**Software Engineering Project Report**

****

Members:

Francisco Vasco - 61028

Iago Paulo - 60198

James Furtado - 61177

João Oliveira - 61052

Ricardo Gonçalo - 60519

Table of Contents

[Phase 1 Sprint 1 3](#_Toc121091020)

[Sprint backlog 3](#_Toc121091021)

[Burndown chart 3](#_Toc121091022)

[Produced content 5](#_Toc121091023)

[**Design Patterns** 5](#_Toc121091024)

[**Code Smells** 18](#_Toc121091025)

[Phase 2: 30](#_Toc121091026)

[Sprint 1 30](#_Toc121091027)

[**Sprint Backlog** 30](#_Toc121091028)

[**Burndown chart** 30](#_Toc121091029)

[**Produced content** 31](#_Toc121091030)

[Sprint 2 32](#_Toc121091031)

[**Sprint Backlog** 32](#_Toc121091032)

[**Burndown chart** 33](#_Toc121091033)

[**Produced content** 34](#_Toc121091034)

[Sprint 3 40](#_Toc121091035)

[**Sprint Backlog** 40](#_Toc121091036)

[**Burndown chart** 40](#_Toc121091037)

[**Produced content** 41](#_Toc121091038)

[Sprint 4 41](#_Toc121091039)

[**Sprint Backlog** 41](#_Toc121091040)

[**Burndown chart** 42](#_Toc121091041)

[**Produced content** 42](#_Toc121091042)

# **Phase 1 Sprint 1**

In this sprint we basically did the assignment of the first phase and the **scrum master** was João Oliveira.

## **Sprint backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| Todo | In Progress | Reviewing | Done |
|  |  |  | Identificar design patterns |
|  |  |  | Identificar code smells |
|  |  |  | Rever design patterns |
|  |  |  | Rever code smells |
|  |  |  | Corrigir eventuais erros |

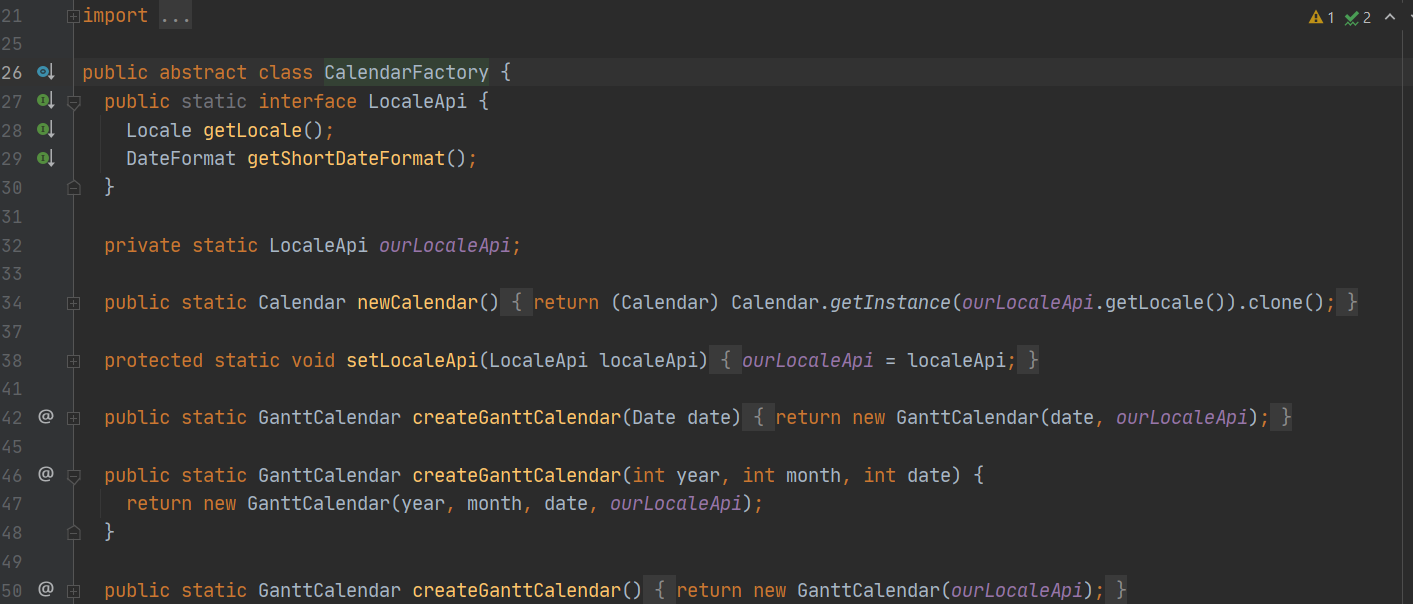
## **Burndown chart**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task ID** | **Task Description** | **Initial Estimate** | **15/Oct/22** | **16/Oct/22** | **17/Oct/22** | **18/Oct/22** | **19/Oct/22** | **20/Oct/22** | **21/Oct/22** |
| **Day 0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | Identificar design patterns | 1 | 0.5 | 0.5 |  |  |  |  |  |
| 2 | Identificar code smells | 1 |  | 0.5 | 0.5 |  |  |  |  |
| 3 | Rever design patterns | 1 |  |  |  | 1 |  |  |  |
| 4 | Rever code smells | 2 |  |  |  | 0.5 | 0.5 |  | 1 |
| 5 | Corrigir eventuais erros | 1 |  |  |  |  |  | 1 |  |
| **Completed Effort** | | 0 | 0.5 | 1 | 0.5 | 1.5 | 0.5 | 1 | 1 |
| **Remaining Effort** | | 6 | 5.5 | 4.5 | 4.0 | 2.5 | 2.0 | 1.0 | 0.0 |
| **Ideal Burndown** | | 6 | 5.1 | 4.3 | 3.4 | 2.6 | 1.7 | 0.9 | 0.0 |

## **Produced content**

### **Design Patterns**

#### **Abstract Factory Pattern**



**File:**biz.ganttproject.core\src\main\java\biz\ganttproject\core\time\CalendarFactory

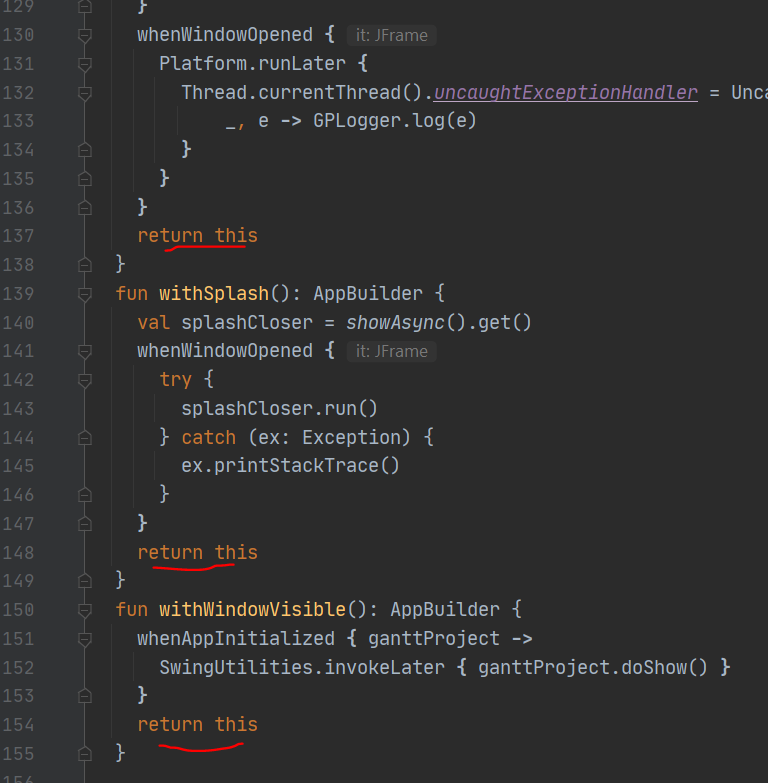
**Text:** Allows the creation of different GanttCalendars depending on the parameters used or a new Calendar instance.

**Author:** Iago Paulo

**Review:** Yes this is a factory pattern since the constructor was altered for the methods createGanttCalendar that receive a number of parameters and return a product **by** João Oliveira

#### **Builder Pattern**



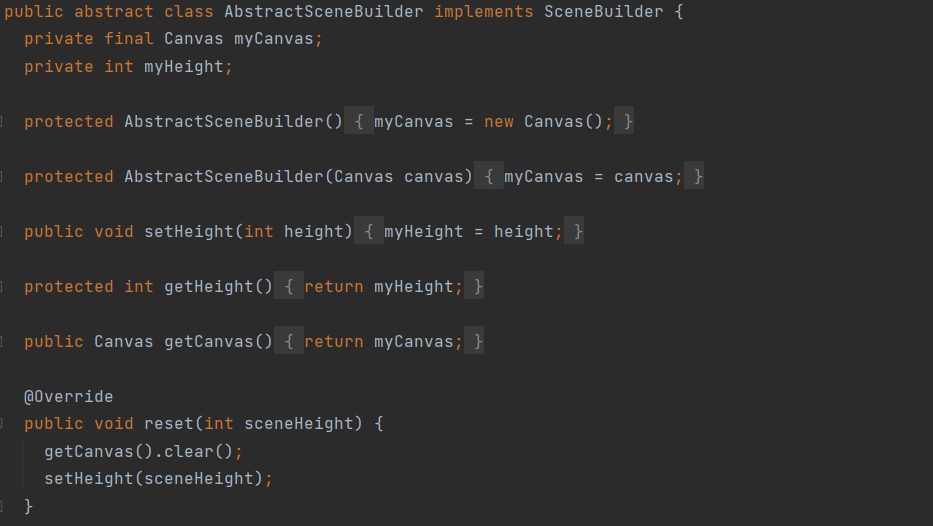
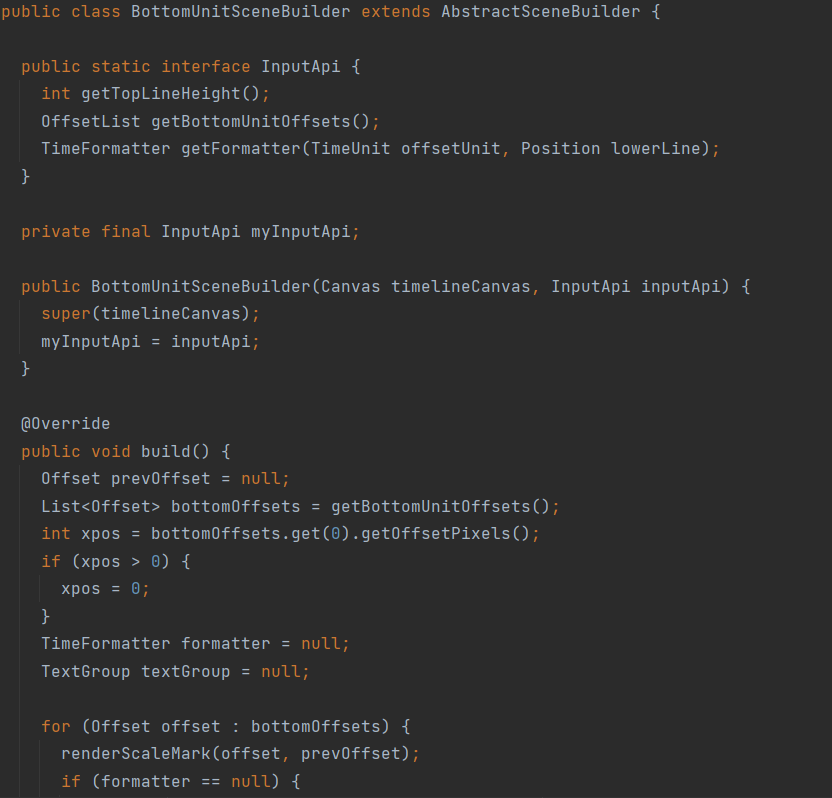


**File:**ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\App.kt

**Text:** This is a builder where each method returns the builder itself, allows step-by-step additions to the the app and finally launches the program.

