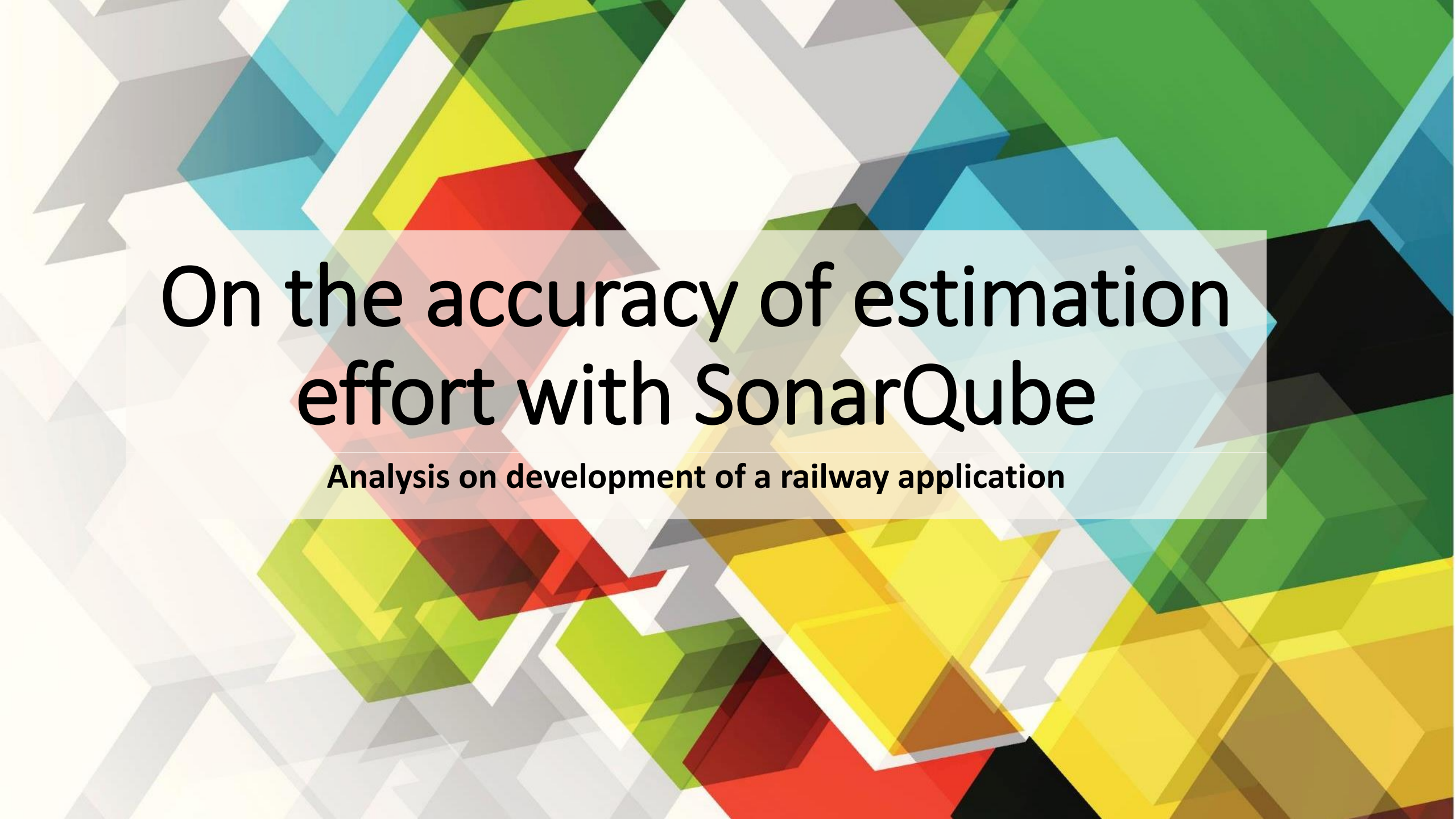

On the accuracy of estimation effort with SonarQube
Analysis on development of a railway application

