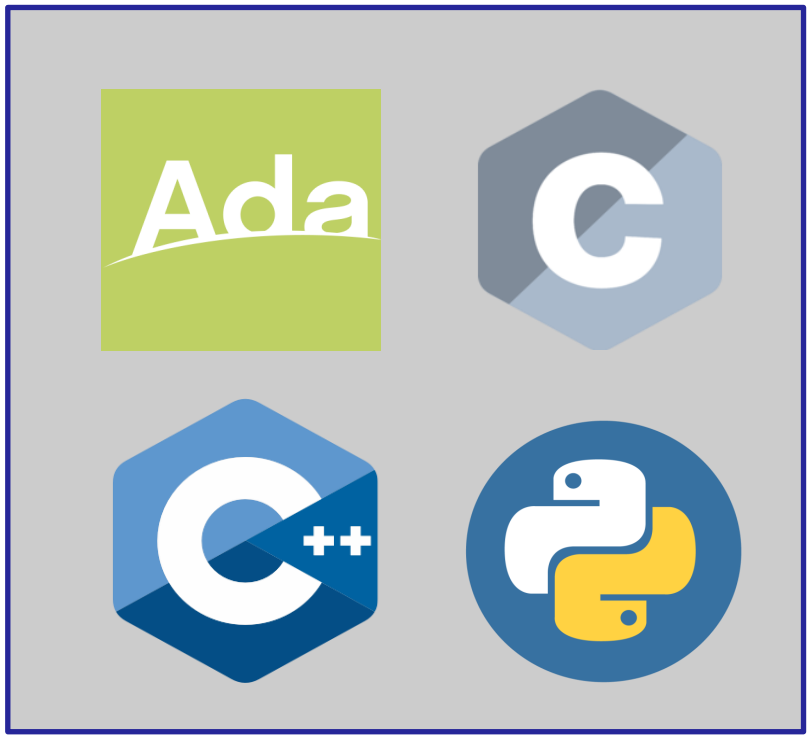
STATIC ANALYSIS FOR ADA, C/C++ AND PYTHON: DIFFERENT LANGUAGES, DIFFERENT NEEDS.



Maurizio Martignano
Spazio IT – Soluzioni Informatiche s.a.s
Via Manzoni 40
46051 San Giorgio Bigarello, Mantova
https://www.spazioit.com

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Agenda

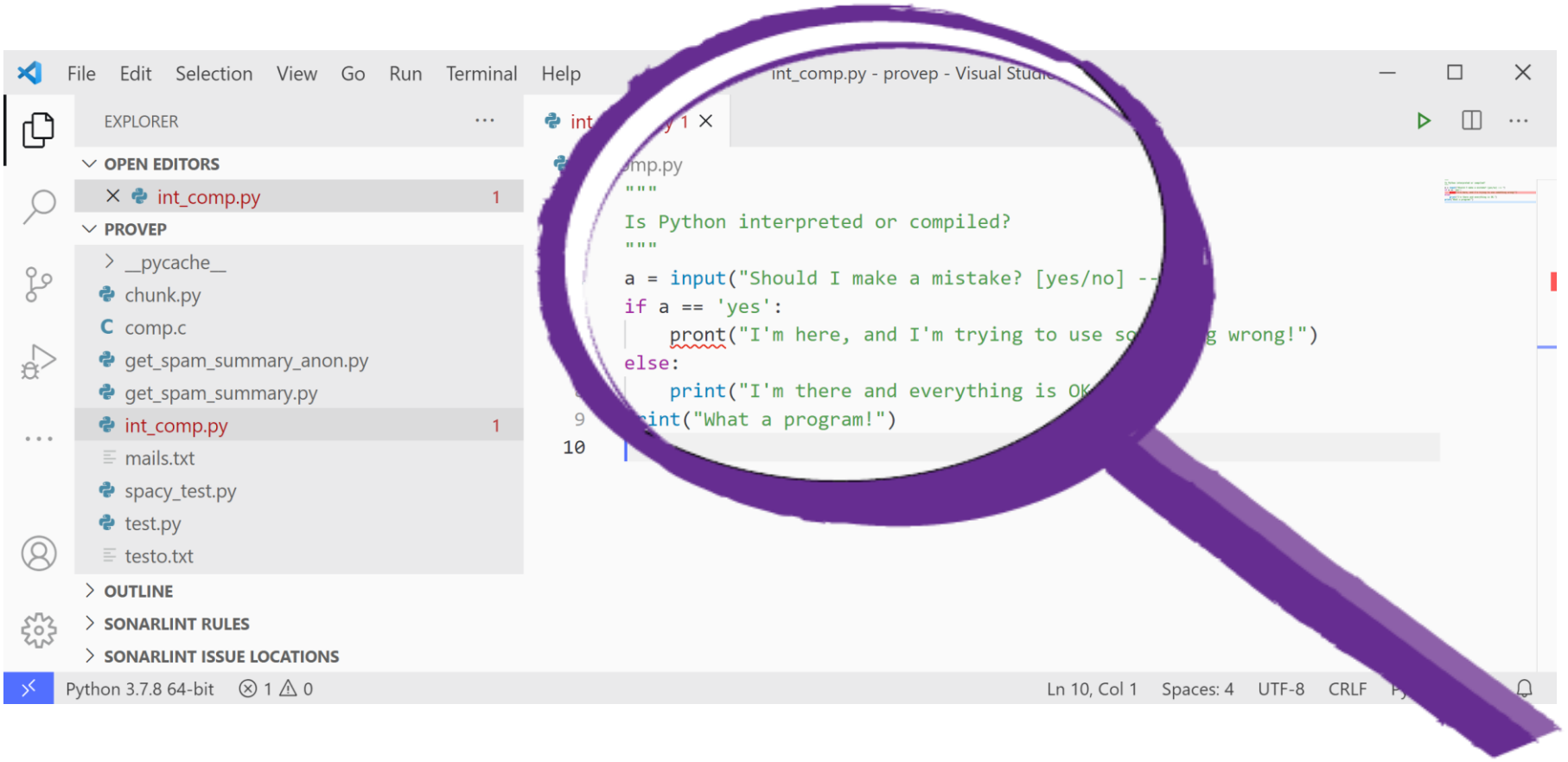
© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Agenda

- Why Static Analysis?
- Ada
- C/C++
- Python
- Future Activities

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Why Static Analysis?



© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Why Static Analysis?

	Metrics Structure Analysis	Guidelines / Standards	Bugs Finding (alternative to testing?)	"Compilation"
Ada	M	M	L	
C/C++	L	M	M	
Python	L	M	M	L

M: more important
L: less important

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Metrics, Maintainability Guidelines, Standards

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Some Ada Analyzers used @ Spazio IT

	Metrics Structure Analysis	Guidelines / Standards	Bugs Finding (Abstract Interpretation, Symbolic Execution)
AdaControl	**M**	**M**	
GNAT tools	**M** GNATtmetric	**M** GNATcheck	**A** CodePeer
PolySpace for Ada	**M**	**M**	**A**
Understand	**M**	**M**	

M: mostly used for
A: also used for

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Ada Static Analysis

- Lots of the issues that can be found in a system written in C /C++ are prevented from happening by design (of Ada language itself); e.g. many data conversion errors are prevented from happening by Ada strong typing.
- The compilation system is “very strict”.
- In many Ada projects thorough testing is mandatory.
- Testing may include the execution of run-time checks.

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Metrics to improve Maintainability

- In 2015 Spazio IT developed for AIRBUS an Ada SonarQube Plugin able to compute the so called “Maintainability Index” as a function(Analysability, Changeability, Stability, Testability).

- In turn these other characteristics were based on metrics like (LOCs, Cyclomatic Complexity, Comment/Code Ratio, etc…).

- The overall computation followed a model implemented by AIRBUS and very close to ISO/IEC 9126, ISO/IEC 25010.

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

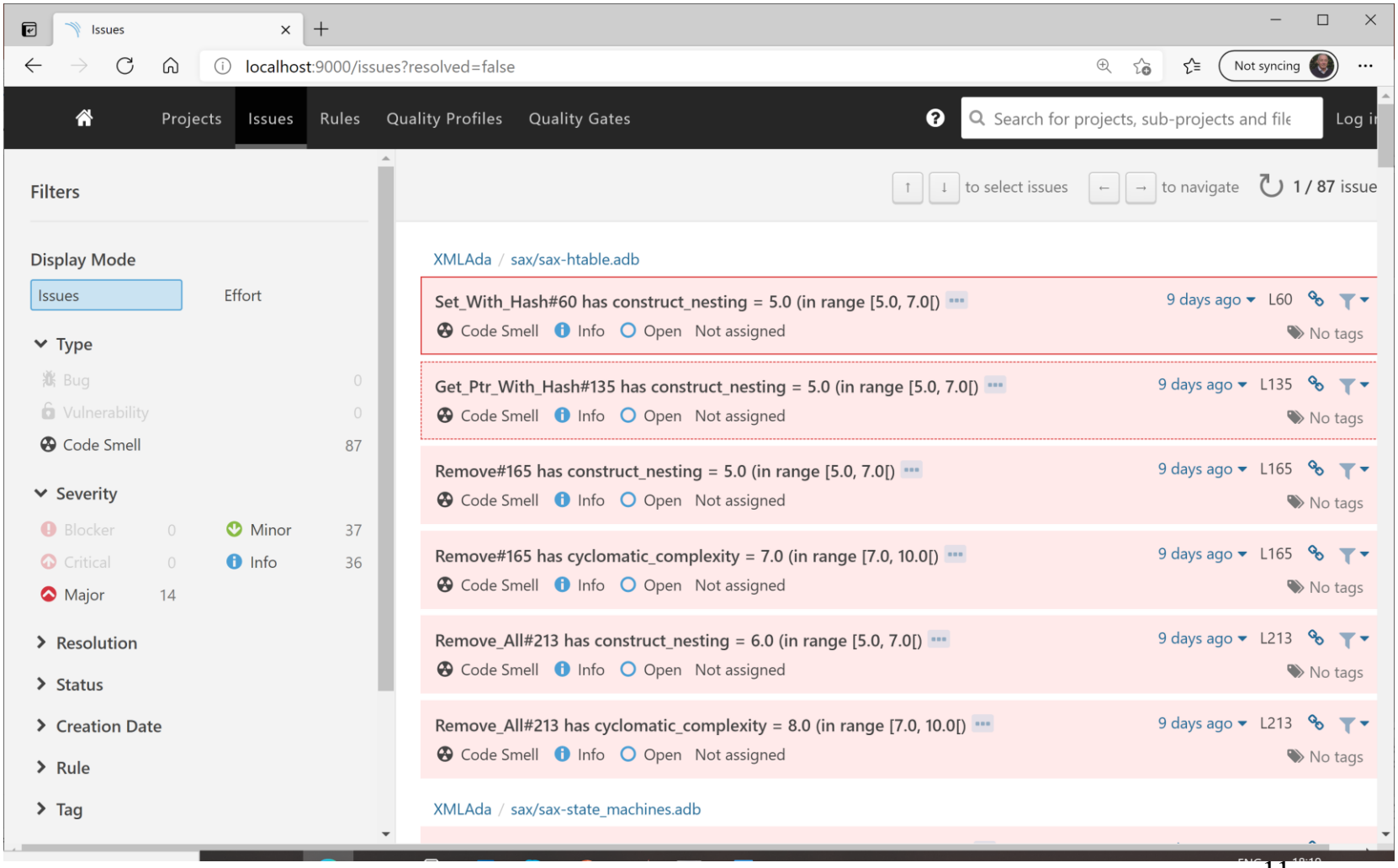
Metrics to improve Maintainability (cont)