**Author:** Iago Paulo

**Review:** Agreed, it provides the necessary methods to build the app **by** Ricardo Gonçalo



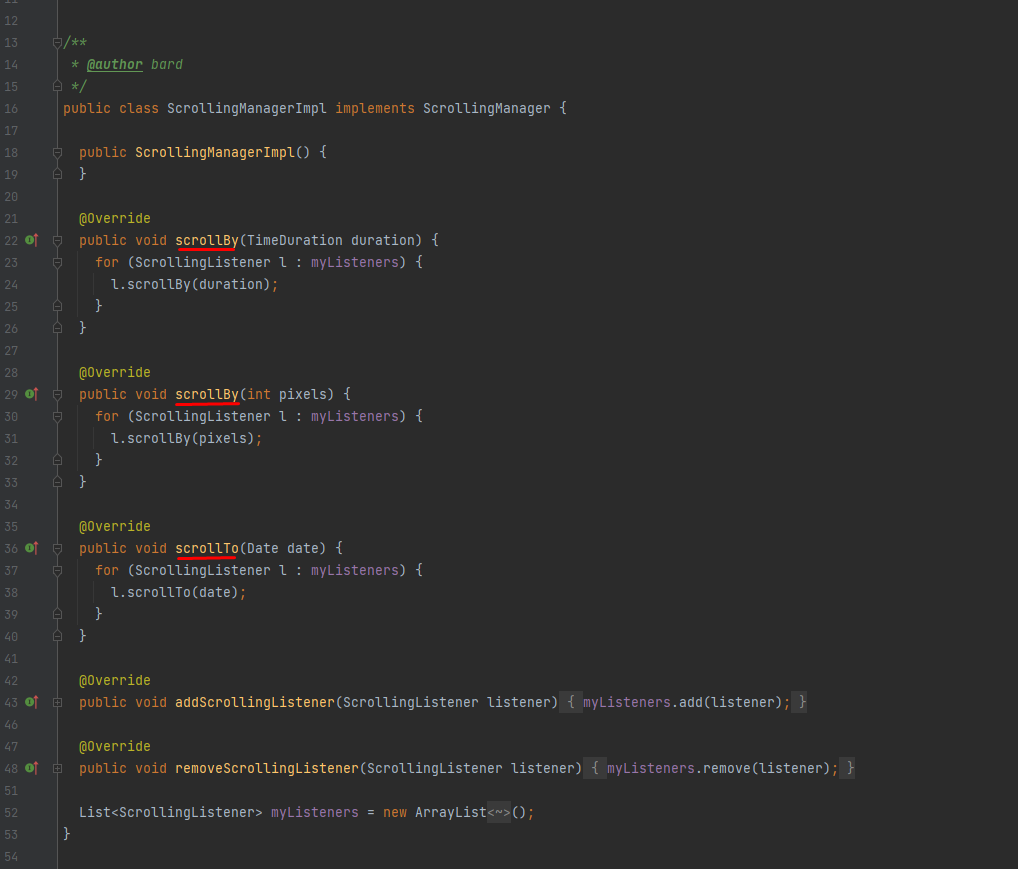
**File:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/chart/scene/AbstractSceneBuilder.java

**Text:** Build pattern in AbstractSceneBuilder and SceneBuilder,which are the foundation for classes like ChartRendererBase and BottomUnitSceneBuilder, builders for Canvas

**Author:** Ricardo Gonçalo

**Review:** Has the characteristics of a standard builder pattern and appears to be working as such **by** Iago Paulo

#### **Command Pattern**

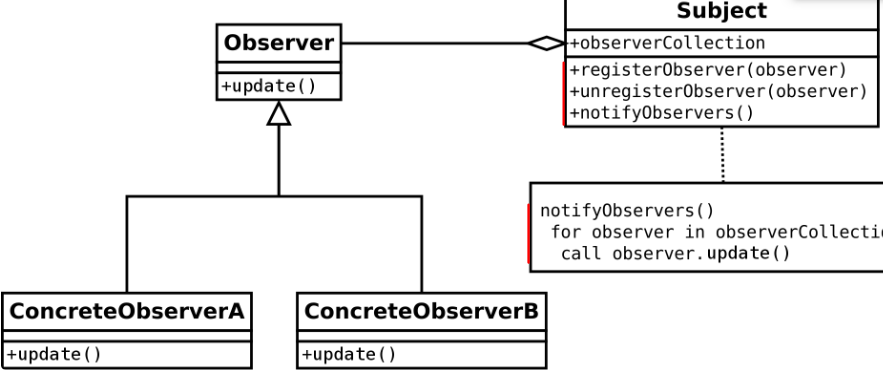


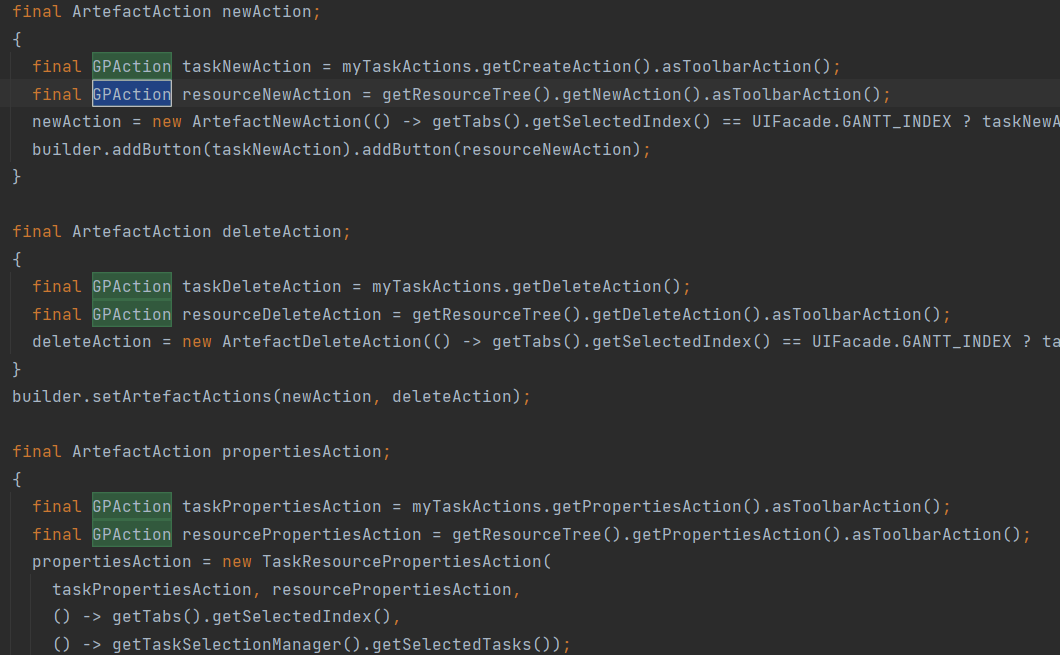
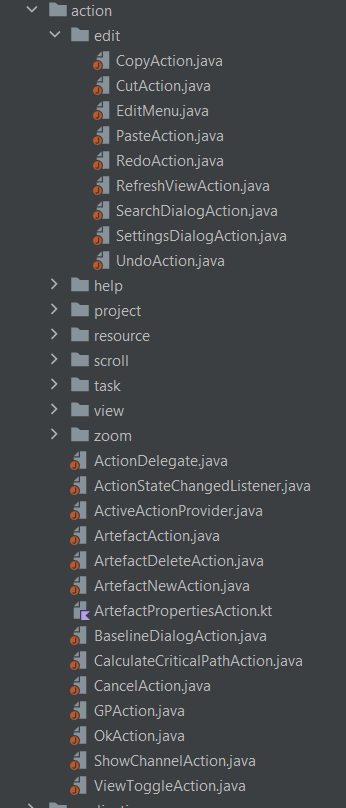
**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/gui/scrolling/ ScrollingManagerImpl.java

**Text:** The class Scrolling ManagerImpl serves as a command manager for “scrolling”.

**Author:** Francisco Vasco

**Review:** Francisco says it’s a Command Pattern although to me it looks more like an Observer Pattern due to the similarities to what we saw in the practical lessons:

 **by** Iago Paulo



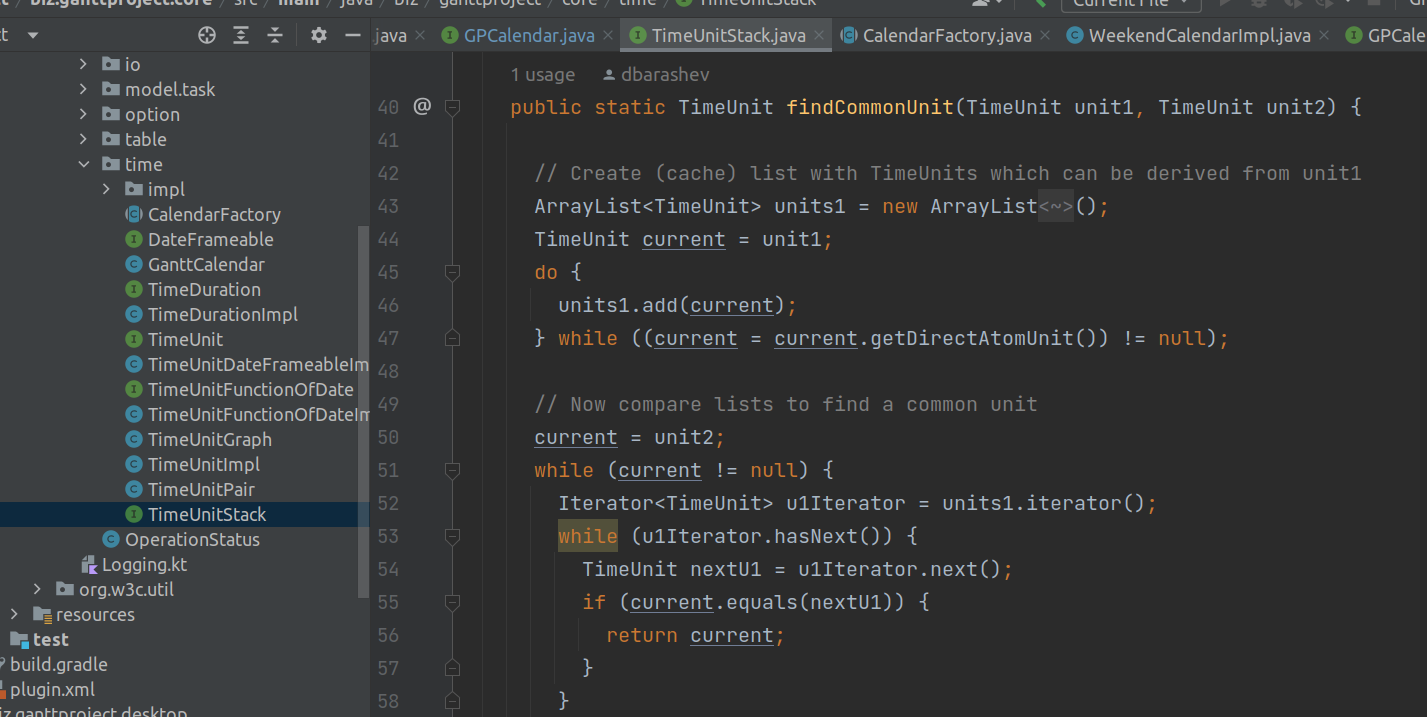
**File:** *ganttproject\src\main\java\net\sourceforge\ganttproject\action*

**Text:** This is a **Command** pattern, where the actions are classes instead of functions, verified for example in their usage in ganttproject/src/main/java/net/sourceforge/ganttproject/GanttProject.java

**Author:** Ricardo Gonçalo

**Review:**Several actions like “zooming”, “scrolling”, etc, are separated in classes, encapsulating the necessary information for executing each action **by** João Oliveira

#### **Iterator Pattern**



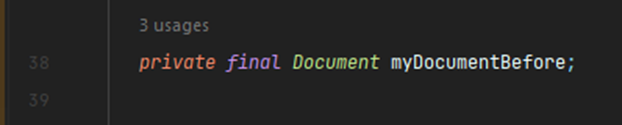
**File:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/time/TimeUnitStack.java

**Text:** The Util class is an inner class for the TimeUnitStack Interface and uses the Iterator Pattern on the static method findCommonUnit.

**Author:** James Furtado

**Review:** The class does in fact have an iterator and uses it to access elements, so it is an Iterator Pattern **by** Iago Paulo

#### **Memento Pattern**



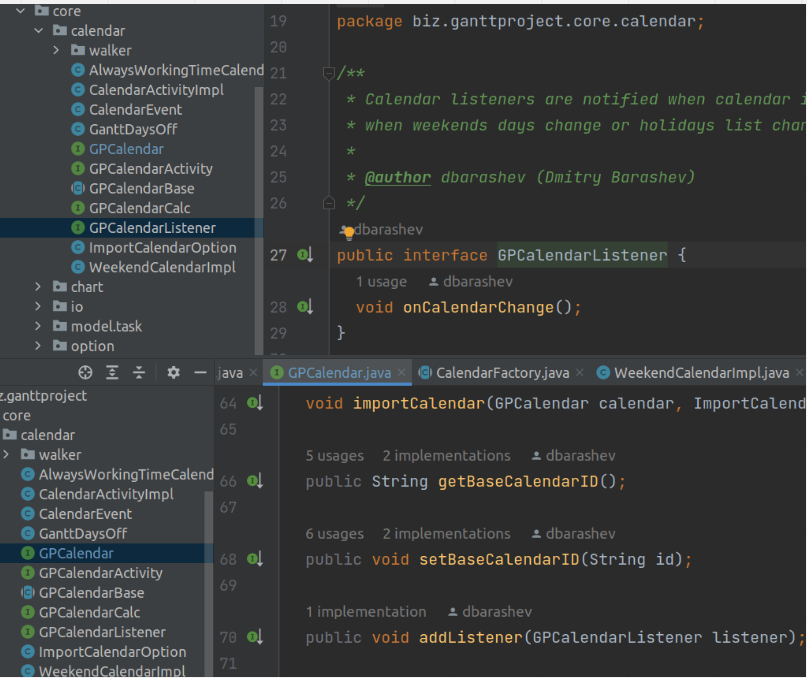
**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/undo/UndoableEditImpl.java

**Text:** Saves an object with the previous state of the document and has an undo() method that restores the document to it’s previous state.

**Author:** João Oliveira

**Review:** Correct, it is possible to restore the previous document state with the undo() method by keeping a snapshot of the previous state **by** Ricardo Gonçalo

#### **Observer Pattern**



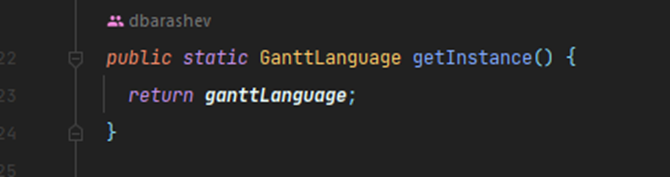
**File:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/

**Text:** In the biz.ganttproject.core.calendar there’s an observer pattern where the Subject interface is GPCalendar and the Observer is the GPCalendarListener.

**Author:** James Furtado

**Review:** GPCalendar implementations (the Subject) keep a collection of GPCalendarListener (added through the addListener method) and notify each one when changing the calendar, the pattern seems correct **by** Francisco Vasco

#### **Singleton**

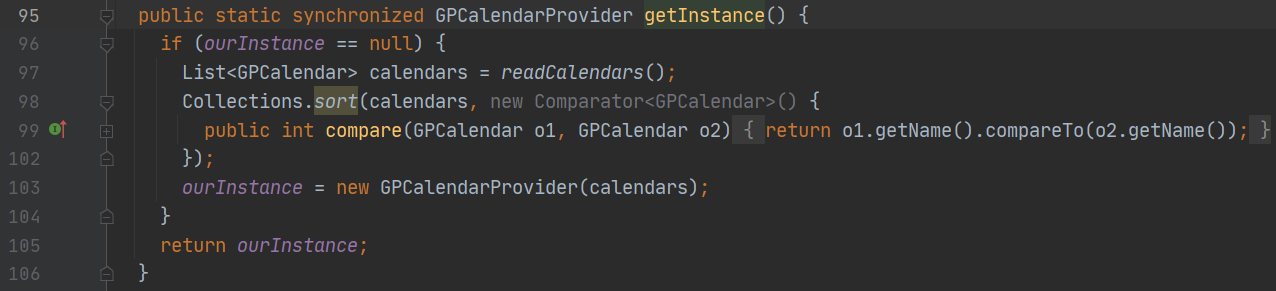


**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/parser/AbstractTagHandler.javaganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java

**Text:** Usage of the getInstance() method that returns a single object instance that is initialized only once through the usage of a static variable (line 79).

**Author:** João Oliveira

**Review:** Correct, standard Singleton Pattern implementation **by** Francisco Vasco

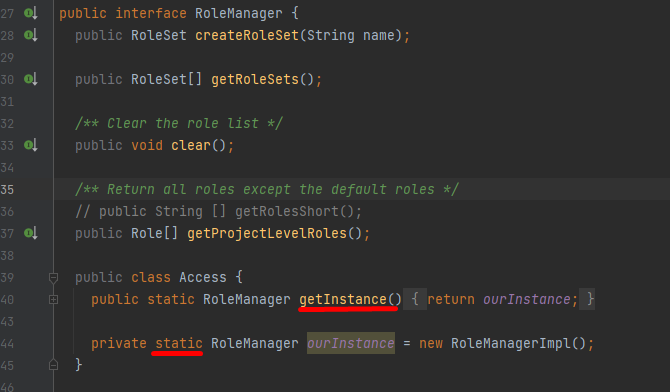


**File:**ganttproject\ganttproject\src\main\java\net\sourceforge\ganttproject\calendar\GPCalendarProvider.java

**Text:** Usage of the GetInstance method implemented as expected for a Singleton Pattern.

**Author:** Iago Paulo

**Review:** This pattern, using the ourInstance variable, does in fact assure only one GPCalendarProvider instance exists **by** Francisco Vasco

