QAA assignment

Blekinge Tekniska Högskola, Oct 2023

Hammam Dowah - Karam Kirawan - Abdulkarim Dawalibi - Adnan Altukleh

1. Introduction and Description of the System

1.1 Choice of System and Quality Attribute

Pandas is an open-source library for Python programming language, pandas is used in many use cases such as data analysis/manipulation, reading and writing data, handling missing data, reshaping datasets, aggregating, transforming, and much more. We have chosen the following quality attributes:

- Modifiability

Given the sensitive scenarios the library was created to perform, users might need to add new functionalities or adapt existing ones to specific needs. Therefore pandas have to be dynamic when making changes to the system without introducing unforeseen complications.

- Testability

Because of the sensitivity of the use cases of the library, it can’t tolerate any undefined behaviour when applying changes to the library. For this reason, it is crucial that any modifications or additions to pandas are verifiable for correctness. This aids in preventing regressions and ensures consistent behaviour across versions.

- Reliability

Reliability is important for the Pandas library since many developers rely on Pandas to perform data analysis, especially in Artificial Intelligence and Machine Learning. These areas often have real-world applications, so it's crucial that the results from Pandas are trustworthy and consistent. This trust in the tool's reliability is paramount for developers/users who base their decisions and insights on the results produced using pandas. This is why reliability is chosen as the third quality attribute as testability contributes to increasing the reliability.

1.2 Summary of Status Quo

1.2.1 Modifiability

In the current state, the pandas library does support modifying the existing functions to achieve modifiability. Also, looking at the change coupling diagram, we can observe that certain files have some kind of coupling which decreases the level of modifiability [2].

1.2.2 Testability

The Pandas library has an overall coverage of 96% according to their code coverage report [1]. Also, there are a lot of test files which test the systems different functions [2].

1.2.3 Reliability

The Pandas library currently has 2,565 bugs, resulting in a low reliability rating of 'E' due to the presence of severe Blocker bugs. Sonarcloud estimates that addressing all bugs would require approximately 44 days of effort [3].

2. Modifiability

2.1 General Quality Attribute Scenario

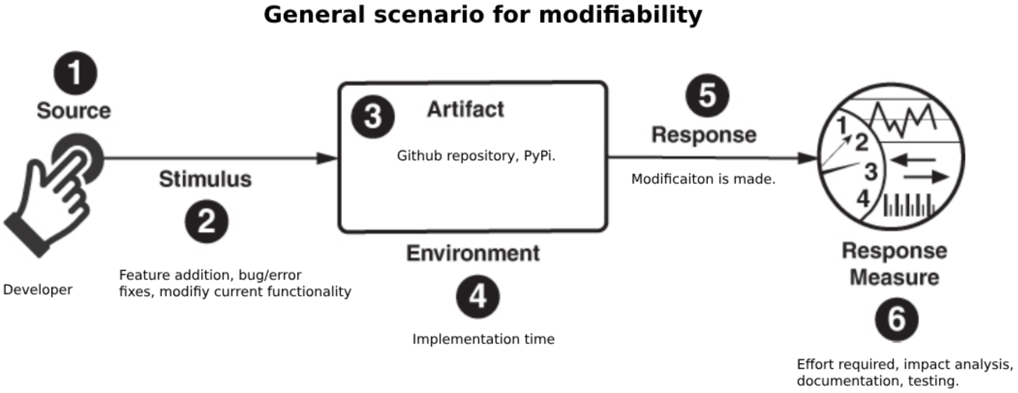

General scenario for modifiability
1
Source
Developer
Stimulus
2
Feature addition, bug/error fixes, modifiy current functionality
3
Artifact
Github repository, PyPi.
Environment
4
Implementation time
5
Response
Modificaiton is made.
Response Measure
6
Effort required, impact analysis, documentation, testing.