- The **maintainability index** proved to correspond well with the «experienced» actual maintainability of real case projects.

- The **issues** found by the tool identified the elements, the points requiring a fix to improve the overall maintainability of the analysed projects.

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Metrics to improve Maintainability (cont)



© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Bugs Finding
Guidelines, Standards

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Some C/C++ Analyzers used @ Spazio IT

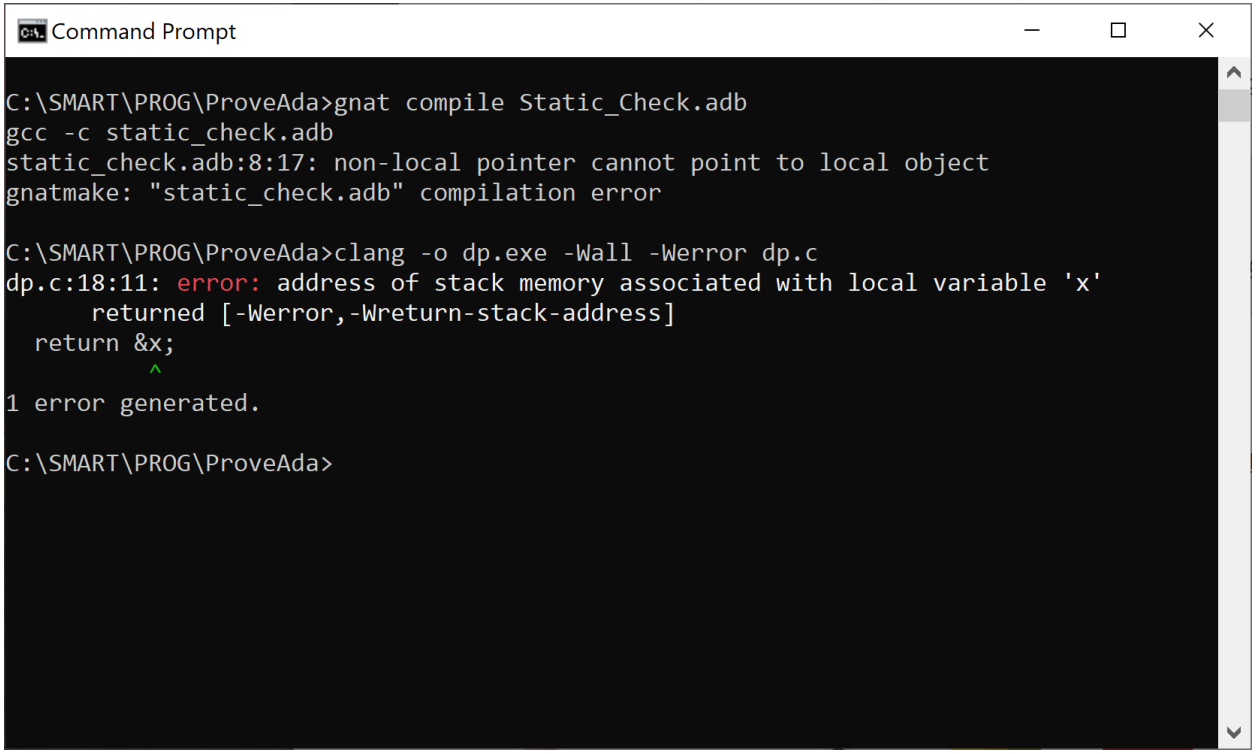
	Metrics Structure Analysis	Guidelines / Standards	Bugs Finding (Abstract Interpretation, Symbolic Execution)
PC-Lint-Plus	**M**	**M**	**A**
Cppcheck	**A**	**A**	**M**
Clang-SA Clang-Tidy			**M**
FB-Infer			**M**