# Work Plan

## PROJETC

PROJECT DATAILS

TEAM COMPOSITION

ROADMAP

## SCANNING REPORT

LEGENDA

ISSUE OPEN

## TIME REPORT

ESTIMATION

METRICS

## ACCURACY OF ESTIMATIONS

COMPARE

RELATED WORKS

# RoadMap

ONLINE KEY MANAGEMENT

# Project details

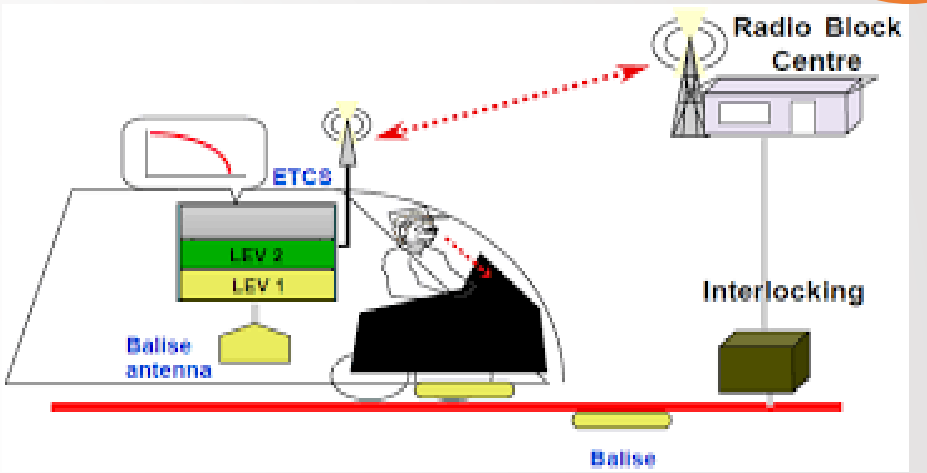
Implementation of an application for

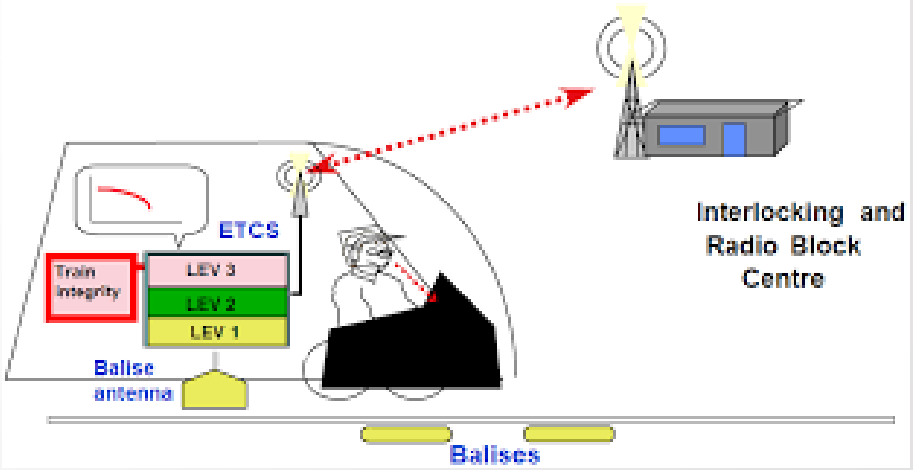
«**On-line Key Management**» protocol.

Part of the on-board system, responsibles for the exchange over the network of cryptographical keys need by EuroRadio.

Use to grant permission at driver to move the train in ERTMS/ETCS for:

- **Level 2**: train data exchanged from on-board radio and track equipments
- **Level 3**: train data exchanged only from on-board radio equipment



# Team components

## Project Architect

Handle architecture and organization.

Responsable of manage the documentation.

Handle the relation with the Product Owner.

## Developer #1

Contributes in design, develope and test the software component.

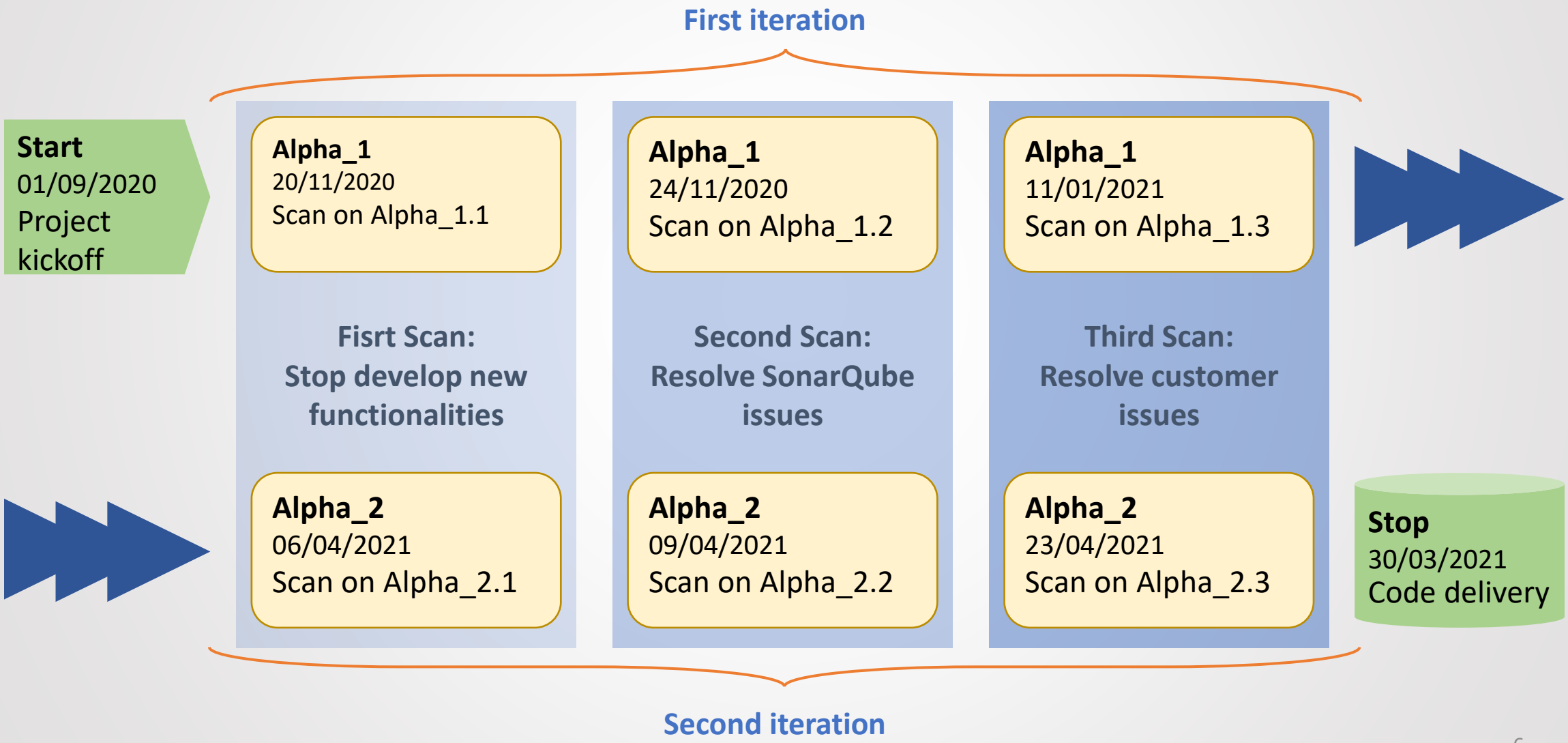
Mainly software developer.