Figure 1: The figure shows a general scenario for Modifiability

2.2 Specific Quality Attribute Scenario

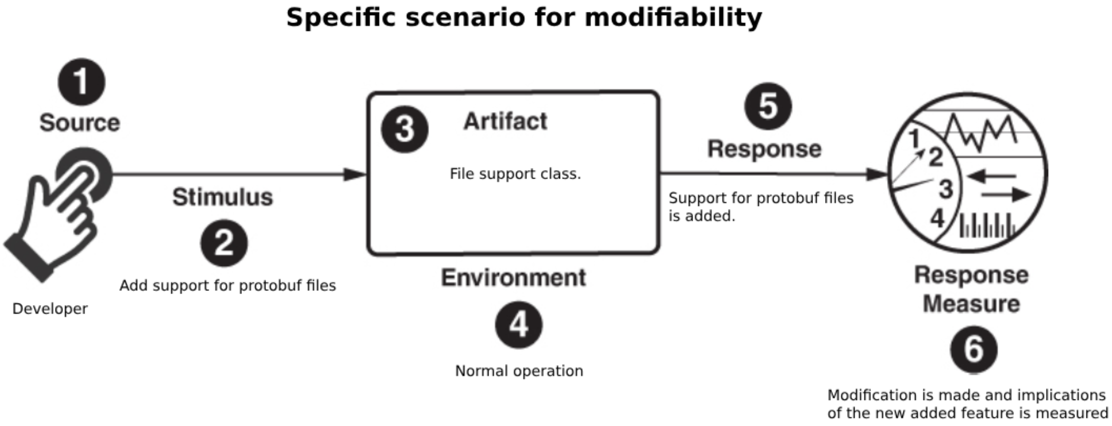

Specific scenario for modifiability
1
Source
Developer
Stimulus
2
Add support for protobuf files
3
Artifact
File support class.
Environment
4
Normal operation
5
Response
Support for protobuf files is added.
Response Measure
6
Modification is made and implications of the new added feature is measured

Figure 2: The figure shows a specific scenario for Modifiability

2.3 Measurements

In the current state, Pandas supports library extension, especially for users who seek to augment or refine the existing code behaviour to meet their requirements better. The extending can be done through registering custom accessors, extension types, subclassing pandas data structures, plotting backends, and arithmetic with 3rd party types [4].

Examining the change coupling diagram, it's evident that certain files are dependent on one another. When one file undergoes modification, there's a high chance that the other file will require adjustments as well. A notable absence of coupling between the test files and the actual system is observed, which significantly enhances the ease of modifiability.

