

# Project 1

Aulon Mujaj, Henning Lund-Hanssen, Espen Jones

25th February 2016

## 1 Requirement 1 - Brief analysis

### 1.1 Brief description

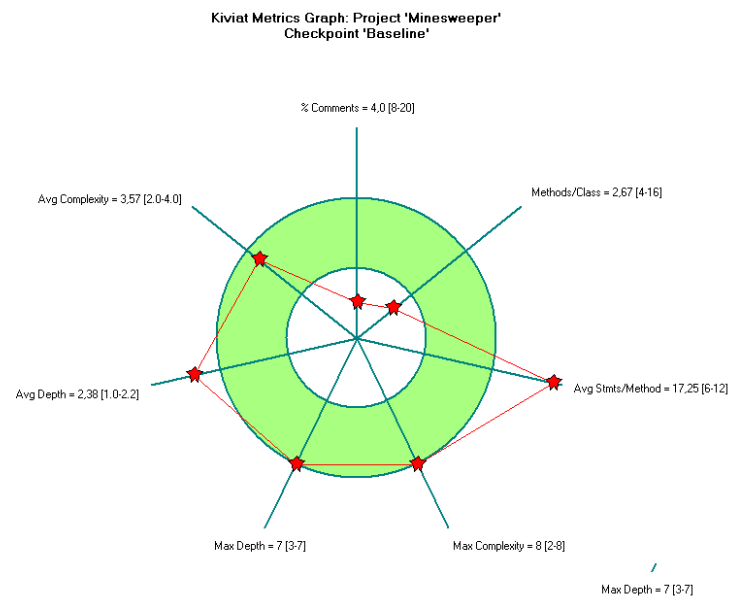
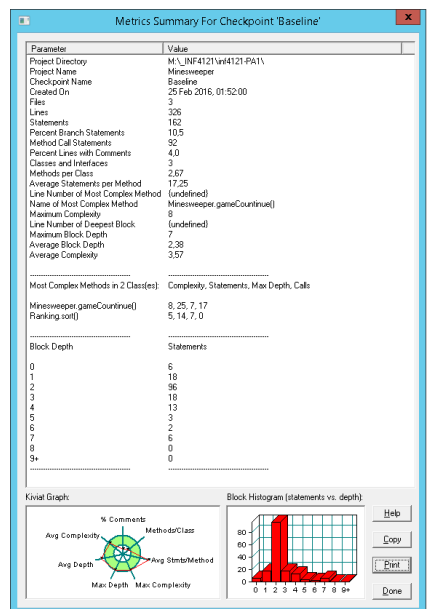
### 1.2 Analysis

### 1.3 Non-functional tests

### 1.4 Test cases

## 2 Requirement 2 - In-depth metrics

### 2.1 Metrics at project level



- What metrics do you spot for the whole project in the window Metrics Summary for Checkpoint? Write a brief description of the metrics. Try

to explain their values (below what is expected, as expected or above the expected level). What metrics do you think need to change?

- Which is the biggest file you have in your project by the number of lines of code?  
MineField.java
- Which is the file with most branches in your project?
- Which is the file with most complex code? What metric did you choose to answer to this question?  
Minesweeper.java

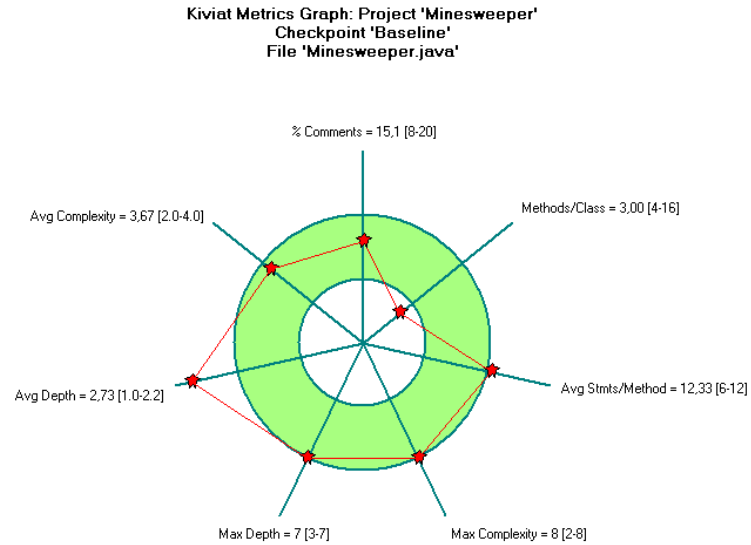
Write a little about each metric, maybe we'll only write about the 6 different metrics from the kivi diagram, instead of each metric from the metric summary image?

If we choose to go only for the metrics from the kivi diagram, then we'll only need to write about:

Avg complexity, avg depth, max depth, % comments, methods/class, avg stats/method, max complexity

Statements: Branch statements: Method call statements: Percent lines with comments: Classes and Interfaces Methods per class: Average Statements per Method: Name of Most complex methods Maximum Complexity Maximum Block Depth Average Block Depth Average Complexity Most complex methods (complexity, statements, max depth, calls) How many statements on each block depth

## 2.2 Metrics at file level



- How do you interpret the metrics applied on your file? How are they different the metrics you obtained on the whole project, compared with the metrics on his file?
- Would you refactor (re-write) any of the methods you have in this file?

## 3 Requirement 3 - Code improvement

### 3.1 Identification of metrics

### 3.2 Results from changes

### 3.3 Final remarks