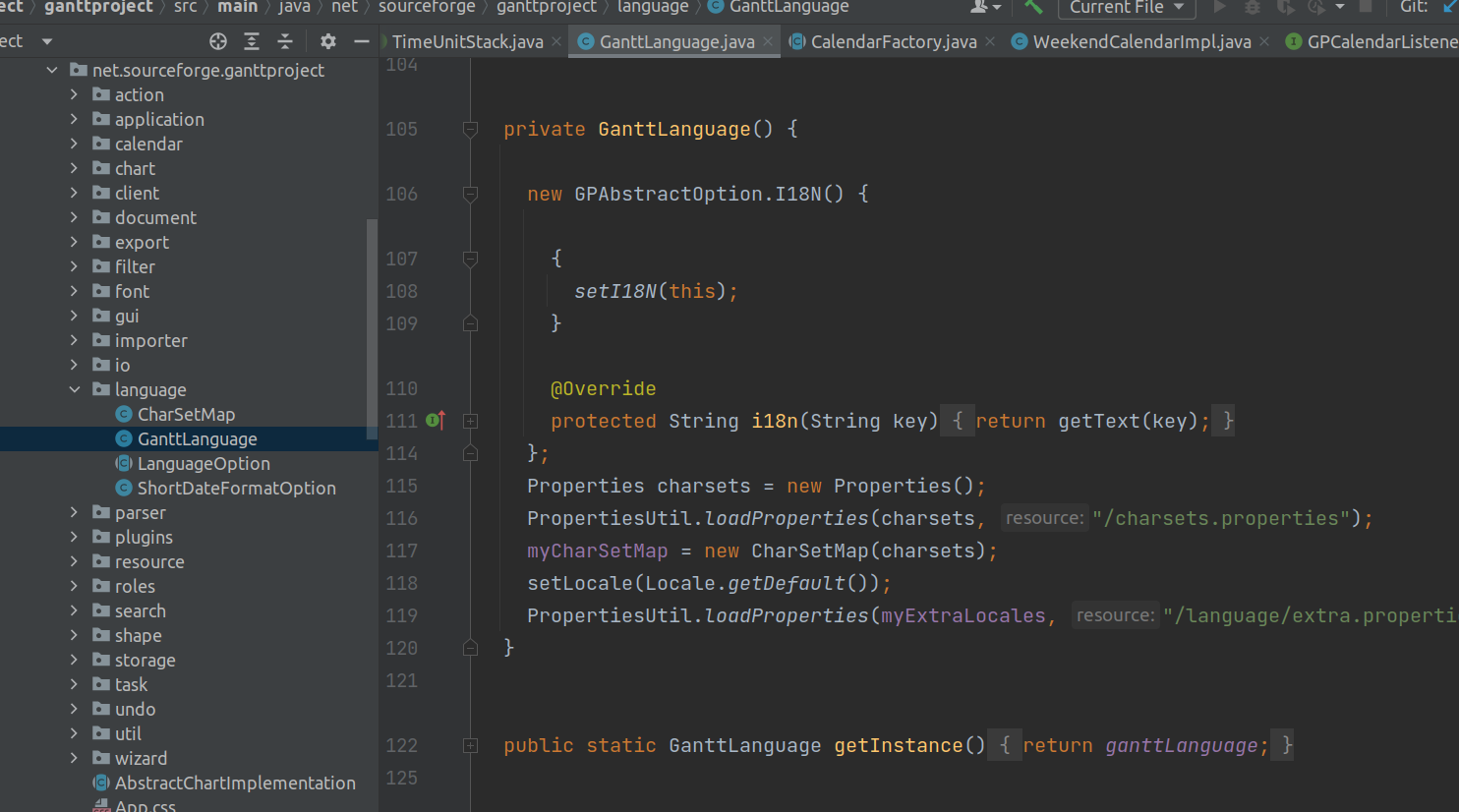


**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/roles/RoleManager.java

**Text:** The Access class on lines 39-45 assures there’s only one instance of RoleManager.

**Author:** Francisco Vasco

**Review:** The RoleManager’s Access inner class does use the Singleton Pattern **by** James Furtado



**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/language/GanttLanguage.java

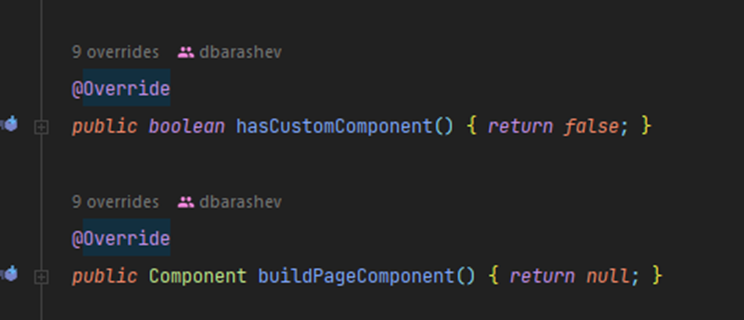
**Text:** The GanttLanguage class has a private constructor and a static method getInstance() that returns the single class instance.

**Author:** James Furtado

**Review:** It is in fact a singleton pattern **by** Ricardo Gonçalo

#### **Template method**

Text

Description automatically generated

**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/gui/options/OptionPageProviderBase.java

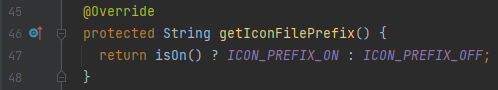
**Text:** Abstract class extended by 11 classes. 9 of them override the methods hasCustomComponent() and buildPageComponent()

**Author:** João Oliveira

**Review:** This is in fact a template method. The class is abstract and the subclasses override those and other methods **by** James Furtado

Text

Description automatically generated



Graphical user interface, text

Description automatically generated

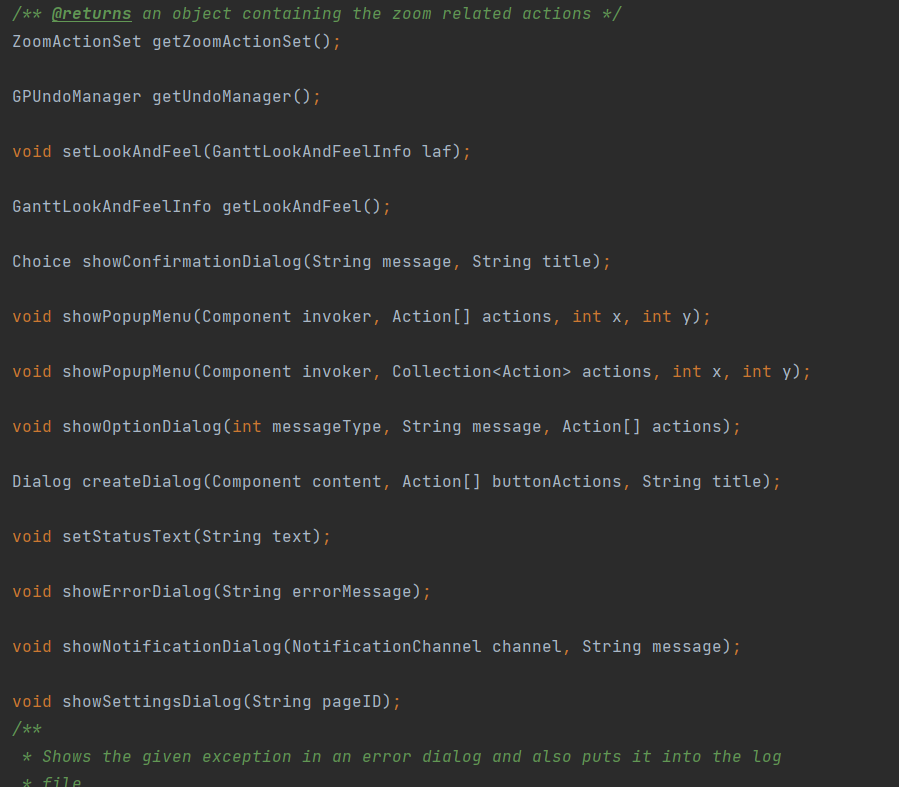
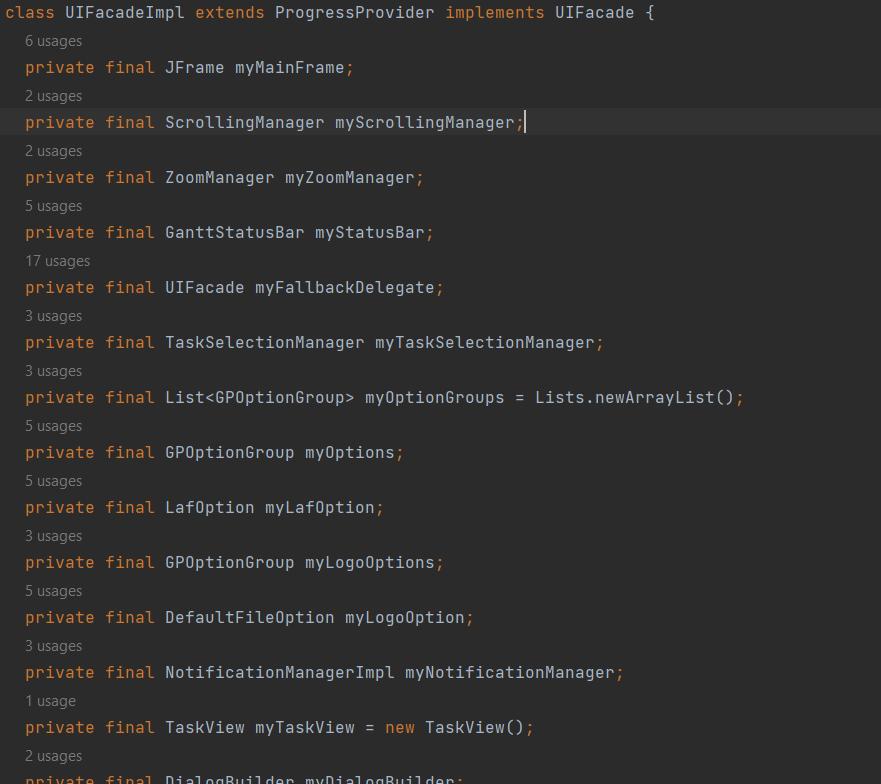
**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/action/GPAction.java (linhas 138-151 e 193-195)

**Text:** The final method createIcon(String iconSize) in the abstract class GPAction calls the getIconFilePrefix() method whose concrete implementation varies in the classes that extend GPAction.

**Author:** Francisco Vasco

**Review:** This is a behavioral pattern where in the class there’s a getIconFilePrefix() that is then changed by 19 classes that extend the class **by** João Oliveira

#### **Facade pattern**



**File:** ganttproject/src/main/java/net/sourceforge/ganttproject/UIFacadeImpl.java ganttproject/src/main/java/net/sourceforge/ganttproject/gui/UIFacade.java

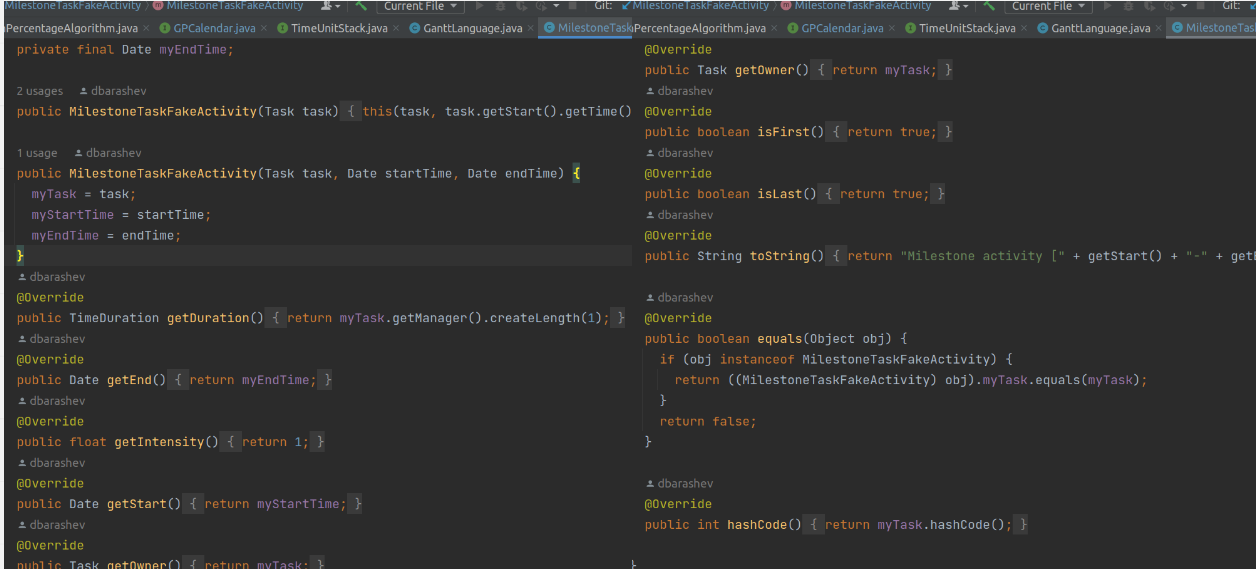
**Text:** Represents a facade template that tries to hide the UI’s complexity.

**Author:** Ricardo Gonçalo

**Review:** The class name is very descriptive, I agree with my colleague **by** James Furtado

### **Code Smells**

#### **Data Class**



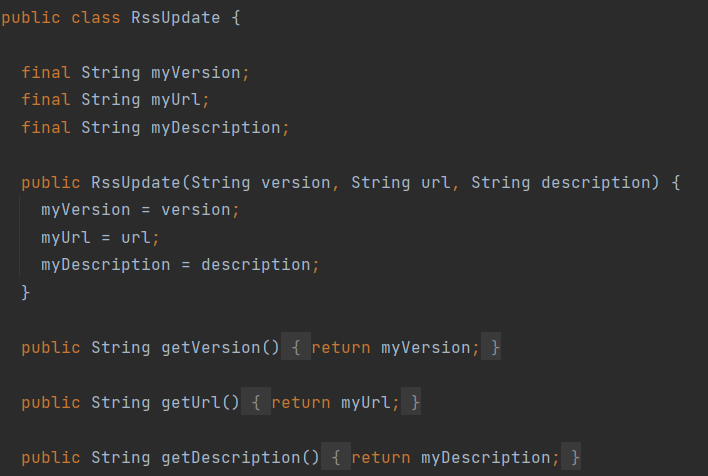
**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/chart/MilestoneTaskFakeActivity.java

**Text:** The MilestoneTaskFakeActivity only has getters.

**Solution:** Add some functionality to the class.

**Author:** James Furtado

**Review:** Yes the class only has getters. I agree with the solution, another one would be moving the variables to the classes that actually use them **by** Francisco Vasco



**File:** ganttproject/src/main/java/net/sourceforge/ganttproject/client/RssUpdate.java

**Text:** This is a Data Class, completely defined by the 3 primitives it has, it’s isolated existence adds unnecessary complexity.

**Solution:** This class could be nested in the RssParser, since it’s only used there.

**Author:** Ricardo Gonçalo

**Review:** Viable solution to reduce complexity **by** Iago Paulo

#### **Data Clump**



**File:**ganttproject-master\biz.ganttproject.core\src\main\java\biz\ganttproject\core\chart\canvas\DummySpatialIndex.java

**Text:** Long parameter list.

**Solution:** Create to group some of the arguments like x and y.

**Author:** Iago Paulo

**Review:** It is in fact a data clump **by** João Oliveira

#### **DeadCode**



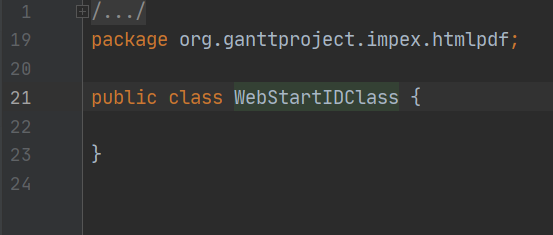
**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/filter/GanttXMLFileFilter.java

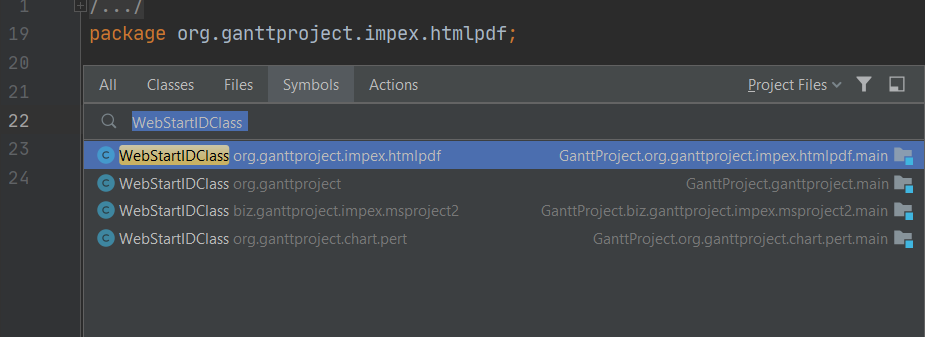
**Text:** The GanttXMLFileFilter class is never used.

**Solution:** Delete it.

**Author:** Francisco Vasco

**Review:** The GanttXMLFileFilter class is not being used so it can be safely deleted **by** João Oliveira





**File:**gantProj\ganttproject\biz.ganttproject.impex.msproject2\src\main\java\biz\ganttproject\impex\msproject2\WebStartIDClass

**Text:** The WebStartIDClass class is empty (all classes with this name are), could also be considered Specular Generality depending on the context.

**Solution:** Delete the classes or use them.

**Author:** Iago Paulo

**Review:** I agree it’s DeadCode, not used and could also be a situation of over-engineering anticipating future functionality **by** Francisco Vasco



**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/action/ArtefactAction.java

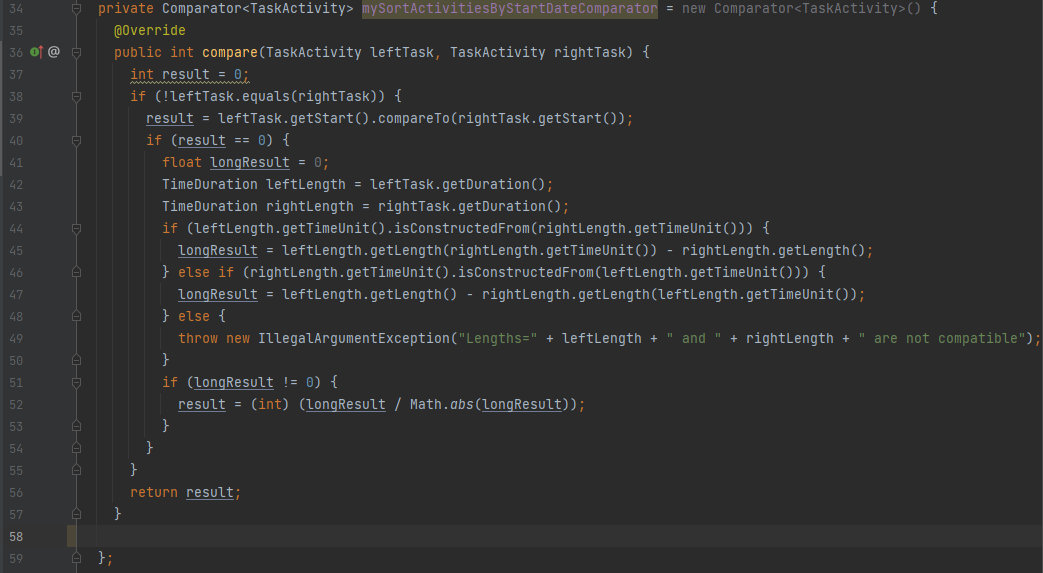
**Text:** The myDelegates (private final Action[] myDelegates) is initialized but never used.

**Solution:** It can be safely deleted.

**Author:** João Oliveira

**Review:** Yes, it's true that, being private, it isn't accessible from outside the scope and the only thing it could be doing is preventing some object from being taken by the garbage collector **by** Ricardo Gonçalo

#### **Duplicated Code**



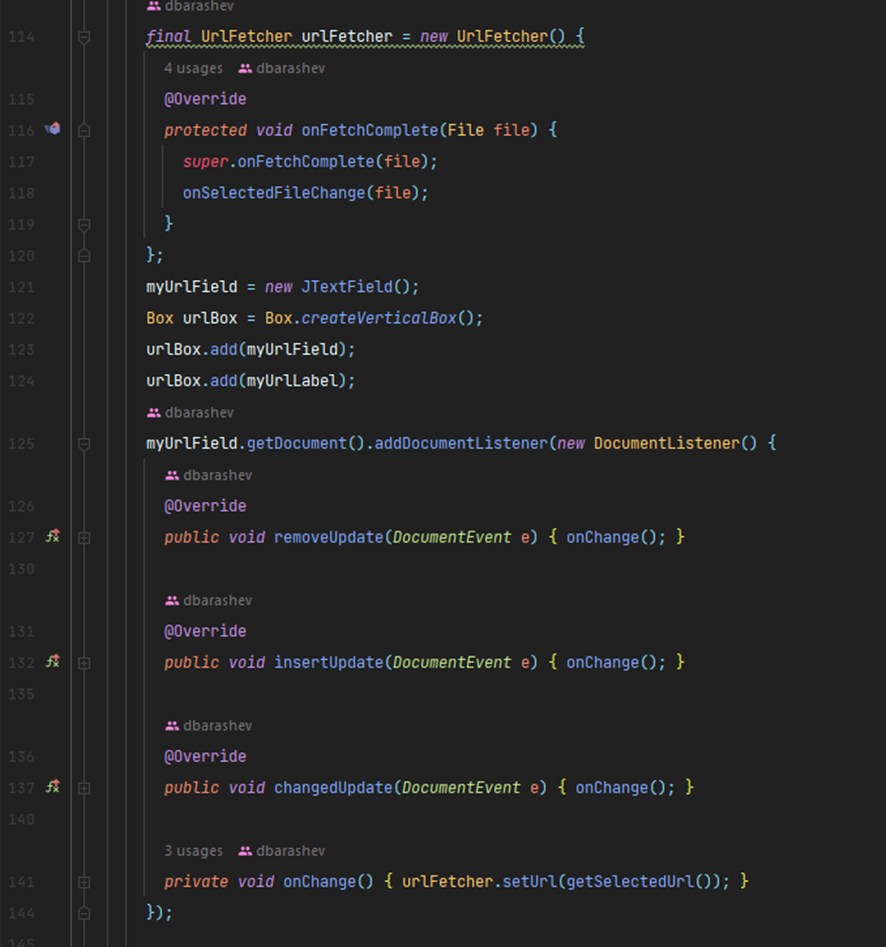
**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/task/algorithm/SortTasksAlgorithm.java