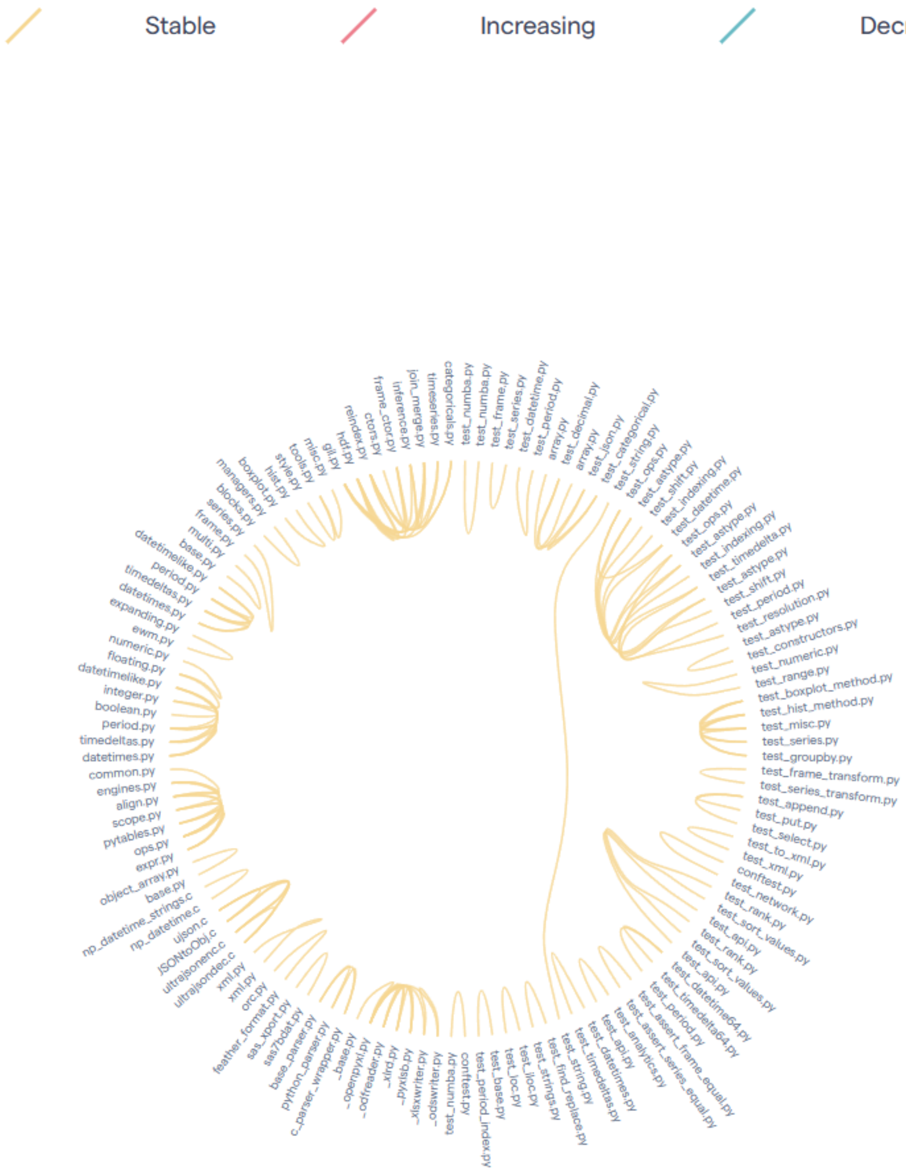


Figure 3: The figure shows a change coupling diagram

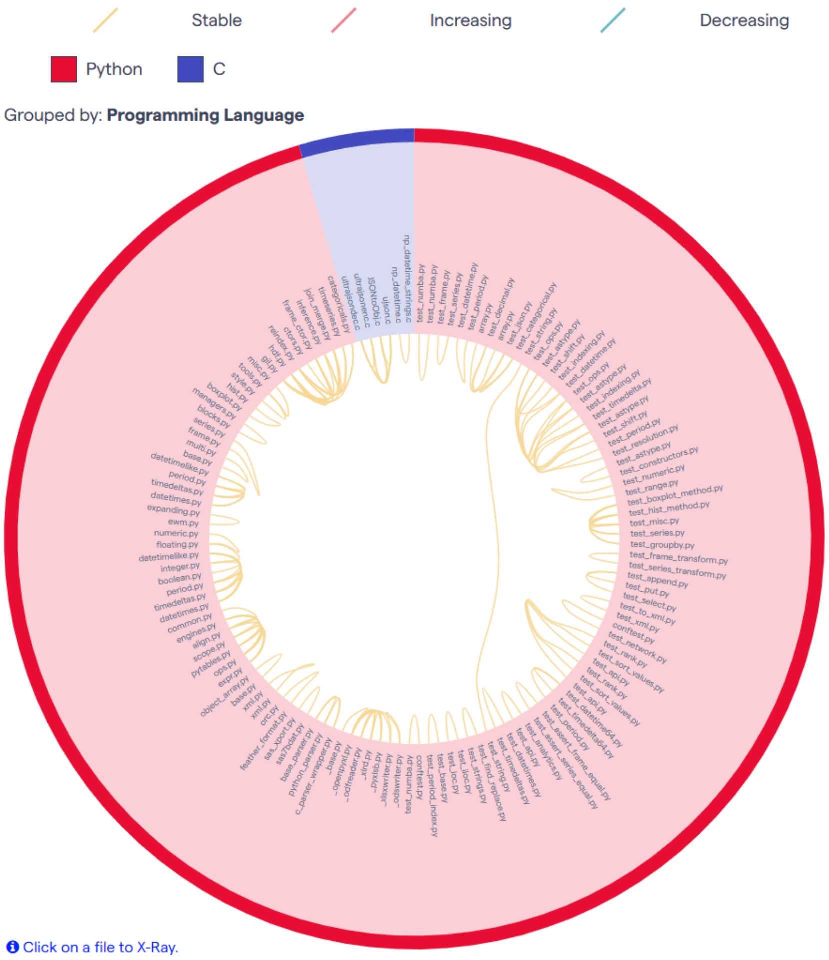


Figure 4: The figure shows a change coupling diagram grouped by programming language

We are examining the change coupling diagram based on the programming language. It's observed that the system's main programming language is Python, which further facilitates modifiability.

2.4 Decisions to improve

To improve modifiability, we need to:

- Separate tightly coupled components and aim for a modular design.
