2023.11.27 量产代码静态扫描接入ipipe方案调研

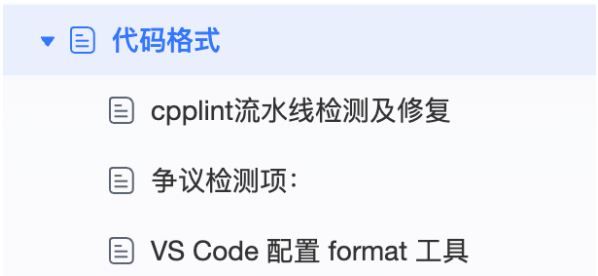
目录

一、背景

量产代码规范扫描和整改，以规范代码。集度的目标是3月底整改完成，计划由一名量产代码规范的QA来筛选一轮量产必须follow的规则，补充到ANP3编码规范中，并且加到ipipe中，不依赖parasoft这样的商业软件。

规范大致分为代码格式类的、bug和隐患类的。已有的资料包括，

1. ANP C++编码风格，其基于Google C++ Style整理了ANP的规范；
2. 代码格式，基于开源的cpplint工具，检测ANP的代码规范问题；

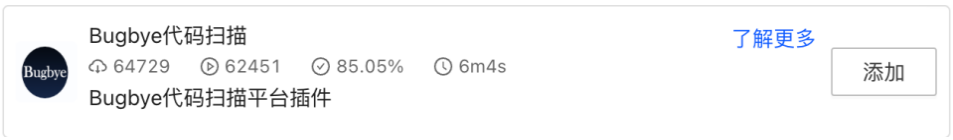


3. 代码质量，给了一些good/bad代码隐患和bug的示例；

二、工具调研

2.1 默认：厂内bugbye

介绍



BCA-Cpp（默认）：**Bugbye团队自研扫描引擎。**不需编译，支持C和C++，不检查语法错误，而是作为编译器的补充，**检测边界溢出、内存泄漏、空指针**等多种类型的代码缺陷。

参考：

1. Bugbye用户手册
2. https://cloud.baidu-int.com/icloud/ipipe/%E6%93%8D%E4%BD%9C%E6%8C%87%E5%8D%97/%E6%8F%92%E4%BB%B6%E4%BD%BF%E7%94%A8/%E6%B5%8B%E8%AF%95%E6%8F%92%E4%BB%B6/Bugbye%E6%8F%92%E4%BB%B6%E6%9B%B4%E6%96%B0%E8%AF%B4%E6%98%8E/

规则集合

https://bugbye.baidu.com/rules

扫描结果

：https://bugbye.baidu.com/taskinfo/33151608，包括代码缺陷、重复代码块、可维护性三个方面。

2.2 开源cpplint.py：偏代码规范

介绍

`cpplint.py` 是一个用于检查 C++ 代码风格的工具，它可以帮助开发者确保代码遵循特定的编码规范和样式指南。它是由 Google 开发的，并且是开源的。

`cpplint.py` 可以检查代码中的格式问题、潜在的错误、注释规范等方面，并提供相应的警告和建议。它支持多种 C++ 编译器和代码风格，例如 Google's C++ Style Guide、LLVM coding standard 等。

要使用 `cpplint.py`，您需要在终端中运行它，并提供要检查的源代码文件作为参数。例如，要检查名为 `my_source.cpp` 的文件，您可以在终端中运行以下命令：

```bash
cpplint.py my_source.cpp
```

这将输出有关代码问题的警告和错误信息。您还可以将多个文件名作为参数传递给 `cpplint.py`，以便一次检查多个文件。

除了命令行界面，`cpplint.py` 还提供了 Python 模块接口，您可以在 Python 脚本中使用它来自动检查代码风格。

总之，`cpplint.py` 是一个非常有用的工具，可以帮助您确保 C++ 代码遵循特定的编码规范和样式指南，并提高代码质量和可维护性。

共69条规则，支持自定义规则，如下使用65条规则全量扫描andes代码库。

扫描参数

```bash
python2 cpplint.py --linelength=120 --extensions=c,cc,cpp,hpp,h --filter=-legal/copyright,-build/include_subdir,-build/printf_format,-runtime/references --exclude=modules/canbus/vehicle --recursive
```

扫描结果

cpplint_andes.txt
10.3MB

梳理&分类后如下表，

cpplint_andes.xlsx
14.8KB

2.3 cpp-check

介绍

官网是https://cppcheck.sourceforge.io/，代码静态扫描工具，用于在 C 和 C++ 代码中查找编程错误，未使用的代码和性能问题。 主要关注的是：

1. 代码质量问题：它可以检测到许多常见的代码质量问题，如空指针引用，数组越界等。
2. 性能问题：它可以检测到可能会导致性能问题的代码，如不必要的内存分配，不必要的拷贝等。
3. 未使用的代码：它可以检测到未使用的函数，变量和类等。
4. 错误的注释：它可以检测到错误的注释，这些注释可能会误导其他程序员。

它提供免费、付费两种方案，共245项规则，https://sourceforge.net/p/cppcheck/wiki/ListOfChecks/

Coding standards

Coding standard	Open Source	Premium
Misra C 2012 - original rules	Yes	Yes
Misra C 2012 - amendment #1	Yes	Yes
Misra C 2012 - amendment #2	Yes	Yes
Misra C 2012 - amendment #3		Yes
Misra C 2012 - amendment #4		Yes
Misra C 2012 - Compliance report		Yes
Misra C 2012 - Rule texts	User provided	Yes
Misra C 2023		Yes
Misra C++ 2008		Partial
Cert C		Yes
Cert C++		Yes
Autosar		Partial

陈潜做的调研 静态代码扫描工具调研 和 SA静态代码扫描 显示，

> 从报错数量和准确率来看
>
> 有效数量：TSC[293]>coverity[164]>clang[142] >cppcheck [120]>pclint[116]
>
> 准确率：clang[97%] >TSC[93%]>coverity(88%)>pclint[72%] >cppcheck[55%]
>
> 综合评分：coverity[94分] > TSC[86分] > clang[80分] >cppcheck[63分] >pclint[27分]

> 存在：指针判空通用化规则、存在风险类型或风险操作、非void函数最后需要显式return等规则的误报较多的问题。

2.4 PVS-Studio

PVS-Studio 是一款静态分析软件，用于诊断C/C++/C#应用程序源代码中的错误。它适用于Windows，Linux和macOS环境，包含3套诊断规则：64位错误诊断规则（Viva64）、平行错误诊断规则（VivaMP）和通用诊断规则。相对于其他类型的方法而言，其采用的静态代码分析方法有明显的优越性，因为它可以覆盖整个程序代码。代码检查的过程在任何情况下都不会破坏代码本身。分析过程完全由程序员控制，并决定是否需要修改代码。PVS-Studio工具是俄罗斯"Program Verification Systems"公司自主开发。

有比较详细的规则解释和解决示例，https://pvs-studio.com/en/docs/warnings/v2598/

三、ANP3编码规范接入ipipe