API Sanity Checker:

An automatic generator of basic unit tests for a shared C/C++ library

Prof.: Breno Miranda

Student: Victor Augusto Medeiros Balbino

Centro de Informática UFPE

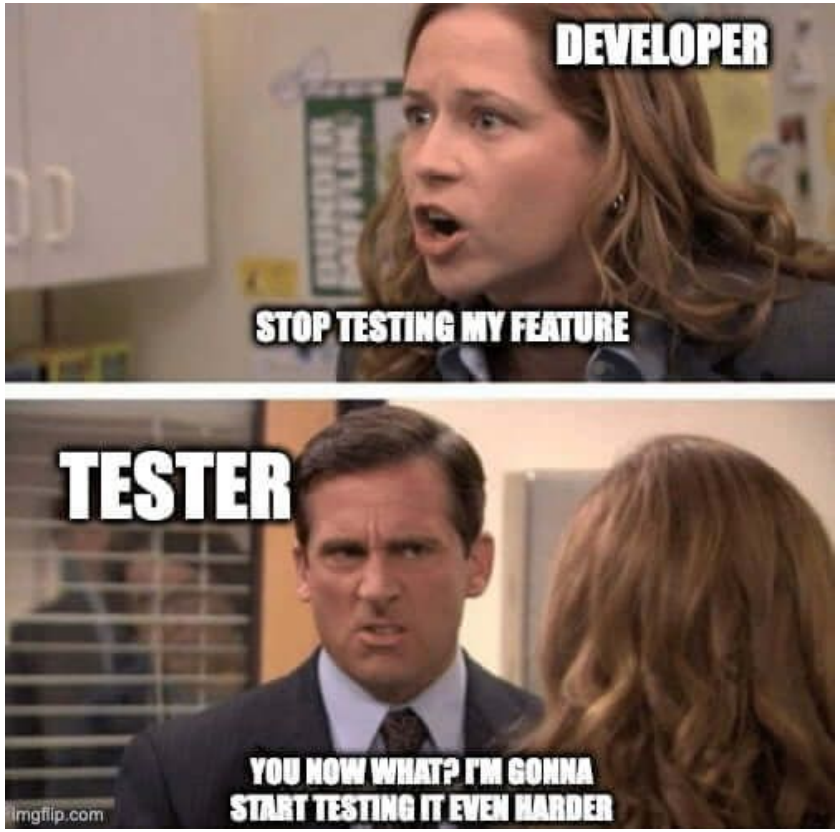
Universidade Federal de Pernambuco

STELLANTIS

FACEPE
Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco

Tool Category

- Category: Automatic generator Tests cases
  - Focus on "Sanity Testing" (Quick Verification Tests)
  - Alternative for initial detection of critical errors



How It Works?

- Analyzes API headers (.h files)
- Automatically generates test cases
- Compiles and runs the tests
- Detects failures such as Segmentation Fault, error codes, and unexpected signals

Positive Aspects

- Fast generation of unit tests
- Detects critical initialization and execution errors
- Can be used as a base for advanced testing
- Supports both C and C++

Negative Aspects

- Superficial tests, without business logic
- May generate false positives and false negatives
- Requires manual configuration of the XML file

Tutorial Created

- Available on GitHub (add repository link)
- Example used: libsample (fictional library for testing)
- Difficulty: Intermediate (requires familiarity with C/C++ and compilation)

descriptor.xml
~/Documentos/api-sanity-checker

```
<version>
    1.0
</version>

<headers>
    /home/victor/Documentos/api-sanity-checker/Tutorial-API_SC
</headers>

<libs>
    /home/victor/Documentos/api-sanity-checker/Tutorial-API_SC
</libs>
```

Example Test Output

Test for soma function

soma (int *a*, int *b*)

Info

Header File	libsample.h
Function	soma
Return Type	int
Parameters	2

Parameters

#	Name	Type
0	*a*	int
1	*b*	int

Code

```
1  #include <libsample.h>
2  int main(int argc, char *argv[])
3  {
4      soma(1, 2); //target call
5      return 0;
6  }
```

Test results for the NAME-1.0 library on x86_64

Summary

Total tests	4
Passed / Failed tests	3 / 1
Verdict	Test Failed

Problem Summary

Received signal SEGV	1

Failed Tests (1)

[−] **Received signal SEGV** (1 problem)

libsample.h, libsample.so
[−] **deve_falhar** ( )

received signal SEGV

Segmentation fault (core dumped)

```
1  #include <libsample.h>
2  int main(int argc, char *argv[])
3  {
4      deve_falhar(); //target call
5      return 0;
6  }
```

Conclusion

- Useful tool for initial API validation
- Complements but does not replace functional tests
- Recommended for quick error detection

OBRIGADO!

Centro de Informática UFPE

Universidade Federal de Pernambuco

STELLANTIS

FACEPE
Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco