# <ASSESSMENT>

# LIST OF PROBLEMS

VERSION <1.0>

**Ιορδάνης-Ραφαήλ Φυδανάκης AM:420**

**Ευάγγελος Δημουλής  AM:421**

–

# TABLE OF CONTENTS

# INTRODUCTION

The objectives of this phase is to report a list of architecture, design and src style problems. After identifying and listing the problems, we provide a preliminary plan of actions towards re-engineering the legacy code.

# ARCHITECTURE & DESIGN LEVEL

## ARCHITECTURE

We found two main problems at the Package level of the project.

First, there is a lack of back-end "server" package that hosts the business logic and as a result the gui.MainEngine package has dependencies to all other packages which the GUI depends on and increases the coupling. The back-end "server" package would play the role of an intermediate between the DataKeeper and the GUI populated with "load" methods, which are currently in the GUI package.

Furthermore, in the GUI there are methods that do some useful processing on data and we need to move the logic to the DataKeeper i.e. "Move behavior close to data" pattern. This will result in converting the Data class to something like a "service provider". The DataKeeper will become a "God" class that way. So as a next step we will divide that class into smaller subsystems that delegate work to other classes "Facade" pattern.

Packages where problematic behavior has been observed:
(1) gui.MainEngine
(2) tablelustering.clusterExtractor.common
(3) data.dataPPL.pplSQLSChema
(4) data.dataKeeper

# DETAILED DESIGN

At the class level of the project we identified problems  mainly at three packages.

1. At the Data package the GlobalDataKeeper makes use of the  Worker class to create the data and then copies it to his own class attributes (unnecessary step). We will remove these attributes from GlobalDataKeeper.

2. At the Gui package the Gui class acts as a "God" class, it has many dependencies inside the Gui package and also outside of it, it is composed of very long methods and contains too many attributes. As a result we will work on refactoring this class so that it is populated only with the front-end code logic. The PldConstruction class could be implemented by 3 Abstract classes each for a version of getNumOfAttributesOfNextSchema() method and the shared attributes of these classes. Also the getSegmentSize() private method is the same for every class, It could be created in these 3 abstract classes.

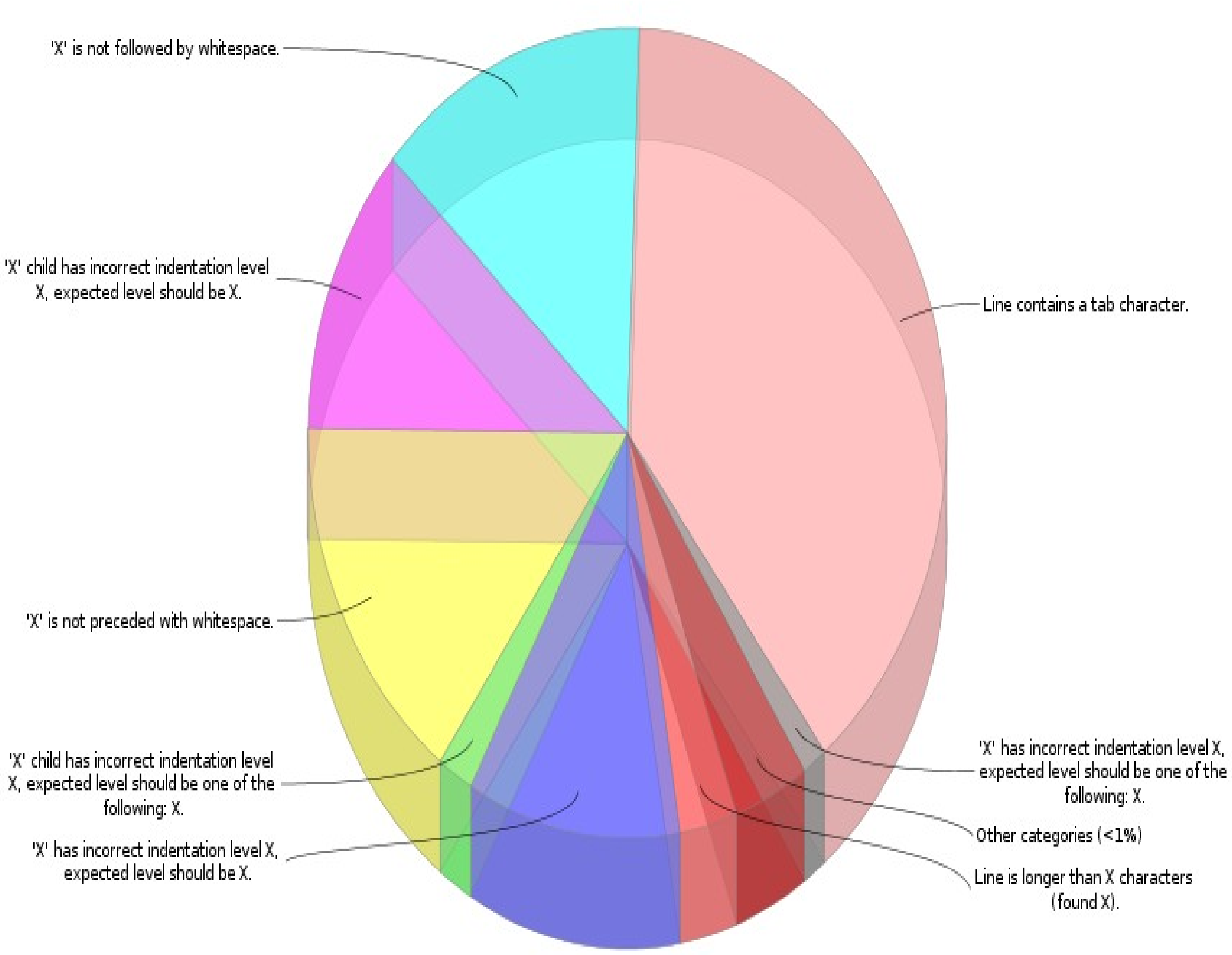
3. At the TableClustering package, the Cluster class is the responsible for keeping the information of a cluster. The ClusterInfoKeeper class should be renamed to ClusterMetricsComputator or something relevant because its name is misleading compared to the functionality it provides.

4. The Phase package doesn’t appear to have any serious issues to report at first glance.

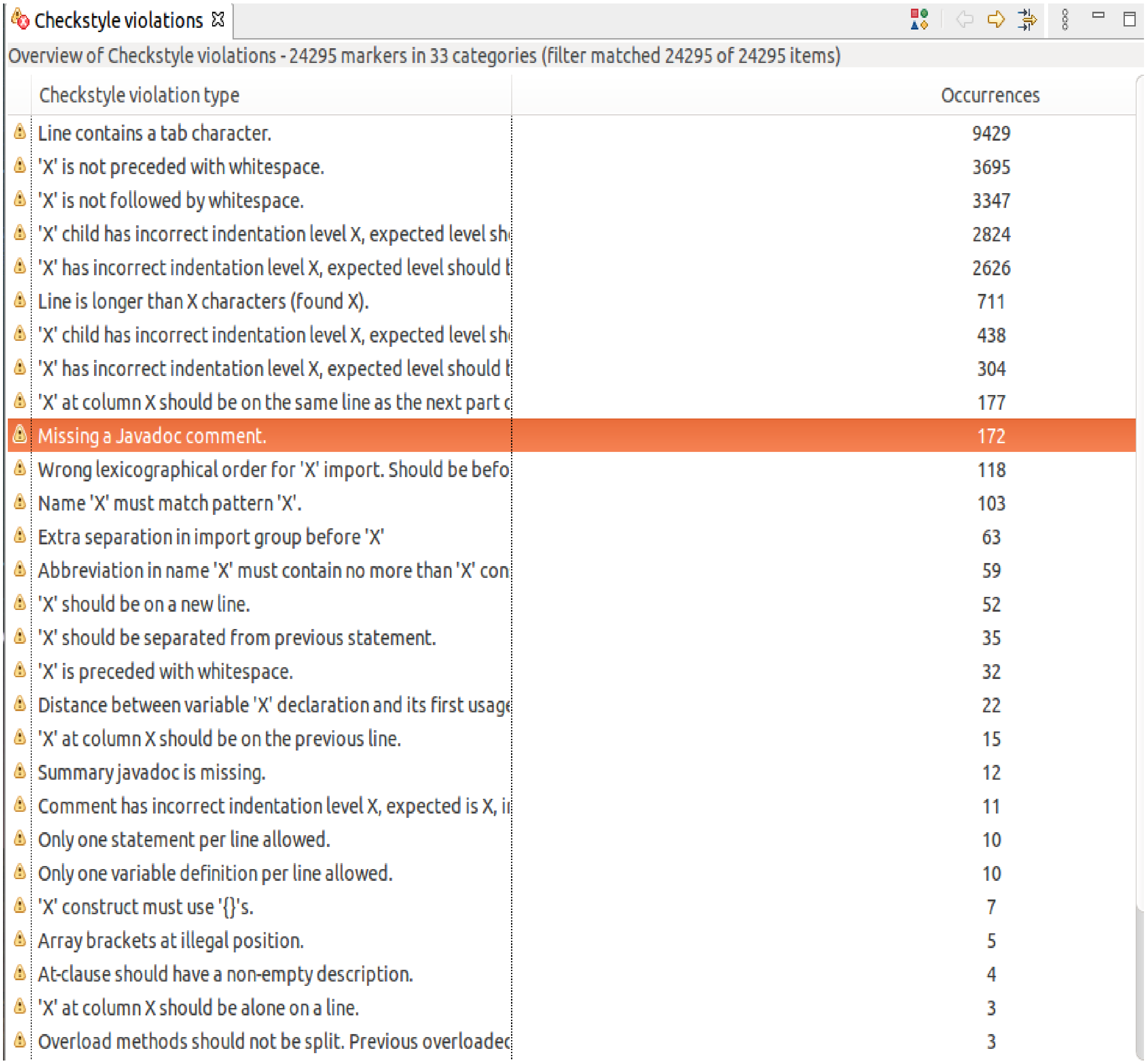
# CODE & STYLE LEVEL

## STYLE VIOLATIONS

### CheckStyle Pie Chart



# CheckStyle Violations



Checkstyle violations

Overview of Checkstyle violations - 24295 markers in 33 categories (filter matched 24295 of 24295 items)

| Checkstyle violation type | Occurrences |
|---|---|
| Line contains a tab character. | 9429 |
| 'X' is not preceded with whitespace. | 3695 |
| 'X' is not followed by whitespace. | 3347 |
| 'X' child has incorrect indentation level X, expected level sh | 2824 |
| 'X' has incorrect indentation level X, expected level should b | 2626 |
| Line is longer than X characters (found X). | 711 |
| 'X' child has incorrect indentation level X, expected level sh | 438 |
| 'X' has incorrect indentation level X, expected level should b | 304 |
| 'X' at column X should be on the same line as the next part c | 177 |
| Missing a Javadoc comment. | 172 |
| Wrong lexicographical order for 'X' import. Should be befo | 118 |
| Name 'X' must match pattern 'X'. | 103 |
| Extra separation in import group before 'X' | 63 |
| Abbreviation in name 'X' must contain no more than 'X' con | 59 |
| 'X' should be on a new line. | 52 |
| 'X' should be separated from previous statement. | 35 |
| 'X' is preceded with whitespace. | 32 |
| Distance between variable 'X' declaration and its first usage | 22 |
| 'X' at column X should be on the previous line. | 15 |
| Summary javadoc is missing. | 12 |
| Comment has incorrect indentation level X, expected is X, ir | 11 |
| Only one statement per line allowed. | 10 |
| Only one variable definition per line allowed. | 10 |
| 'X' construct must use '{}'s. | 7 |
| Array brackets at illegal position. | 5 |
| At-clause should have a non-empty description. | 4 |
| 'X' at column X should be alone on a line. | 3 |
| Overload methods should not be split. Previous overloadec | 3 |

# PRELIMINARY PLAN OF ACTIONS

1. Develop tests for the existing legacy system.

2. Add a Business logic package intermediary to the GUI and the Data packages.

3. Move methods that process data to the DataKeeper.

4. Divide the DataKeeper into smaller subsystems that handle the logic.

5. Apply general refactoring patterns on the code e.g. extract methods, duplicate code, clean dead code, spaghetti code etc.

6. If needed, add more tests to verify the functionality of the refactored version of the system.