M: mostly used for
A: also used for

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

C/C++ Static Analysis

- C/C++ compilers, if properly used, can be as strict as Ada compilers (e.g. https://www.adacore.com/gems/gem-33)



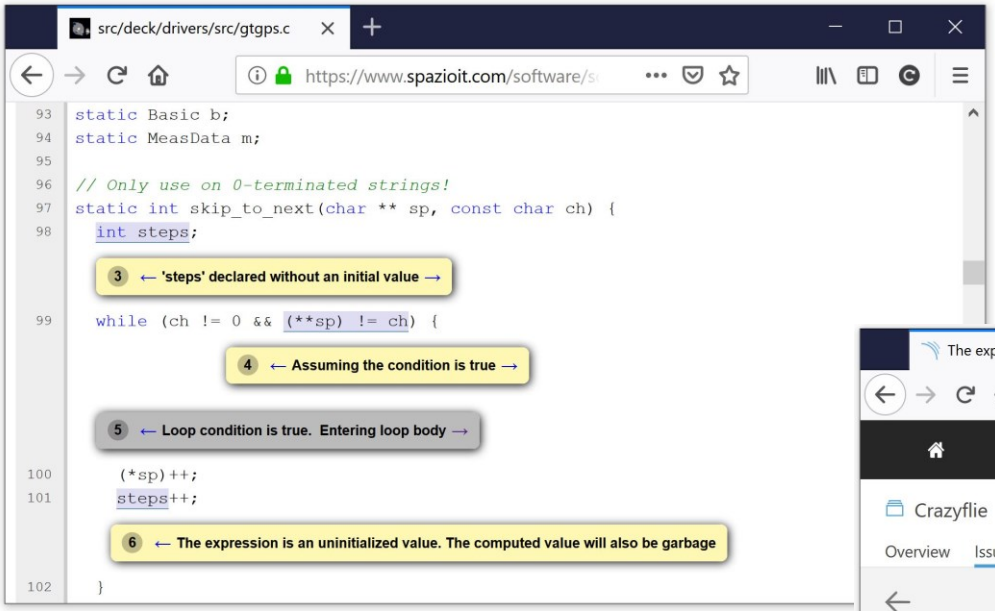
© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

C/C++ Static Analysis (cont)

- In some C/C++ projects thorough testing is mandatory.
- In the Clang compilation system the compiler and the static analyzers (Clang-SA and Clang-Tidy) are based on the very same libraries → **the Compiler is the Static Analyzer**
- GCC analysis capabilities are continuously evolving (also thanks to the competition with "Clang").
- Abstract Interpretation / Symbolic Execution in Clang static analyzers and FB Infer are already rather powerful and improving…

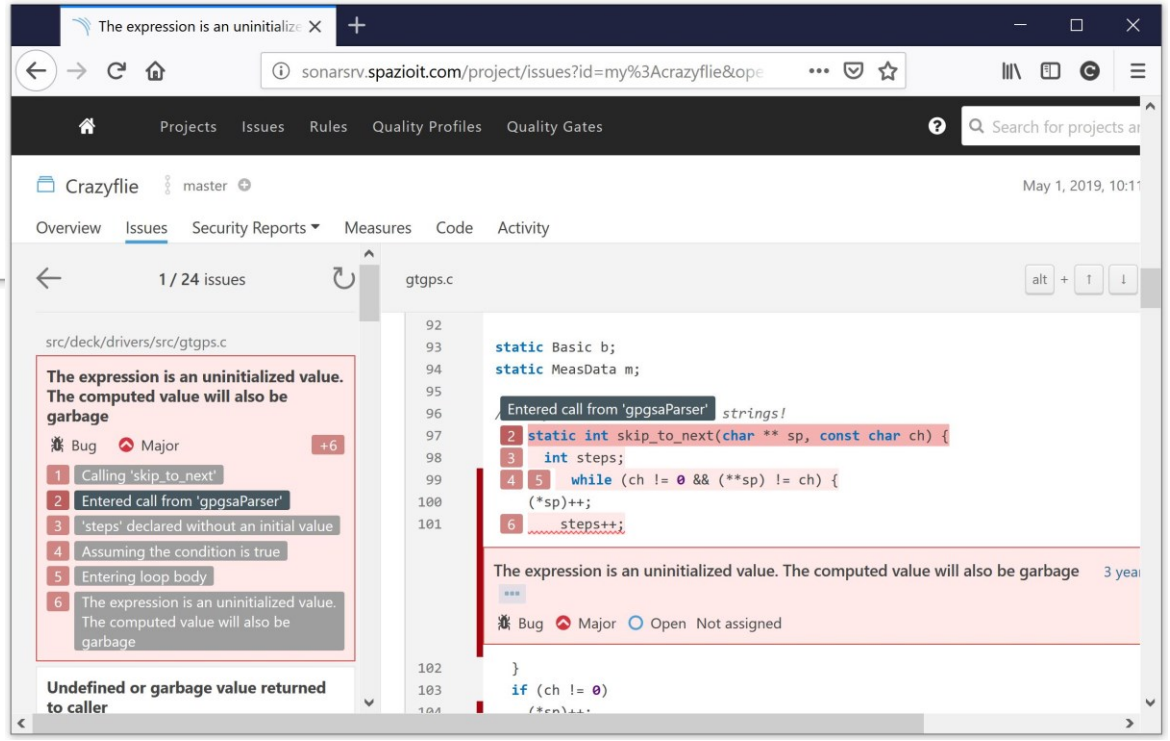
© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