**Text:** On the SortTasksAlgorithm class the code on lines 44-53 is identical to the code on lines 71-80.

**Solution:** Extract part of the duplicated code to an auxiliar method.

**Author:** Francisco Vasco

**Review:** These 2 pieces of code are in fact identical, the solution seems appropriate **by** Iago Paulo



**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/gui/FileChooserPageBase.java e ganttproject/src/main/java/net/sourceforge/ganttproject/wizard/AbstractFileChooserPage.java

**Text:** Code on line 114 of the first class is duplicated in both classes.

**Solution:** The code could be extracted to a method or helper class

**Author:** João Oliveira

**Review:** The identification seems correct. Although this is just a line, errors could appear after future refactoring, this solution prevents that **by** Francisco Vasco

#### **Large class**



**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/GPTreeTableBase.java

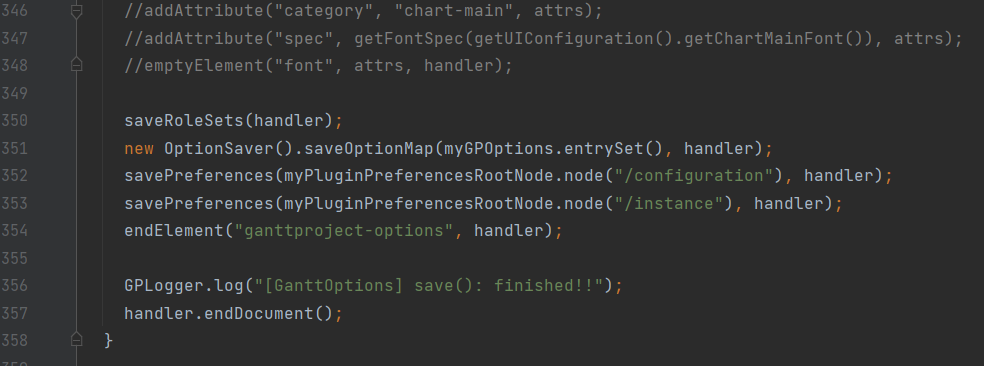
**Text:** The GPTreeTableBase class has more than 1000 code lines and has many other classes in it and long methods.

**Solution:** Dividir a class em outras classes mais pequenas.

**Author:** James Furtado

**Review:** Class has over 1200 lines and has 6 classes in it, it is in fact a long class. **by** Iago Paulo

#### **Long Method**



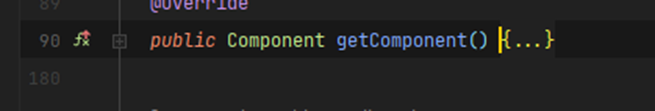
**File:** ganttproject/src/main/java/net/sourceforge/ganttproject/GanttOptions.java

**Text:** In the doSave method, lines 210-358, there’s a very large method with commented out code.

**Solution:** Simplify de method’s tasks in subtasks to make it more readable and understandable.

**Author:** Ricardo Gonçalo

**Review:**The method is too large, although it’s purpose is to initialize the UI. Auxiliar functions could have been used to clarify what the method does. The large sections of commented code should be deleted **by** João Oliveira



**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/gui/FileChooserPageBase.java

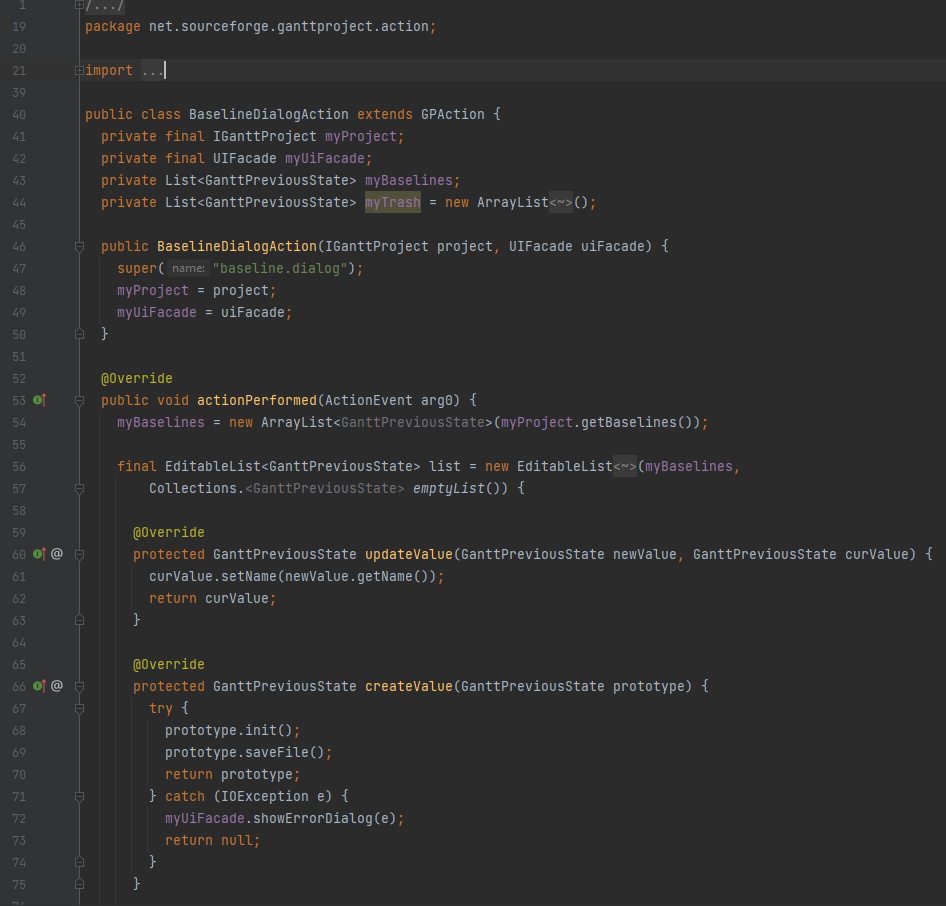
**Text:** The getComponent()is too large (90 lines).

**Solution:** Create auxiliar methods.

**Author:** João Oliveira

**Review:** I also think this could be considered a code smell **by** James Hertz

#### **No comment**



(a classe é grande por isso apresenta-se apenas uma parte aqui)

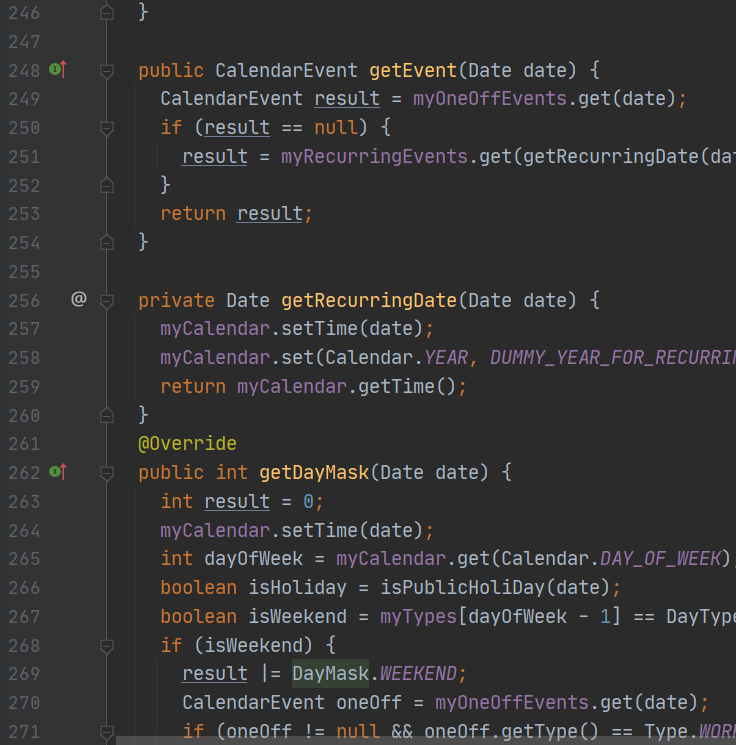
**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/action/BaselineDialogAction.java

**Solution:** Add comments.

**Text:** The BaselineDialogAction has no comments despite it’s complexity.

**Author:** Francisco Vasco

**Review:** The class has no comments so I agree it is a code smell **by** James Hertz



**File:**biz.ganttproject.core/src/main/java/biz/ganttproject/core/calendar/WeekendCalendarImpl.java

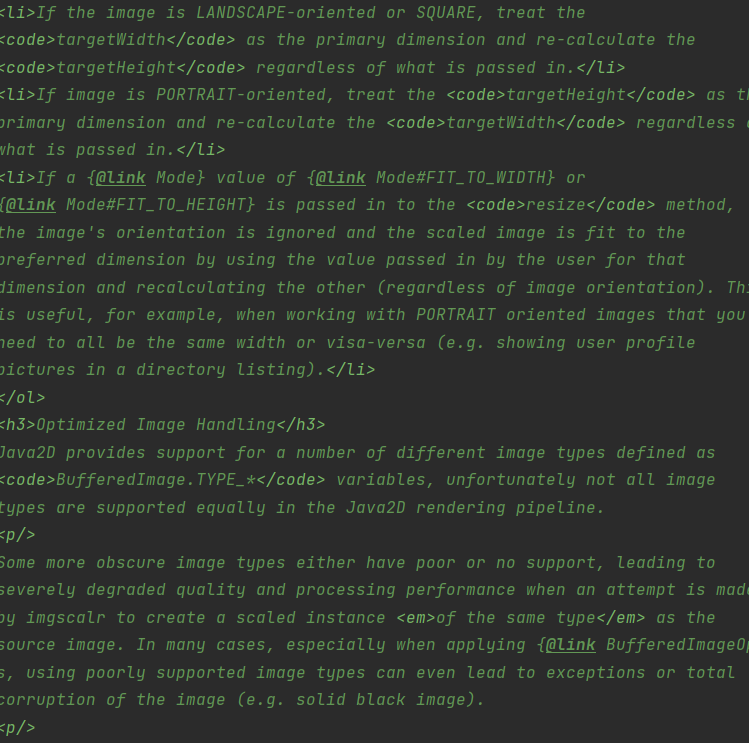
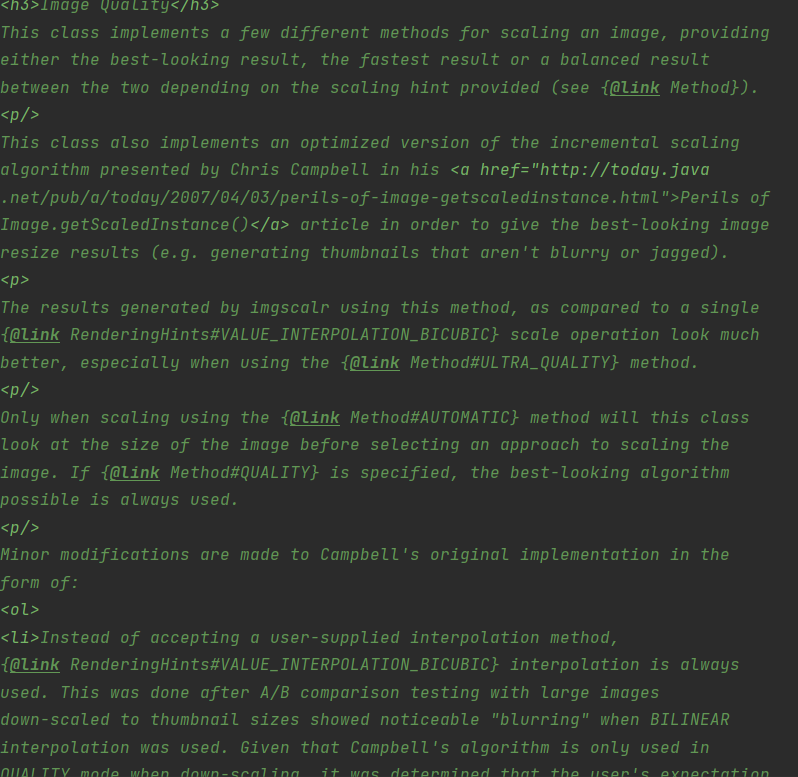
**Text:** This class has no comments.

**Solution:** Add comments.

**Author:** Iago Paulo

**Review:** Yes, for a class with 380 lines it has few comments **by** Ricardo Gonçalo

#### **Over comment**



**File:** ganttproject/src/main/java/org/imgscalr/Scalr.java

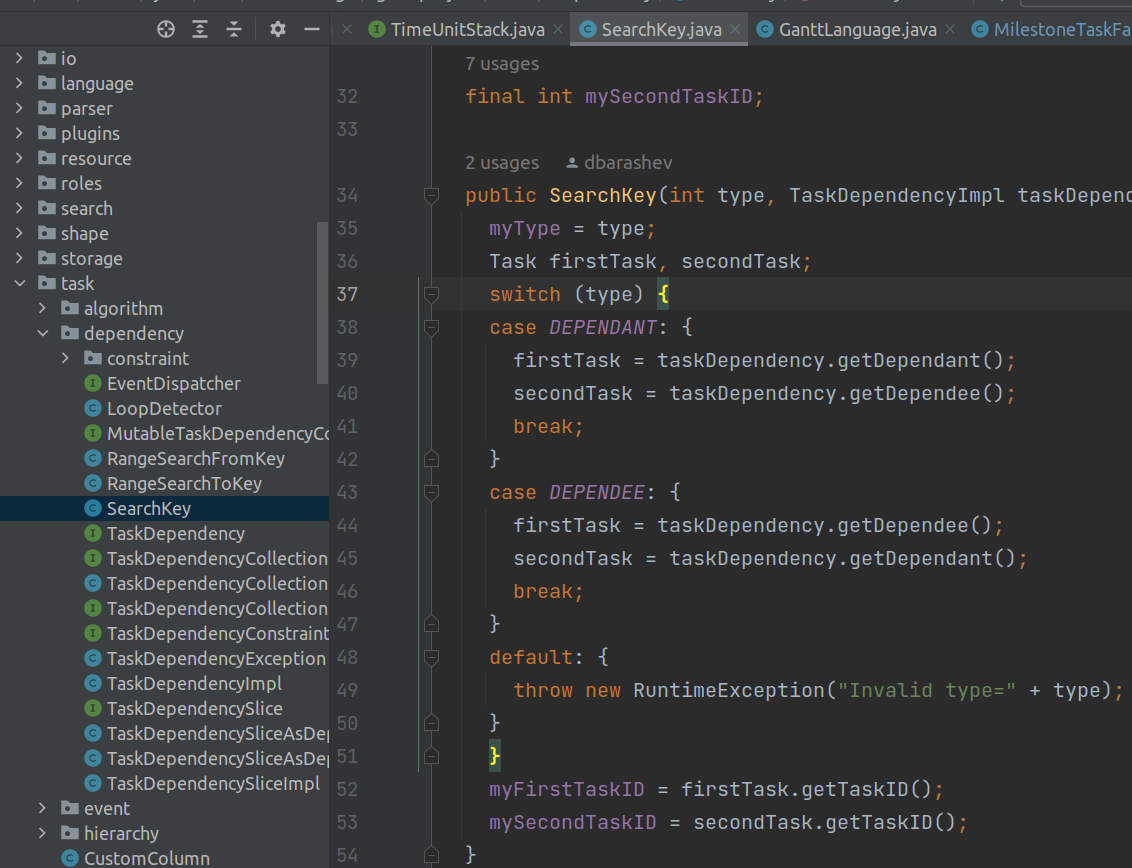
**Text:** The comments are too long. For example, the one right above the class has over 130 lines.

**Solution:**Simplify these comments, is there really a need for explanations this extensive for how the class works? Surely not, too much unnecessary information.

**Author:** Ricardo Gonçalo

**Review:** Unlike the rest of the code (which has almost no comments) this class has too many, so it’s clearly a code smell **by** James Furtado

#### **Switch statement**



**File:**ganttproject/src/main/java/net/sourceforge/ganttproject/task/dependency/SearchKey.java

**Text:** The class has an attribute myType which is an integer. The constants DEPENDEE and DEPENDANT are used as types. And it has a switch statement in the constructor for both cases and an exception otherwise.

**Solution:** Create two subclasses. Not only do we eliminate the myType attribute but we also eliminate the exception which is also good.

**Author:** James Furtado

**Review:** Yes, at least replace it with an Enum **by** Ricardo Gonçalo

# **Phase 2:**

## **Sprint 1**

During the sprint the main task was to identify two features that we wanted to add to the ganttproject. The **Scrum master** of this week was Ricardo Gonaçalo.

### **Sprint Backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| Todo | In Progress | Reviewing | Done |
|  |  |  | Sugerir 2 Features (Francisco) |
|  |  |  | Sugerir 2 Features (Iago) |
|  |  |  | Sugerir 2 Features (James) |
|  |  |  | Sugerir 2 Features (Joao) |
|  |  |  | Sugerir 2 Features (Ricardo) |
|  |  |  | Criar User Story Primeira Feature |
|  |  |  | Criar User Story Segunda Feature |
|  |  |  | Descrever Totalmente Primeita Feature |
|  |  |  | Descrever Totalmente Segunda Feature |

### **Burndown chart**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint 2 Burndown Chart | | | | | | | | | |
|  | | | | | | | | | |
| **Task ID** | **Task Description** | **Initial Estimate** | **7/Nov/22** | **8/Nov/22** | **9/Nov/22** | **10/Nov/22** | **11/Nov/22** | **11/Nov/22** | **13/Nov/22** |
| **Day 0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | Sugerir 2 Features (Francisco) | 1 | 1 |  |  |  |  |  |  |
| 2 | Sugerir 2 Features (Iago) | 1 | 1 |  |  |  |  |  |  |
| 3 | Sugerir 2 Features (James) | 1 | 1 |  |  |  |  |  |  |
| 4 | Sugerir 2 Features (Joao) | 1 | 1 |  |  |  |  |  |  |
| 5 | Sugerir 2 Features (Ricardo) | 1 | 1 |  |  |  |  |  |  |
| 6 | Criar User Story Primeira Feature | 1 |  | 1 |  |  |  |  |  |
| 7 | Criar User Story Segunda Feature | 1 |  | 1 |  |  |  |  |  |
| 8 | Descrever Totalmente Primeita Feature | 2 |  |  | 1 | 1 |  |  |  |
| 9 | Descrever Totalmente Segunda Feature | 2 |  |  | 1 | 1 |  |  |  |
| **Completed Effort** | | 0 | 5 | 2 | 2 | 2 | 0 | 0 | 0 |
| **Remaining Effort** | | 11 | 6.0 | 4.0 | 2.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **Ideal Burndown** | | 11 | 9.4 | 7.9 | 6.3 | 4.7 | 3.1 | 1.6 | 0.0 |

### **Produced content**

After a bit of discussion in the end we choose these two features. We list below the features and their respective user story.

**Feature 1:**

**Description:**

When a file is loaded into the application, there will be an option for the user to extend the duration of tasks that were supposed to end in the past to end during the current day, also rearranging tasks according to their dependencies. Additionally, there will be a button that will also have this feature.

It should also (optionally) automatically make the end date of a task supposed to end in the future to the current one, when it is marked as finished.

**User Story: As** a project manager, **I want to** see the task duration changing dynamically in case a task takes more or less time than anticipated **so that** I may have a more accurate project duration time.

**Feature 2:**

**Description:** Statistics revealing the state of the project regarding how complete it is.

**User Story:** **As a** user, **I want to** easily see the project completion state, **so that** I can easily know if the project is evolving as expected.

## **Sprint 2**

In this sprint we produced the use cases diagrams asked for the whole project. The **Scrum master** of this week was Francisco Vasco.

### **Sprint Backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| Todo | In Progress | Reviewing | Done |
|  |  |  | Find/discribe related use cases (Francisco) |
|  |  |  | Find/discribe related use cases (Iago) |
|  |  |  | Find/discribe related use cases (James) |
|  |  |  | Find/discribe related use cases (Joaoa) |
|  |  |  | Find/discribe related use cases (Ricardo) |
|  |  |  | Develop an use case sub-diagram (Francisco) |
|  |  |  | Develop an use case sub-diagram (Iago) |
|  |  |  | Develop an use case sub-diagram (James) |
|  |  |  | Develop an use case sub-diagram (Joao) |
|  |  |  | Develop an use case sub-diagram (Ricardo) |

### **Burndown chart**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint 3 Burndown Chart | | | | | | | | | |
|  | | | | | | | | | |
| **Task ID** | **Task Description** | **Initial Estimate** | **14/Nov/22** | **15/Nov/22** | **16/Nov/22** | **17/Nov/22** | **18/Nov/22** | **19/Nov/22** | **20/Nov/22** |
| **Day 0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | Find/discribe related use cases (Francisco) | 2 |  |  | 1 | 1 |  |  |  |
| 2 | Find/discribe related use cases (Iago) | 2 |  | 1 | 1 |  |  |  |  |
| 3 | Find/discribe related use cases (James) | 2 |  | 1 | 1 |  |  |  |  |
| 4 | Find/discribe related use cases (Joaoa) | 2 |  |  | 2 |  |  |  |  |
| 5 | Find/discribe related use cases (Ricardo) | 2 |  |  | 1 | 1 |  |  |  |
| 6 | Develop an use case sub-diagram (Francisco) | 1 |  |  |  | 1 |  |  |  |
| 7 | Develop an use case sub-diagram (Iago) | 1 |  |  | 1 |  |  |  |  |
| 8 | Develop an use case sub-diagram (James) | 1 |  |  | 1 |  |  |  |  |
| 9 | Develop an use case sub-diagram (Joao) | 1 |  |  |  | 1 |  |  |  |
| 10 | Develop an use case sub-diagram (Ricardo) | 1 |  |  |  | 1 |  |  |  |
| **Completed Effort** | | 0 | 0 | 2 | 8 | 5 | 0 | 0 | 0 |
| **Remaining Effort** | | 15 | 15.0 | 13.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **Ideal Burndown** | | 15 | 12.9 | 10.7 | 8.6 | 6.4 | 4.3 | 2.1 | 0.0 |

### **Produced content**

Here below are the diagram produced along with its author and the reviewer comments.