- Monitoring the system's performance and analysing the impact of modifications can help in understanding and improving the modifiability of the system over time.

3. Testability

3.1 General Quality Attribute Scenario

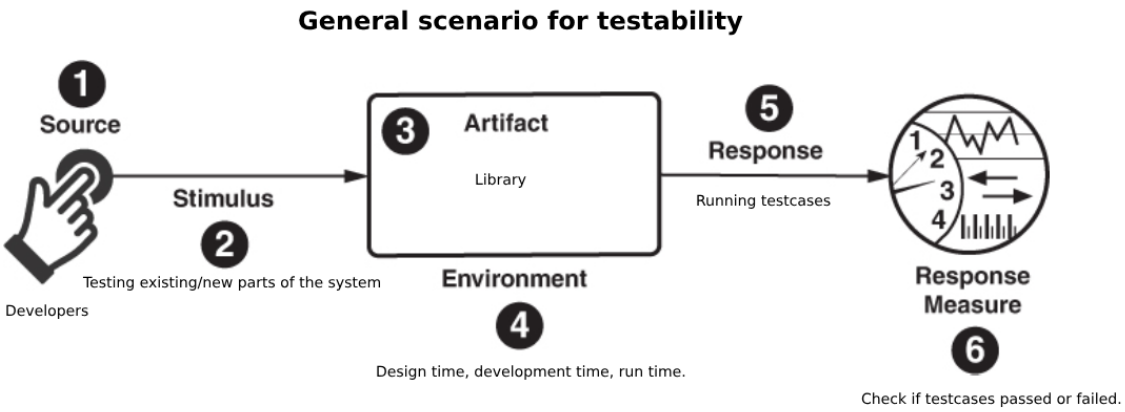

General scenario for testability
1
Source
Developers
Stimulus
2
Testing existing/new parts of the system
3
Artifact
Library
Environment
4
Design time, development time, run time.
5
Response
Running testcases
Response Measure
6
Check if testcases passed or failed.