C/C++ Static Analysis (cont)



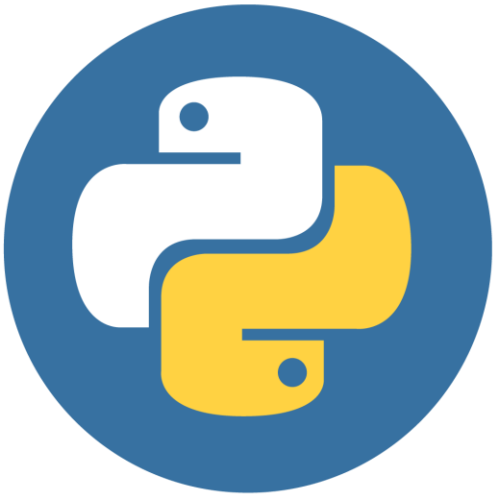
Clang Bugs Finding displayed in Clang UI

Clang Bugs Finding displayed in SonarQube



© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Python



Bugs Finding
Guidelines, Standards
Compilation

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Some Python Analyzers used @ Spazio IT

	Metrics Structure Analysis	Guidelines / Standards	Bugs Finding (Logical Errors, Security Issues, Abstract Interpretation)	Compilation
Bandit	A	A	M	
Flake8	M	M		
Pylint	A	A	M	A
SQ Python Plugin	A	A	M	

M: mostly used for
A: also used for

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Is Python interpreted or compiled?

Help int_comp.py - provep - Visual Studio Code

int_comp.py 1

int_comp.py

```
1  """
2  Is Python interpreted or compiled?
3  """
4  a = input("Should I make a mistake? [yes/no] --> ")
5  if a == 'yes':
6      pront("I'm here, and I'm trying to use something wrong!")
7  else:
8      print("I'm there and everything is OK.")
9  print("What a program!")
10
```

Anaconda Prompt (Anaconda3)

```
(base) C:\SMART\PROG\provep>python int_comp.py
Should I make a mistake? [yes/no] --> yes
Traceback (most recent call last):
  File "int_comp.py", line 6, in <module>
    pront("I'm here, and I'm trying to use something wrong!")
NameError: name 'pront' is not defined

(base) C:\SMART\PROG\provep>pylint int_comp.py
************* Module int_comp
int_comp.py:6:4: E0602: Undefined variable 'pront' (undefined-variable)
```

```
>&2 echo "Migrations complete."

python -u manage.py runserver 0.0.0.0:8080
```


© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

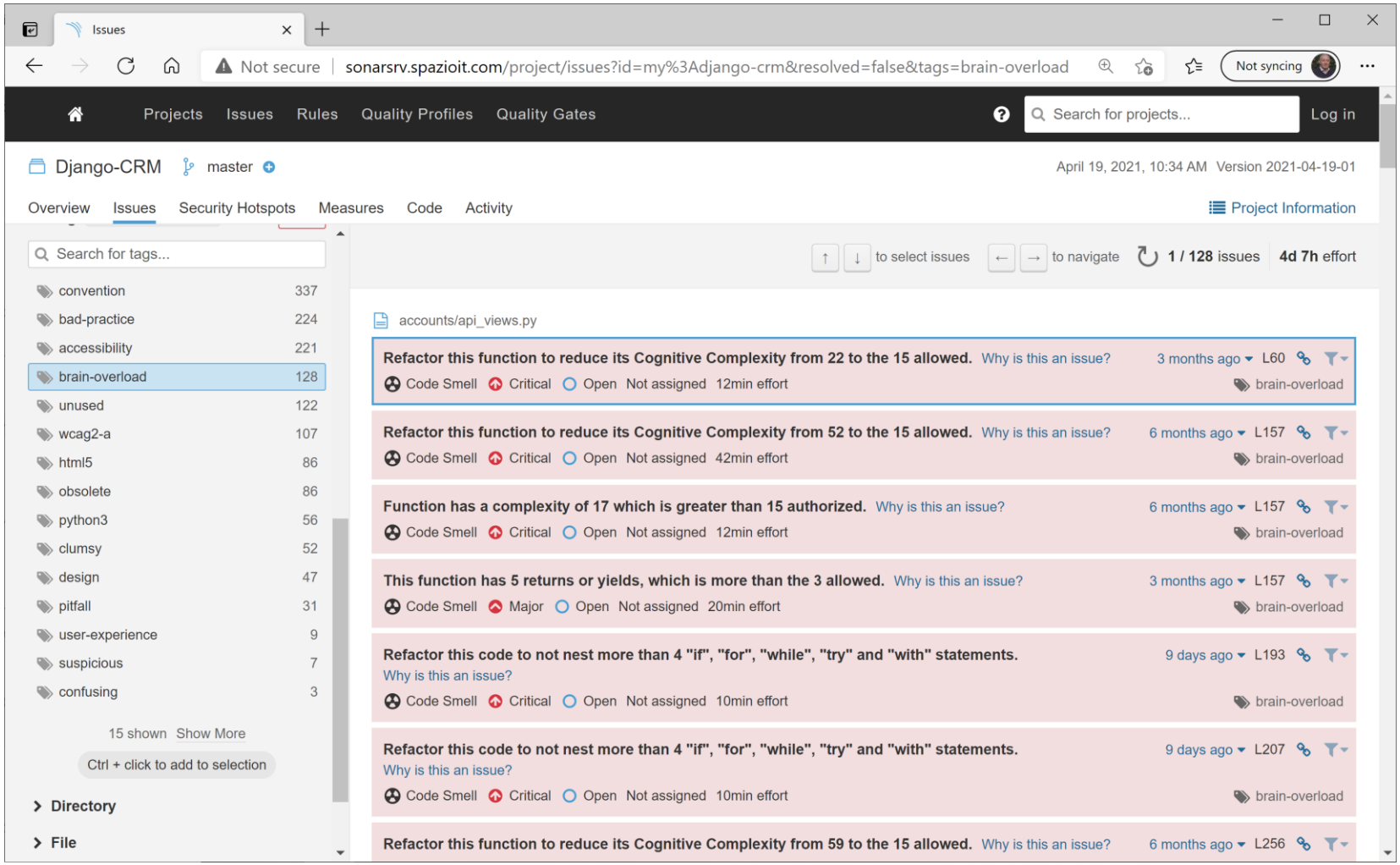
Python Static Analysis

- In case of very large codebases, “linting” can be used also as “compiler”, to guarantee the syntactical correctness of the entire codebase.

- In many Python (not TDD)-projects thorough testing is **not** mandatory.

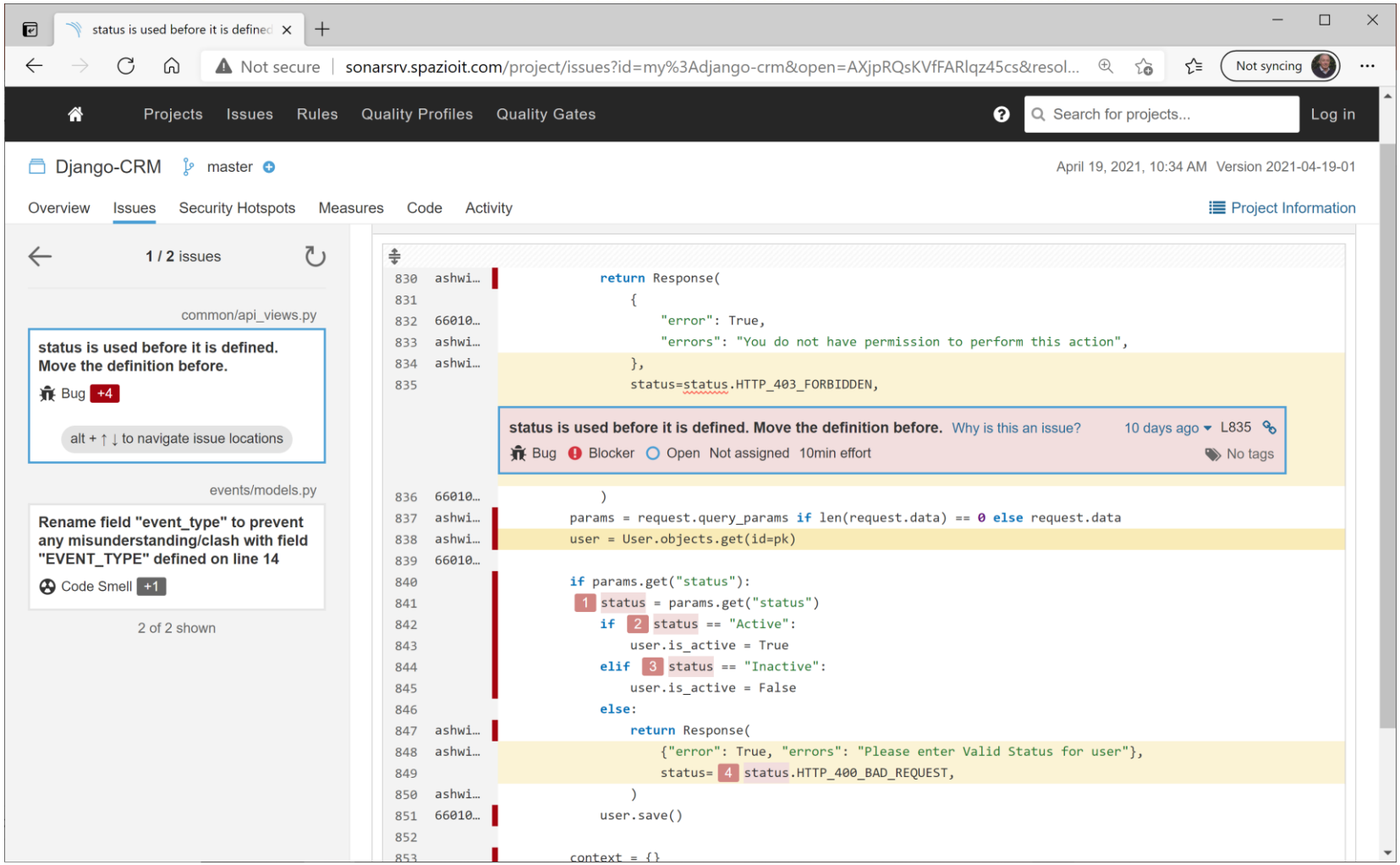
© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Python Static Analysis (cont)



© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Python Static Analysis (cont)



© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Future/Current Activities

sonar qube

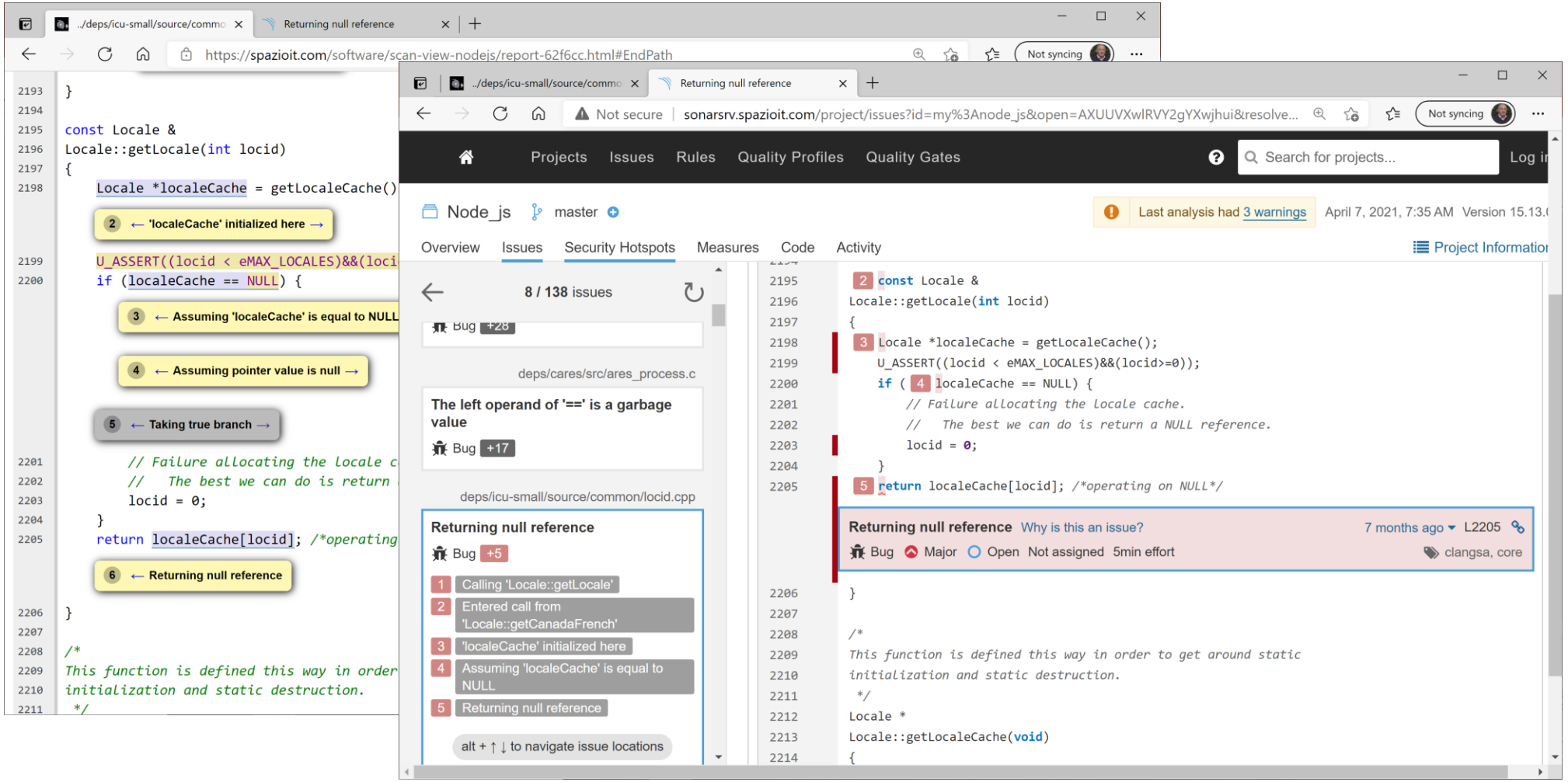
© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Future/Current Activities (cont)

- Keeping up to date with the evolution of Clang-SA, Clang-Tidy and FB Infer.
- Improving their integration with SonarQube.
- Applying these technologies and tools to the SAFe Toolset (https://spazioit.com/pages_en/sol_inf_en/code_quality_en/safe-toolset-en/) and ISVV projects where Spazio IT is going to be involved this year and the next ones. (e.g. adding Python support to the SAFe Toolset).

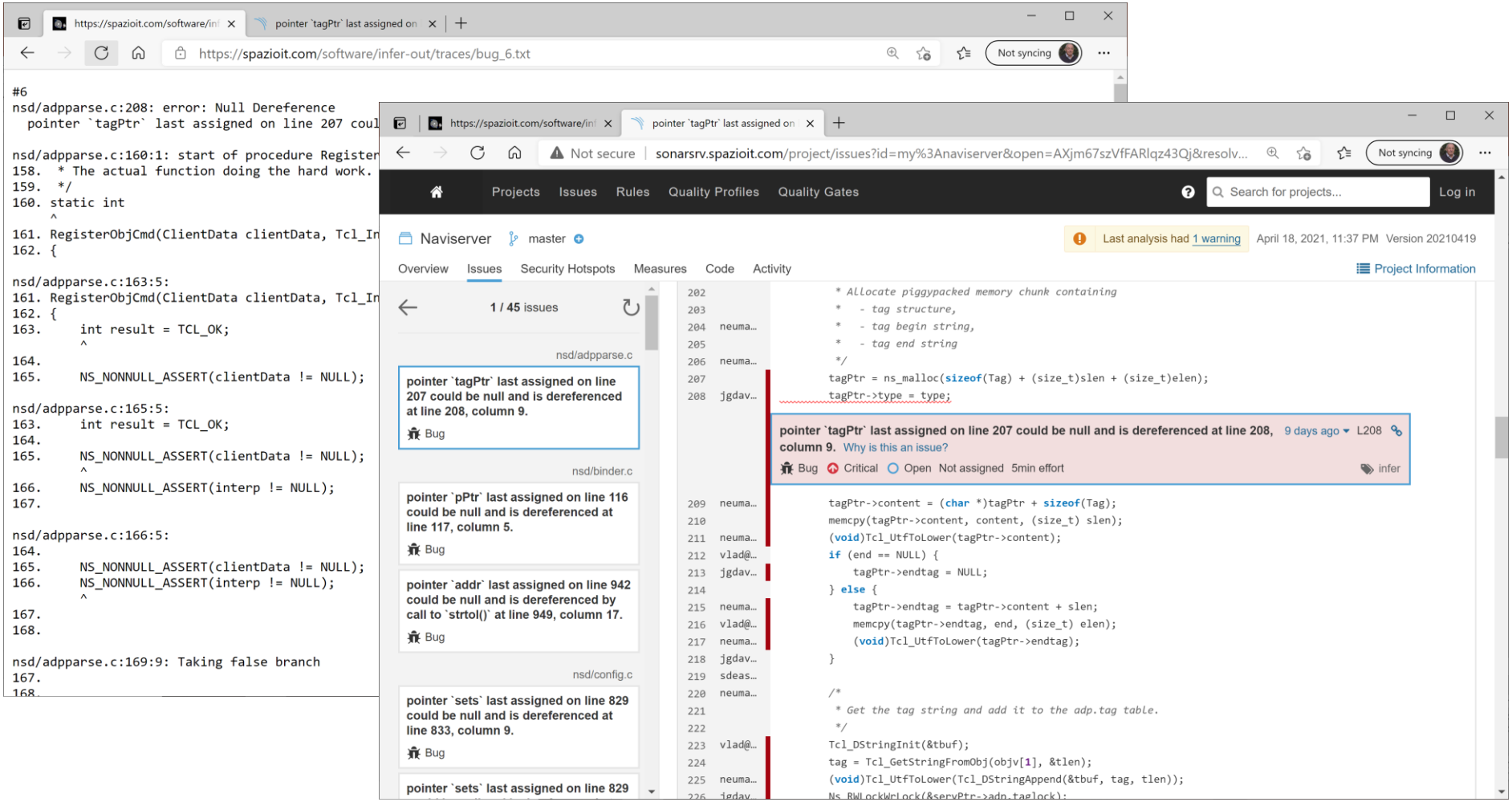
© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Current (04/2021) Clang SA – SonarQube Integration



© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Current (04/2021) FB Infer – SonarQube Integration



© 2021 Spazio IT - Soluzioni Informatiche s.a.s.

Thank you for your time!

© 2021 Spazio IT - Soluzioni Informatiche s.a.s.