**Task manager**

Diagram

Description automatically generated

**Author:** Francisco Vasco

**Description:**

Name: Create task

Description: Create a new task on the chart

Actors: User

Name: Edit task

Description: Edit an existing task's properties

Actors: User

Name: Move up or down

Description: Moves the task on the chart, relative to other tasks

Actors: User

Name: Create dependency

Description: creates a dependency between 2 tasks

Actors: User

Name: Make child of another task

Description: the selected task becomes a requirement for another task

Actors: User

**Review:**  The overall idea and the manipulation of the task was well captured **by** Ricardo Gonçalo.

**Properties manager**

**Diagram

Description automatically generated**

**Author:** Iago Paulo

**Description:**

The only actor is the user, which can do all this use cases.

Name: Change project info

desc: User can change/add things like the name of the project, Organization, Web Link and description.

Name: Add Roles

desc: User can add custom roles that are not on the default roles yet.

Name: Remove Roles

desc: User can remove those custom roles.

Name: Manage days off

desc: User can use this to set the days that he will have a break.

Name: Add custom day-off

desc: User can add a specific day off, no holiday and no weekend day, just an arbitrary day off.

Name: Choose weekend days

desc: User can set the weekend days instead of the default Saturday and Sunday

(Example: Some companies also work on Saturday, so Sunday is the only Weekend Day)

Name: Choose holiday calendar

desc: User can choose his holidays on this calendar from several countries, making it easy to put the custom days-off from each country.

**Review:** He diagram seems to encapsulate all the functionalities associated with the project properties. The usage of the «extends» keyword between "Manages days off" and its related use cases seems correct as it is not required to use any of these use cases, they are optional **by** Francisco Vasco.

**Project manager**

Diagram

Description automatically generated

**Author:** James Furtado

**Description:**

Name: Create new Project

Description: The actor can create a new project that will hold a set of tasks. The actor will have to give some information in order to do so, such as the project name.

Actors: User

Name: Save Project

Description: The actor can save the project (with all its tasks) in a file that can later be opened.

Actors: User

Name: Open Project

Description: The actor selects a project (from a file or an url - webserver). The system closes the current opened project (if any) and loads the selected project's (tasks, resources, etc...) so the project can be edited.

Actors: User

Name: Export

Description: The actor can export (save to a file) the project to a format other the default ganttproject.

Actors: User

Name: Import

Description: The actor chooses a file (or an url) that was previously export as a project and the system adds the task (resources, etc...) of such project to the current opened project.

Actors: User

**Review:** Everything is correct. Contains the name, the description and the actor **by** João Oliveira.

**Timeline manager**

Diagram

Description automatically generated

**Author:** João Oliveira

**Description:**

Use case referring to the timeline manager.

Has two actors, one primary and one secondary.

The primary is the user that interacts with the system

The secondary is the time.

Name: Do Zoom

Description: Action that allows the user to zoom in and out.

Name: Navigate to specific day

Description: The user can navigate in the timeline to select the date range he want to see. Can be done by dragging the timeline or by selecting some options, like "Past", "Future", "Project Start", "Today", "Project End" or "Selection".

Name: See critical path:

Description: The user can choose an option to see the critical path.

Name: Change current day

Description: The secondary actor, the time, updates the day to the current one.

**Review:** I agree, but we can also say that the Time updates de current day \*every day\*, although it is self-explanatory **by** Iago Paulo

**Timeline manager**

Diagram

Description automatically generated

**Author:** Ricardo Gonçalo

**Description:**

Name:Resource Manager

desc: The actor responsible for the management of the human resources, **being the principal actor of all the actions.**

Name: Edit Resource:

desc: User can remove those custom roles. Which is inherited by: Add Resource and extended by: Set Personal Info, Set Role, Set Standard Rate and Edit Days Off.

Name: Delete resource:

desc: The user can delete a resource from the system

Name: Add Resource:

desc: The user can add a resource to the system, setting its values (Edit Resource)

Name: Set Personal Info:

desc: The user can set the name, phone number and mail of the Resource

Name: Set Role

desc: The user can set the role of the resource, choosing from available values

Name: Set Standart (standard) Rate:

desc: The user can set the standart rate of the resource

Name: Edit Days Off:

desc: The user can add or remove intervals of days off

Name: Send Email:

desc: The user can send an email to the resource

Name: Manage columns:

desc: The user can show, hide, create columns, delete created columns and set values in the columns of the Resources Chart.

**Review:**

When looking at the use case diagram I agree with what was done regarding

the use case "Delete Resource" and "Send email".

Regarding "Manage columns" I don't think it's even a use case, I think that the column is a way

to access others, use cases.

My major critic is regarding the use case "Edit resource" which seems to be an empty

use case where everything that makes the use case extends it.

What I think is that probably it should've been inherited by the uses cases that extends it (since these use cases have one thing in common - they change some properties of a resource). I also think that it's a bit weird to make the use case add resource inherit from "Edit resource"

**by** James Furtado.

## **Sprint 3**

In this sprint we spent some time exploring the ganttproject and trying to fix some bugs. The **Scrum master** of this week was Iago Paulo.

### **Sprint Backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| Todo | In Progress | Reviewing | Done |
|  |  |  | Debugs (Iago) |
|  |  |  | Correct bugs (Iago) |
|  |  |  | Analyse how to implement features 1 (Ricardo) |
|  |  |  | Analyse how to implement features 2 (Iago,James,Joao,Francisco) |
|  |  |  | Make java program to encapsulate tasks (Iago) |

### **Burndown chart**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint 4 Burndown Chart | | | | | | | | | |
|  | | | | | | | | | |
| **Task ID** | **Task Description** | **Initial Estimate** | **21/Nov/22** | **22/Nov/22** | **23/Nov/22** | **24/Nov/22** | **25/Nov/22** | **26/Nov/22** | **27/Nov/22** |
| **Day 0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | Debug the code to know how it works | 3 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | Find and correct possible bugs (It had one) | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | See how to implement feature 1 (Ricardo) and 2 (James,Iago,Francisco,Joao) | 2 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 4 | Create a java program to encapsulate tasks so James can use in the ganttproject | 1 | 0 | 0 | 0 | 0 | 0 | 0.5 | 0.5 |
| **Completed Effort** | | 0 | 1 | 1 | 2 | 1 | 1 | 0.5 | 0.5 |
| **Remaining Effort** | | 7 | 6.0 | 5.0 | 3.0 | 2.0 | 1.0 | 0.5 | 0.0 |
| **Ideal Burndown** | | 7 | 6.0 | 5.0 | 4.0 | 3.0 | 2.0 | 1.0 | 0.0 |

### **Produced content**

We didn’t do nothing more than exploring the code and think discussing about it. During this our dear college **Iago Paulo** found a bug and fixed it. Whenever we moved a task a count that is supposed to increase only when we create a new task would increase so if we moved a task **x** times the next task would appear with a counter of the + (**x + 1)** of the last task created.

## **Sprint 4**

In this sprint we implemented the features, the unit test for the features and we did the metrics report.

### **Sprint Backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| Todo | In Progress | Reviewing | Done |
|  |  |  | Implement feature 1 (Ricardo) |
|  |  |  | Implement feature 2 (James, João, Francisco, Iago) |
|  |  |  | Upload sprints to github (Todo agente) |
|  |  |  | Take metrics (Todo agente) |

### **Burndown chart**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint 4 Burndown Chart | | | | | | | | | |
|  | | | | | | | | | |
| **Task ID** | **Task Description** | **Initial Estimate** | **28/Nov/22** | **29/Nov/22** | **30/Nov/22** | **1/Dec/22** | **2/Dec/22** | **3/Dec/22** | **4/Dec/22** |
| **Day 0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | Implement feature 1 (Ricardo) | 5 | 2 | 2 | 1 |  |  |  |  |
| 2 | Implement feature 2 (James, João, Francisco, Iago) | 5 | 1 | 2 | 2 |  |  |  |  |
| 3 | Upload sprints to github (Todo agente) | 2 |  |  |  | 0.5 | 0.5 | 0.5 | 0.5 |
| 4 | Take metrics (Toda agente) | 1 |  |  |  |  | 0.25 | 0.5 | 0.25 |
| **Completed Effort** | | 0 | 3 | 4 | 3 | 0.5 | 0.75 | 1 | 0.75 |
| **Remaining Effort** | | 13 | 10.0 | 6.0 | 3.0 | 2.5 | 1.8 | 0.8 | 0.0 |
| **Ideal Burndown** | | 13 | 11.1 | 9.3 | 7.4 | 5.6 | 3.7 | 1.9 | 0.0 |

### **Produced content**

Each one of us did a report about the metrics that is in the folder Phase2/Sprint4/<member\_name>. Since it’s too big and it would mess up with our document format, we decided not to include such reports in this final report. We will only include the reviews the name of the metrics each one of us chose to do.

**Metric:** Mood Metrics Review

**Responsible:** Francisco Vasco

**Review:** This evaluation makes it seem like the project is acceptable in the metrics regarded, being the worst score by far the 75.5% in the AIF.

This doesn’t really make the project harder to understand, harder to change or any less performant **by** Ricardo Goncalo.

**Metric:** Line of code

**Responsible:** Iago Paulo

**Review:** The conclusions Iago has reached from the collected metrics agree with the consensus of the team over the development period of the project,

that is the project has a severe lack of comments which makes it more challenging to understand and improve **by** Francisco Vasco.

**Metric:** Dependency

**Responsible:** James Furtado

**Review:** There are classes that are really accoupled and this makes it hard to extend the project with new features. (Which means that we had some troubles trying to implement the new features) **by** Iago Paulo.

**Metric:** Chidamber and Kemerer

**Responsible:** João Oliveira

**Review:** I could see that he did some research and I agree with him that there is some values that are a bit high **by** James Furtado.

**Metric:** Complexity

**Responsible:** Ricardo Gonçalo

**Review:** The report is complete. Contains the average, sum, maximum and minimum.

Each metric is well explained **by** João Oliveira.