## Developer #2

Contributes in design, develope and test the software component.

Mainly validation and verification developer.

# Roadmap


First iteration
Start
01/09/2020
Project kickoff
Alpha_1
20/11/2020
Scan on Alpha_1.1
Alpha_1
24/11/2020
Scan on Alpha_1.2
Alpha_1
11/01/2021
Scan on Alpha_1.3
Fisrt Scan:
Stop develop new functionalities
Second Scan:
Resolve SonarQube issues
Third Scan:
Resolve customer issues
Alpha_2
06/04/2021
Scan on Alpha_2.1
Alpha_2
09/04/2021
Scan on Alpha_2.2
Alpha_2
23/04/2021
Scan on Alpha_2.3
Stop
30/03/2021
Code delivery
Second iteration

# Scanning report

WITH SONARQUBE DEVELOPER EDITION 8.5

# Legenda

| Rule | Type of Issue | Severity of Issue | Status | Comment Message | Effort Estimeted | Version of Scan |
|---|---|---|---|---|---|---|
| c:S1135 | CODE_SMELL | INFO | OPEN | - | 0 | Alpha_1.1 |
| c:S2095 | BUG | BLOCKER | CLOSED | - | 5 | Alpha_1.2 |
| c:S1079 | VULNERABILITY | CRITICAL | OPEN | - | 10 | Alpha_1.3 |
| c:S1764 | BUG | MAJOR | OPEN | - | 2 | Alpha_2.1 |
| c:S1481 | CODE_SMELL | MINOR | OPEN | - | 5 | Alpha_2.2 |
| c:S1767 | BUG | CRITICAL | OPEN | - | 5 | Alpha_2.3 |

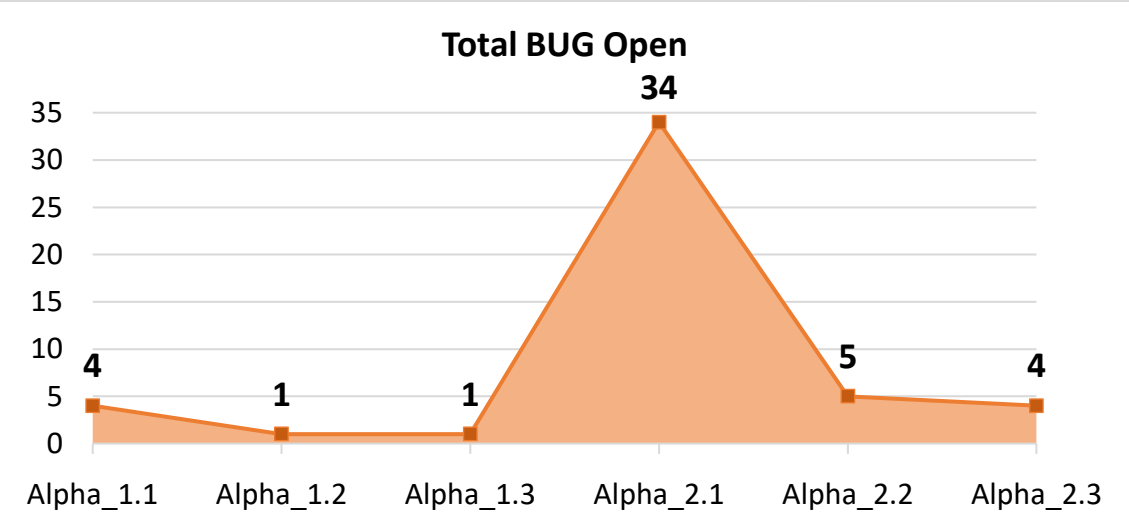
SonarQube definitions used for «Type of Issue»:

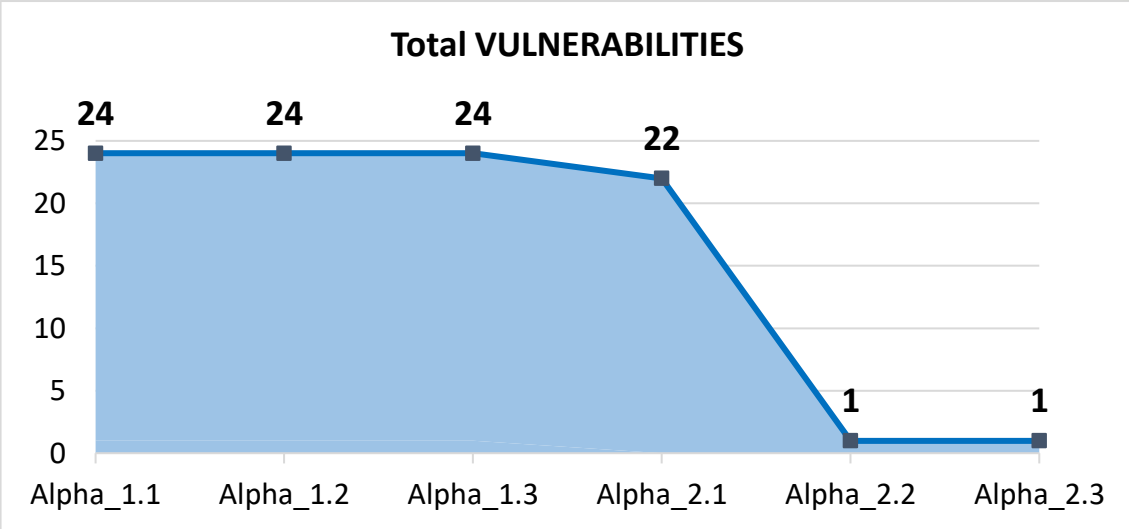
- **BUG**: issue that impact the reliability of the system,
- **VULNERABILITY**: issue that could impact the security of the system,
- **CODE**_SMELL: issue that impact the maintainability and the readability of the code.

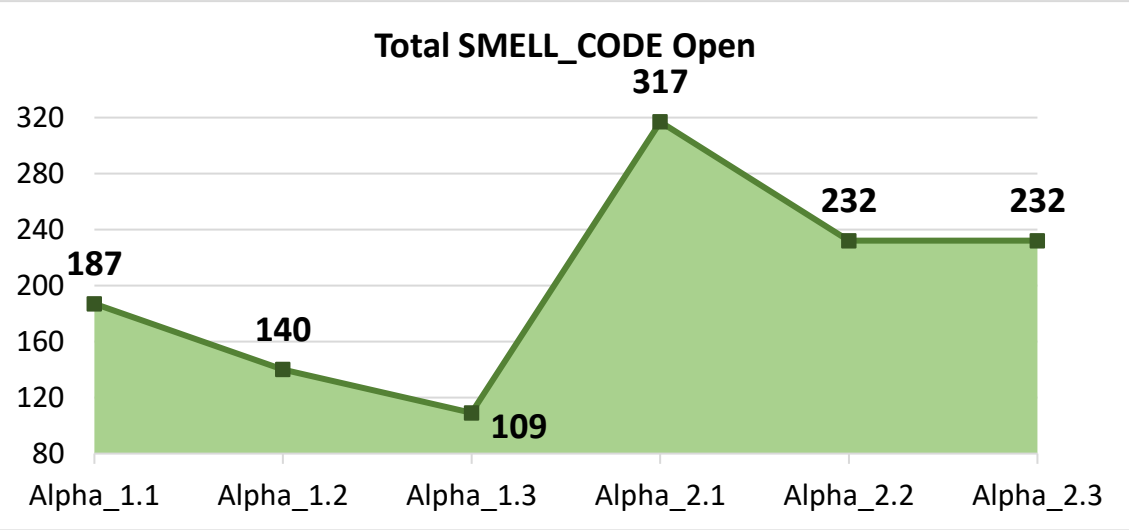
SonaQube definition for «Status»:

- **OPEN**: set automatically by the tool new issues to be resolved,
- **CLOSED**: set automatically by SonarQube for precedent Open issues no more present,
- **RESOLVED**: set manually to indicate that the next analysis should Close the issue.

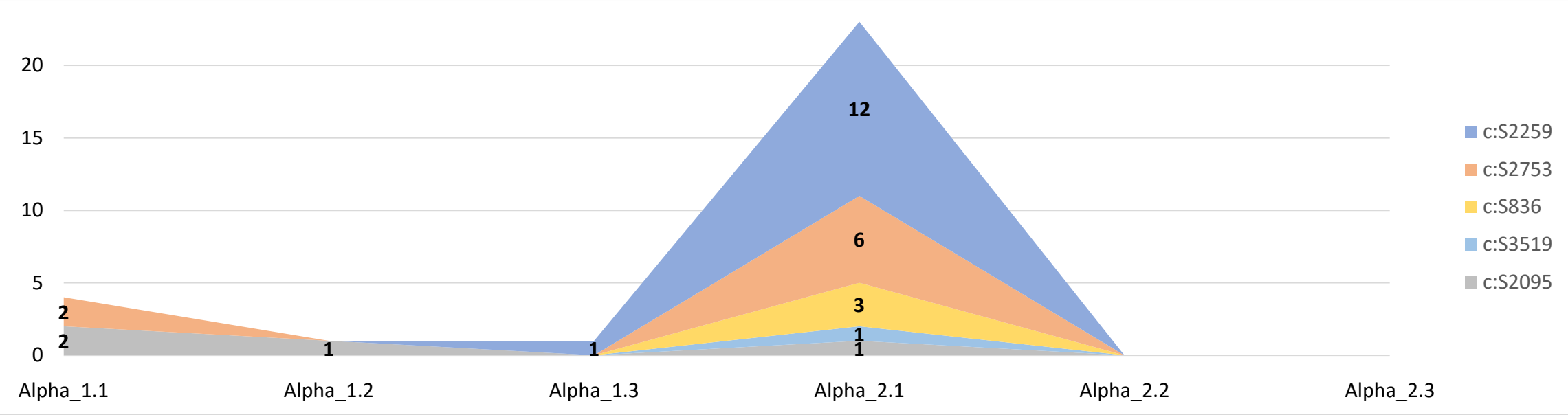
# Issues Open


Total BUG Open
35
30
25
20
15
10
5
0
4
1
1
34
5
4
Alpha_1.1
Alpha_1.2
Alpha_1.3
Alpha_2.1
Alpha_2.2
Alpha_2.3


Total VULNERABILITIES
25
20
15
10
5
0
24
24
24
22
1
1
Alpha_1.1
Alpha_1.2
Alpha_1.3
Alpha_2.1
Alpha_2.2
Alpha_2.3


Total SMELL_CODE Open
320
280
240
200
160
120
80
187
140
109
317
232
232
Alpha_1.1
Alpha_1.2
Alpha_1.3
Alpha_2.1
Alpha_2.2
Alpha_2.3

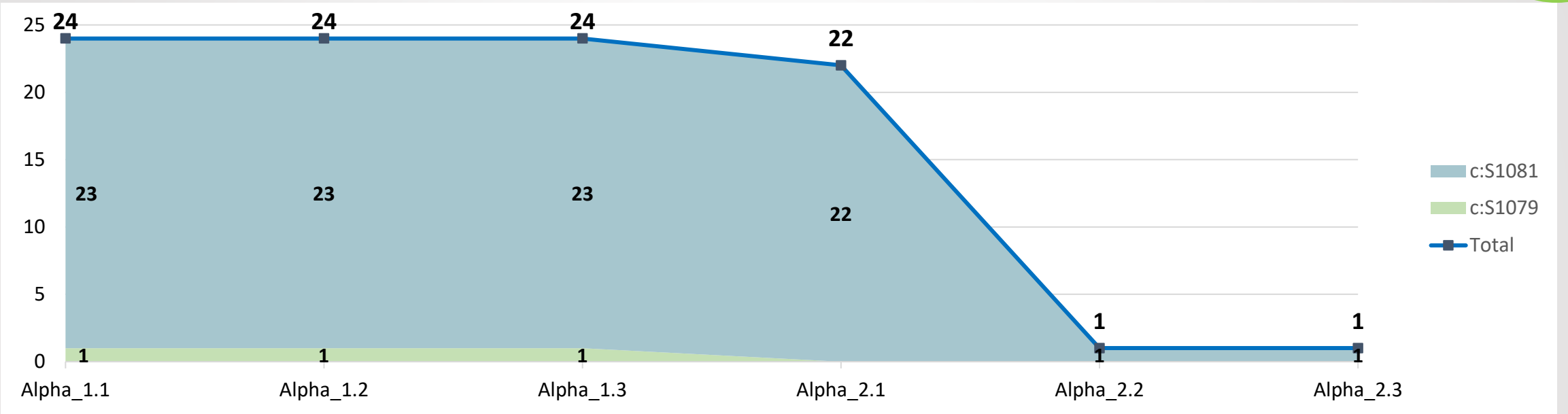
# Selected BUG Open




Selected subset of 5 BUG from all SonarQube Issues that affected the most the code, causing unexpected failure during development:

- c:S2259 – Access to field is NULL pointer
- c:S2753 – Implicit conversion of ENUMERATION
- c:S2095 – Opened file never closed
- c:S5836 – Undefined pointer value
- c:S3519 – Out of bound access

# Selected VULNERABILITIES Open

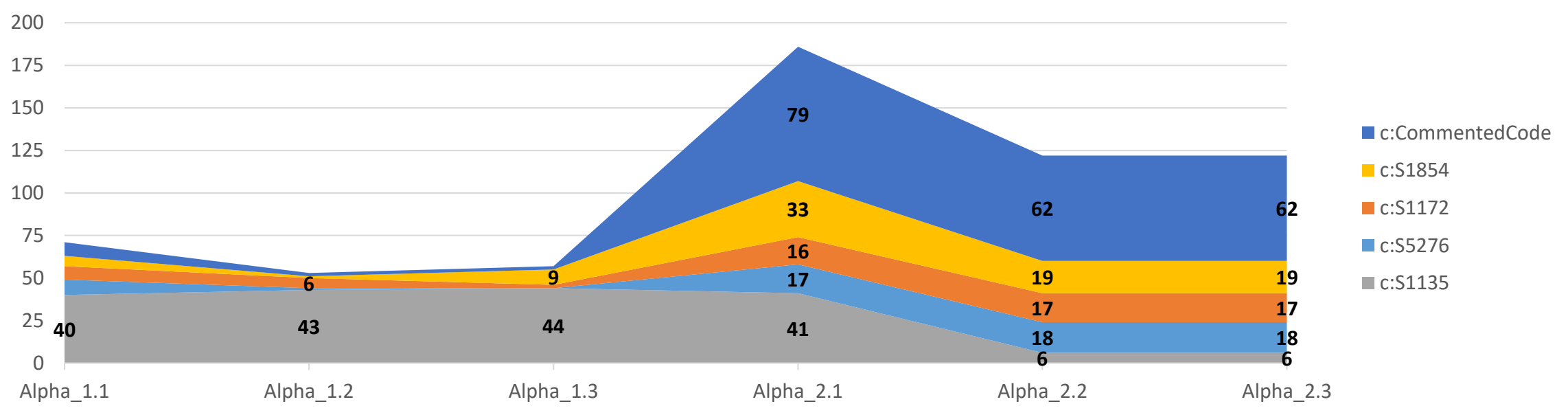



Selected VULNERABILITIES that affected the code.

The issues don't affect the behavior of the code, but we choose to improve the control for future integration:

- c:S1079 – Remove use of insecure function printf()
- c:S1081 - Add a field width specifier to this "%s" placeholder

# Selected SMELL_CODE Open




Selected CODE_SMELL focused by the team.

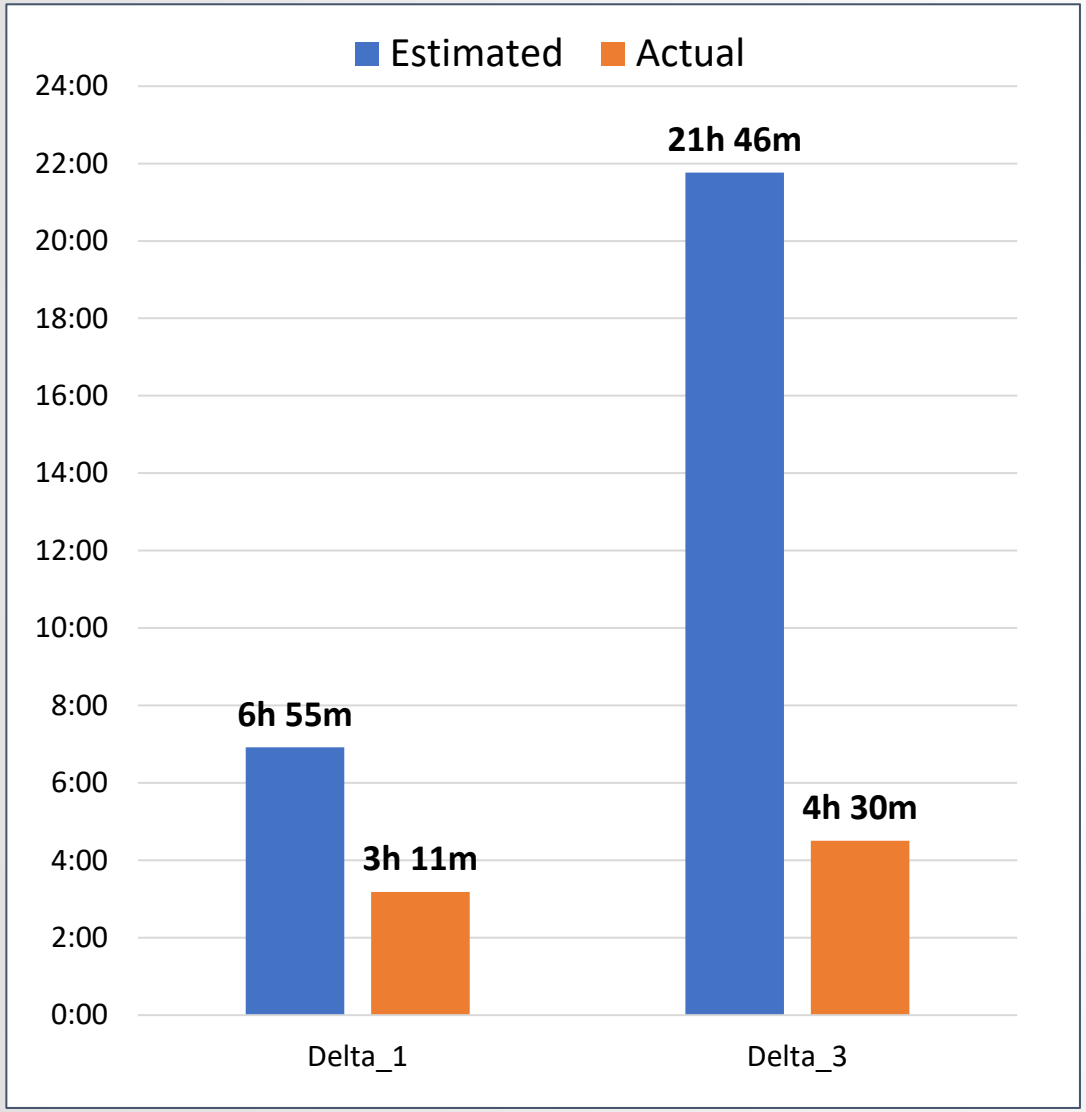
The issues only affect the readability of the code. We mostly focused on them for better interpretation of future integration:

- c:CommentedCode – Comment read as code
- c:S1854 – Value stored in variable is never read
- c:S1172 – Unused parameter
- c:S5276 – Implicit conversion lose precision
- c:S1135 – Complete the task with TODO

# Time report

WITH ACTUAL TIME

# Remediation time of SonarQube Issues




Comparison between the SonarQube estimated time versus the Actual remediation time:

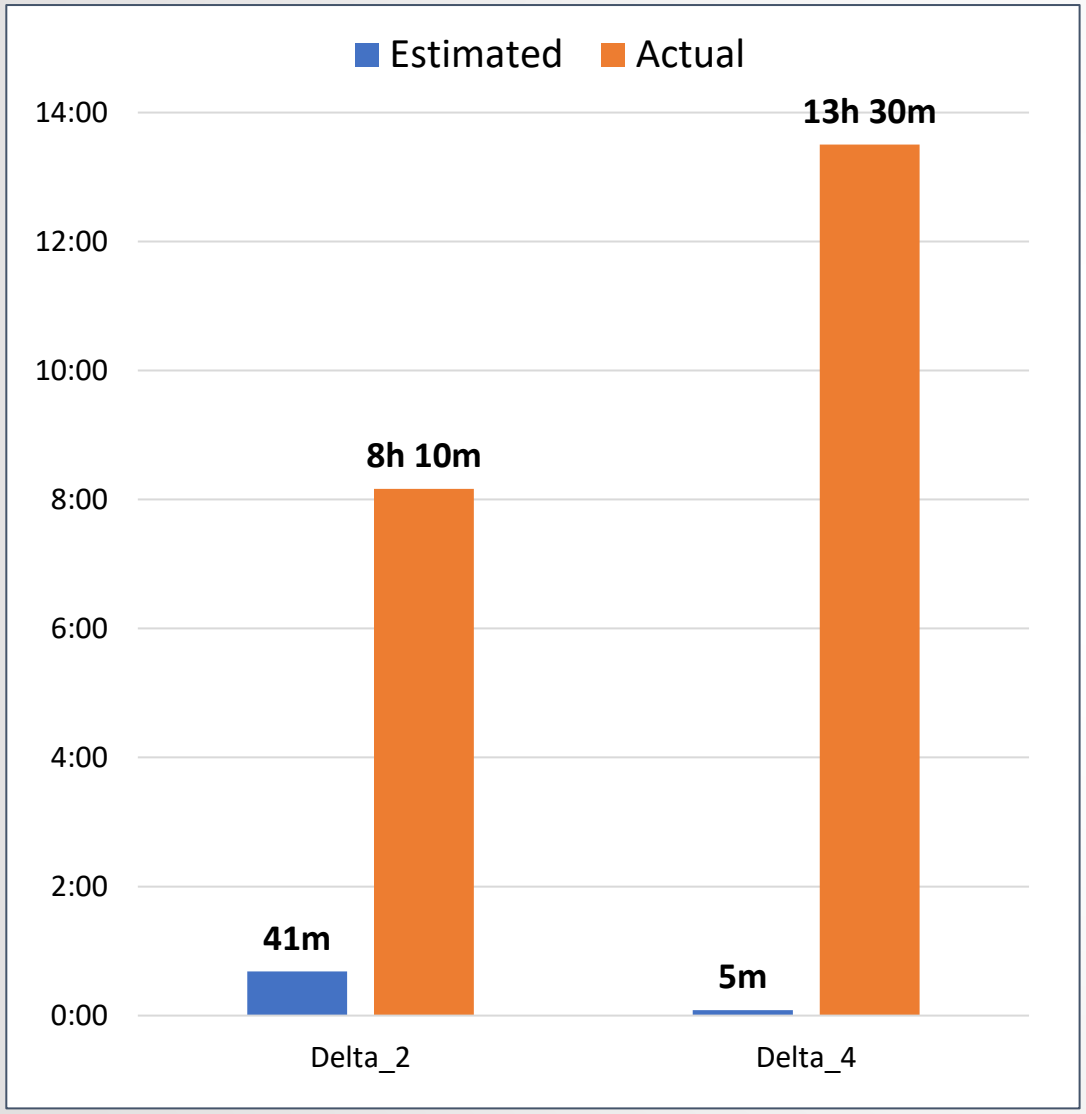
- Delta_1 : defined as the difference in estimation from scan on Alpha_1.1 and Alpha_1.2;
- Delta_3 : defined as the difference in estimation from scan on Alpha_2.1 and Alpha_2.2.

*Magnitude Remediation Estimated:*

$$MRE = \frac{|Act_{GE} - \Delta Est_{GE}|}{Act_{GE}}$$

- ***Delta_1 = 1,17***
- ***Delta_3 = 3,84***

# Remediation time of Costumer Issues




Comparison between the estimated time versus the Actual remediation time for costumer issues:

- Delta_2 : defined as the difference in estimation from scan on Alpha_1.2 and Alpha_1.3;
- Delta_4 : defined as the difference in estimation from scan on Alpha_2.2 and Alpha_2.3.

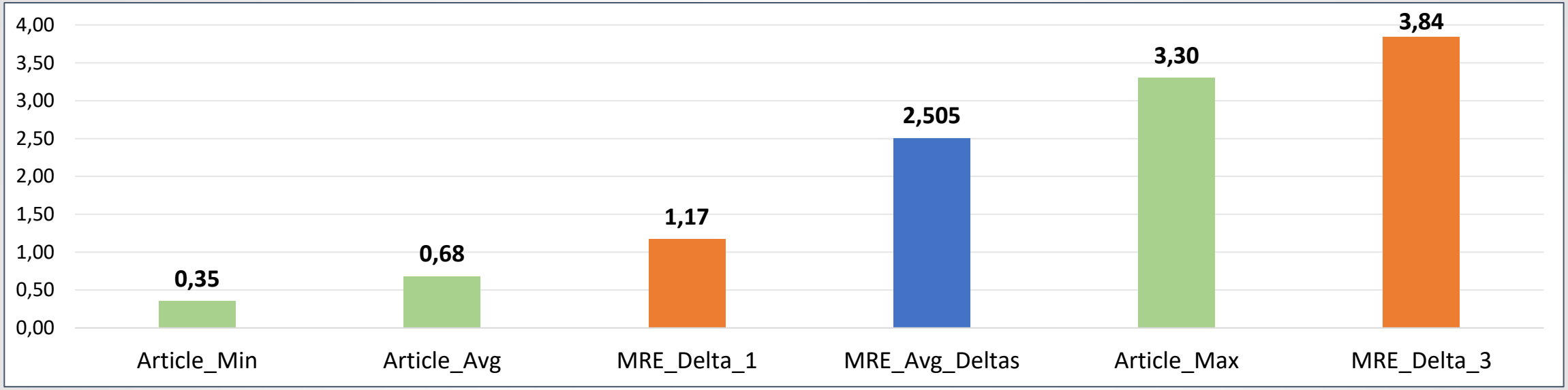
*Impact of remediation:*

$$Imp = \frac{|Act_{GE} - \Delta Est_{GE}|}{\Delta Est_{GE}}$$

- ***Delta_2 = 10,95***
- ***Delta_3 = 161,00***

# Accuracy of estimations

# Comparison with other study




Comparison in Magnitude Relative Error:

- In **Green** the Minumun, Maximum and Average of MRE exposed in

  «**On the diffuseness of technical debt items and accuracy of remediation time when using SonarQube**»;
- In **Orange** the MRE calculated for the Delta_1 and Delta_3;
- In **Blue** the Average of the calculated Deltas.

# Conclusions and further study

Hypotesis to explain the poor accuracy of the tool:

| | | |
|---|---|---|
| Presence of strong coding-rules | → | the code is more standardized, presence of fiew variants for similar functionalities |
| Multiple instances of the same issue | → | same resolution with easy compreension of the code |
| Small team composition | → | The same developer fix the code that wrote, reduce time for compreension |
| No need to generate unit test by design | → | less effort per issue |

Further studies could increase the data-set and/or increase the detail of the analysis for class of issues.

Studies with different teams and organisation could show if the metodologies help in reducing the Technical Debt.