch case in the createTriggerFactory method in the Controller class. | Viktor | 0,5 |
| Creation of test classes for each task | Each member tests what they write | 1 |
| **TD3** | Re-implement the trigger and action classes using the pattern Factory method. |  | 8 |  |  |
|  | Re-implement the action classes using the pattern FactoryMethod |  | Sara | 4 |
| Re-implement the trigger classes using the pattern FactoryMethod | Elena | 4 |
| **TD4** | Replace the Service with the Thread class in Java |  | 8 | Pasquale | 4 |
| **TD5** | Replace the ObservableList implementing the pattern Observer |  | 13 | Pasquale | 6 |
| **B3** | Bug when adding a repeated action before a non-repeated action |  | 2 | Elena | 1 |

## sprint review

**User stories done**

The following user stories have been completed: 012, 013, 014, 015.

**User stories added**

None

**User stories deleted**

016

**Technical debts**

* Re-implement the trigger and action classes using the pattern Factory method, re-implementing the action and the trigger classes.

By employing the Factory Method, we aimed to avoid the pitfalls of extensive nested conditionals, which led our code to become difficult to read, understand, and maintain. With this modification, each factory method is responsible for creating a specific type of object, isolating the instantiation logic within its own scope.

* Replace the Service with the Thread class in Java. This decision was made prioritizing the portability of the application. Since Services are specific to JavaFX, we opted to replace them with a Java Thread as it offers a broader portability scope. This adjustment ensures compatibility across different environments and platforms, aligning with our goal of enhancing the application's versatility.
* Replace the ObservableList implementing the pattern Observer. The motivation behind this decision is to enhance the flexibility and maintainability of the codebase. By adopting the Observer pattern, we decouple the components in the system, making it easier to accommodate changes and updates. This modification provides a more modular and extensible solution compared to the direct use of ObservableList, aligning the code with design principles that promote separation of concerns and ease of future enhancements.

**Bugs**

We solved a bug in the Thread class where every time we executed a non-repeated rule after a repeated one, it wouldn't be executed.

**Sprint velocity**

Sprint velocity: 26 + 31 for bug and technical debts

## sprint RETROSPECTIVE

Immagine che contiene testo, schermata, Carattere, diagramma

Descrizione generata automaticamente