Figure 5: The figure shows a general scenario for Testability

3.2 Specific Quality Attribute Scenario

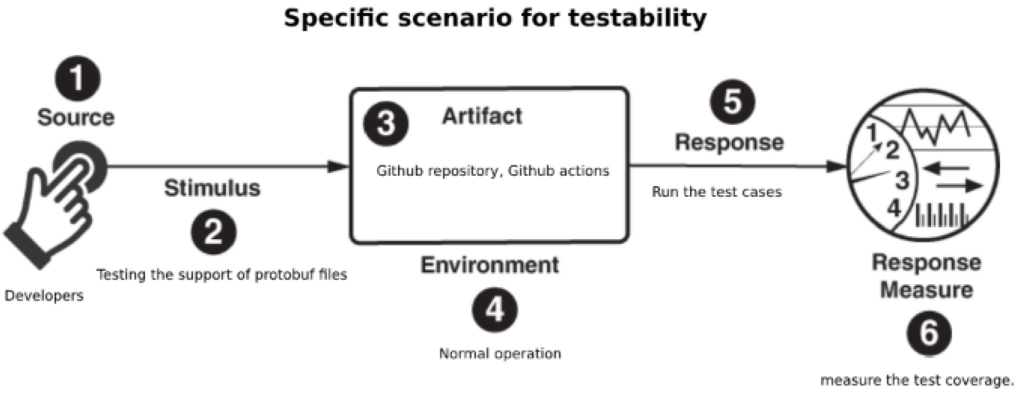
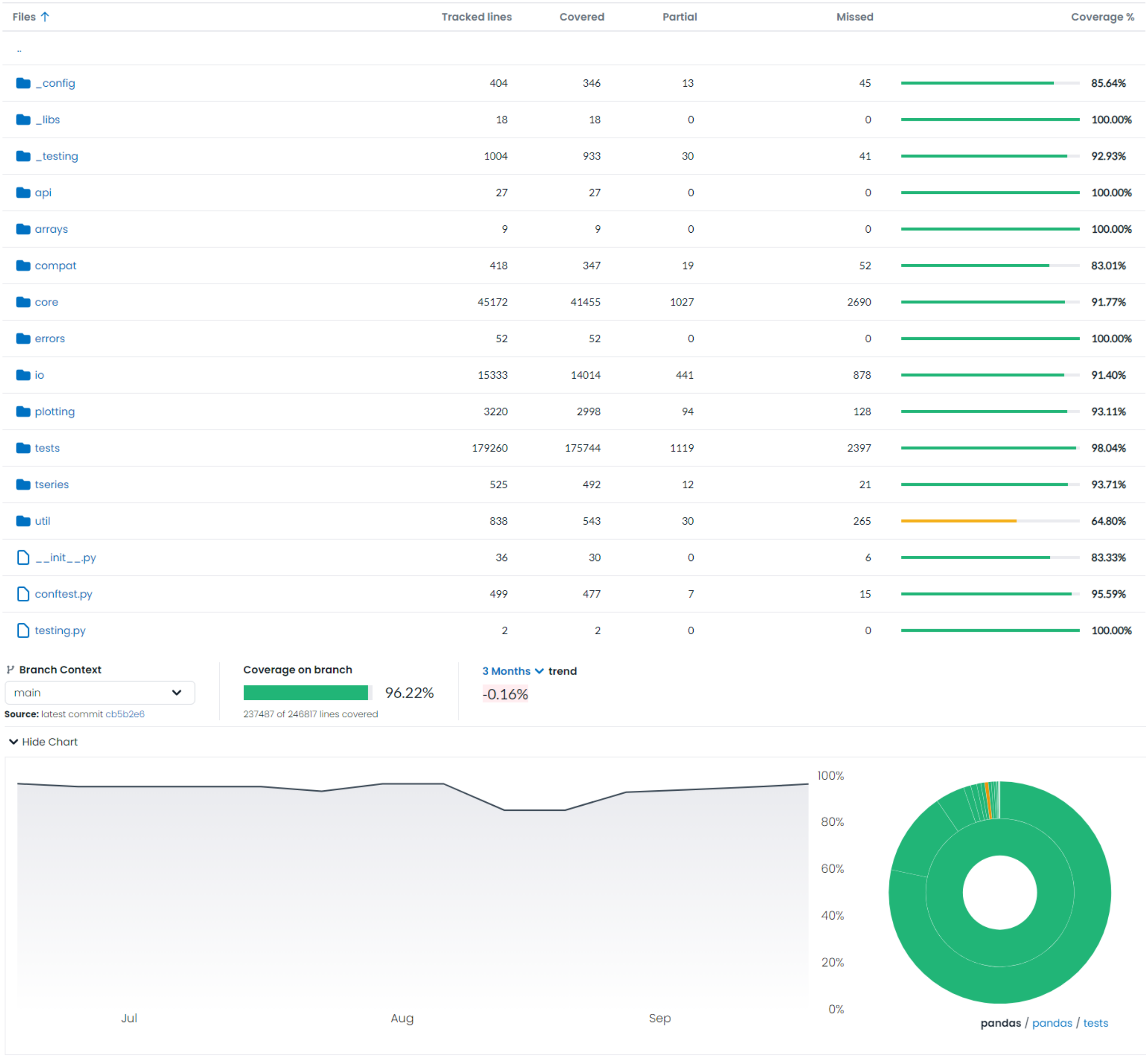

Specific scenario for testability
1
Source
Developers
Stimulus
2
Testing the support of protobuf files
3
Artifact
Github repository, Github actions
Environment
4
Normal operation
5
Response
Run the test cases
Response Measure
6
measure the test coverage.

Figure 6: The figure shows a specific scenario for Testability

3.3 Measurements

In the current state, the test coverage of the main branch is around 96 % which means that a big portion of the code is covered by tests. By looking at figure 2, we can see that most of the folders in Pandas exhibit high coverage, with only a few exceptions where coverage levels vary. Some folders have 100% coverage indicating comprehensive testing. Additionally, over the past three months, the test coverage of the Pandas library has experienced a slight decrease, with a decrease of -0.16%. [1]



Files ↑	Tracked lines	Covered	Partial	Missed	Coverage %
..					
_config	404	346	13	45	85.64%
_libs	18	18	0	0	100.00%
_testing	1004	933	30	41	92.93%
api	27	27	0	0	100.00%
arrays	9	9	0	0	100.00%
compat	418	347	19	52	83.01%
core	45172	41455	1027	2690	91.77%
errors	52	52	0	0	100.00%
io	15333	14014	441	878	91.40%
plotting	3220	2998	94	128	93.11%
tests	179260	175744	1119	2397	98.04%
tseries	525	492	12	21	93.71%
util	838	543	30	265	64.80%
__init__.py	36	30	0	6	83.33%
conftest.py	499	477	7	15	95.59%
testing.py	2	2	0	0	100.00%

Branch Context

main

Source: latest commit cb5b2e6

Coverage on branch

96.22%

237487 of 246817 lines covered

3 Months trend

-0.16%

Figure 7: The figure shows code coverage for different system files

3.4 Decisions to improve

To improve testability, we can:

- Analyse the areas with low coverage and direct the effort to increase the coverage.
- We can check the reason behind the decrease of the overall coverage over the past three months. This might provide valuable information and might reveal areas where testing was overlooked or deemphasized.

4. Reliability

4.1 General Quality Attribute Scenario

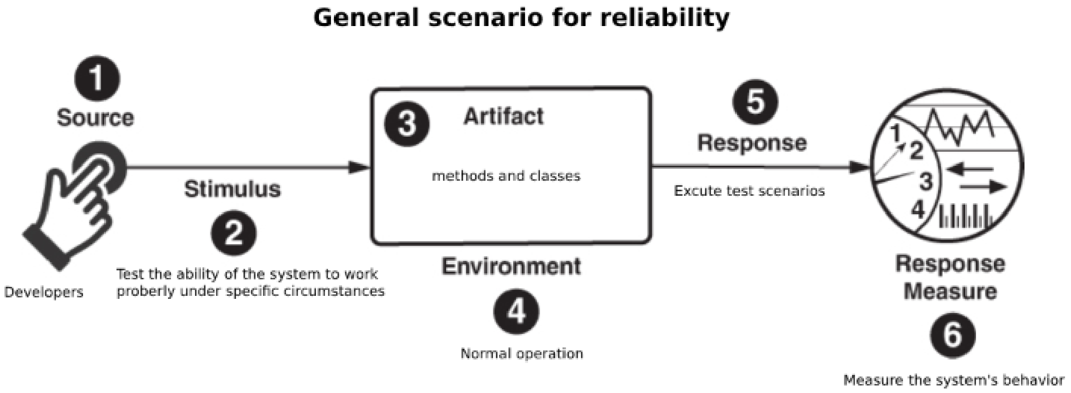

General scenario for reliability
1
Source
Developers
Stimulus
2
Test the ability of the system to work proberly under specific circumstances
3
Artifact
methods and classes
Environment
4
Normal operation
5
Response
Excute test scenarios
Response Measure
6
Measure the system's behavior

Figure 8: The figure shows a general scenario for Reliability

4.2 Specific Quality Attribute Scenario

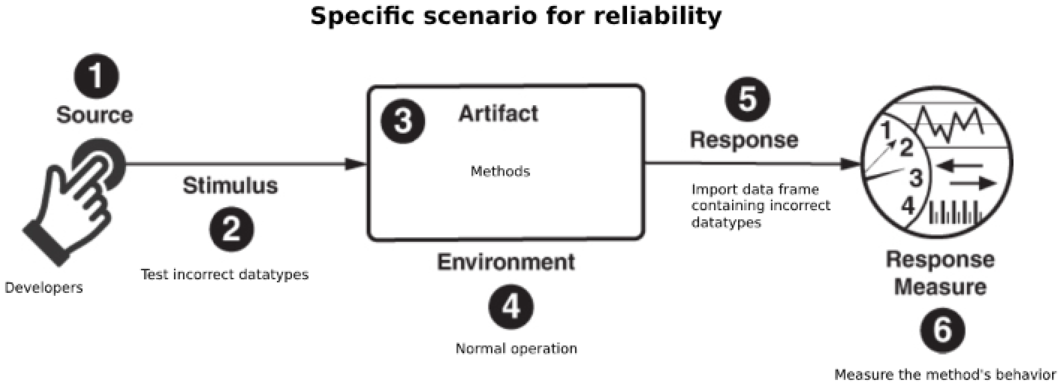
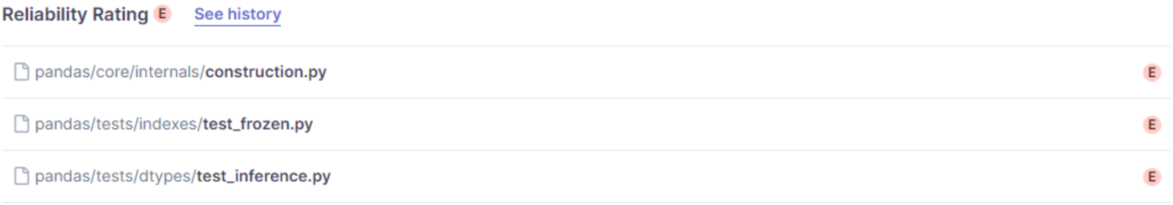


Figure 9: The figure shows a specific scenario for Reliability

4.3 Measurements

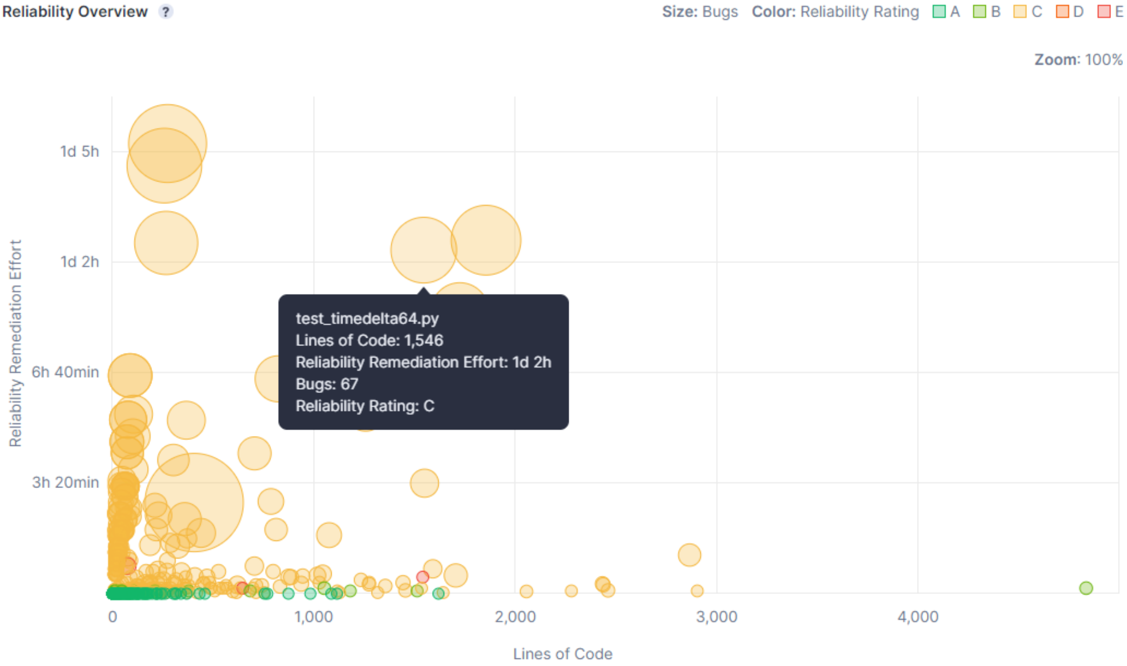
The current state of reliability for the Pandas library reveals a total of 2565 bugs, earning it a rating of 'E'. This rating proves the presence of more than one Blocker bug within the system, as illustrated in Figure 10. A Blocker bug represents a severe category of bugs, causing a substantial negative impact on the system. In such frameworks, a low-reliability rating of 'E' for Pandas is quite detrimental, and ideally, it should be significantly higher. Figure 11 provides a comprehensive overview of the reliability aspect. Within this figure, the size of the bubbles corresponds to the number of bugs identified in the file, the larger the bubble gets, the more bugs exist within the file. Additionally, the color intensity of a bubble, approaching red, denotes the severity of the worst bugs encountered. The vertical position of the bubbles reflects the estimated time to address the bugs. The majority of the bubbles have an estimated time < 4h. In order to obtain better reliability than the current state in the system it requires more costly effort to reduce the number of bugs, the estimated time effort to solve all bugs according to sonarcloud is around 44 days [3].



Reliability Rating E See history

pandas/core/internals/**construction.py**	E
pandas/tests/indexes/**test_frozen.py**	E
pandas/tests/dtypes/**test_inference.py**	E

Figure 10: The figure shows a Reliability rating



test_timedelta64.py
Lines of Code: 1,546
Reliability Remediation Effort: 1d 2h
Bugs: 67
Reliability Rating: C

Figure 11: The figure shows a bugs overview

4.4 Decisions to improve

- Implementing robust error handling mechanisms within Pandas to handle unexpected data and user inputs. This will prevent crashes and ensure the library behaves predictably, even when faced with challenging scenarios.
- Improve encryption mechanisms to protect sensitive data. This includes encrypting data at rest and in transit to prevent unauthorised access.
- Improve compatibility with other popular libraries in the data science ecosystem, such as NumPy, SciPy, and scikit-learn.

References

[1] Codecov, https://app.codecov.io/gh/pandas-dev/pandas/tree/main/pandas
[2] Codescene, https://codescene.io/projects/44293/jobs/1612127/results?scope=month#code-health
[3] Sonarcloud, https://sonarcloud.io/summary/overall?id=Hemofrags_pandas
[4] Pandas, https://pandas.pydata.org/docs/development/extending.html