Hisa 静态分析工具需求说明

(第一周期)

撰写者：南京大学 程珂

南京大学 陈昊东

南京大学 施超烜

南京大学 吴晓阳

南京大学 周若衡

目录

1 引言

在当下信息时代，软件产品日益增多，发展日新月异，代码是软件产品中的主要构成部分。而在程序员编码的过程中总会出现一些隐含的问题，这些问题在测试的过程中可能不会被察觉，给系统的安全性和健壮性留下潜在的隐患。程序静态分析是与程序动态分析相对应的代码分析技术，它通过对代码的自动扫描发现隐含的程序问题， 在不实际执行程序的情况下对代码进行分析，通过在真实或模拟环境中执行程序进行分析的方法，多用于性能测试、功能测试、内存泄漏测试等方面。与之相反，静态分析不运行代码只是通过对代码的静态扫描对程序进行分析。静态分析的执行速度快、效率高。目前成熟的代码静态分析工具每秒可扫描上万行代码，相对于动态分析，具有检测速度快、效率高的特点。静态分析的误报率较高。它通过对程序扫描找到匹配某种规则模式的代码从而发现代码中存在的问题，例如可以定位 strcpy（）这样可能存在漏洞的函数，这样有时会造成将一些正确代码定位为缺陷的问题，因此静态分析有时存在误报率较高的缺陷，但是可结合动态分析方法进行修正。

这篇文档主要介绍了静态分析工具的综合描述、系统功能需求、非功能性需求等几个问题，分别会在一下几章中为读者——介绍。

1.1 编写目的

为明确软件需求、安排项目规划与进度、组织软件开发与测试，以及保证软件开发的质量、需求的完整与可追溯性，编写此文档。

通过此文档，以保证业务需求提出者与需求分析人员、开发人员、测试人员及其也相关利益人对需求达成共识。

1.2 项目背景

静态分析：指在不运行代码的方式下，通过词法分析、语法分析、控制流、数据流分析等技术对程序代码进行扫描，验证代码是否满足规范性、安全性、可靠性、可维护性等指标的一种代码分析技术。

该项目采用迭代开发的方式，要求针对 C/C++语言中的一些常见问题，实现一个静态分析检测工具，支持对常见问题的分析定位

1.3 预期的读者和阅读建议

本软件产品需求分析报告针对的各种不同的预期读者，包括:

- 用户：建议阅读本文档的第二章，第三章第 1，2 节，以了解软件的综合信息，具体功能，及配置环境，运行时的要求和限制等。
- 开发人员：建议阅读文档的第二、三、四章，以了解整个软件的系统结构，需完成的功能需求和非功能性需求，用户界面等，更加完善，标准的完成软件的开发。
- 项目经理：建议阅读文档第二章、第三章、第四章，以了解软件的综合信息，具体功能，以及用户的非功能性需求以便掌控项目进度，同时方便模型设计。
- 营销人员：建议阅读本文档的第二章(项目概述)，以了解产品的状况，功能、特性及运行环境等，来更好的向用户介绍产品的相关信息。
- 测试人员：建议阅读文档的第三、四章，了解并测试软件是否完成所要需的要求，是否有漏洞需要调试。
- 文档编写入员：建议阅读全篇文档，对软件有综合，系统的认识，能更好的编写相应的文档，使阅读者能更加透彻的了解该软件的运行、维护、更新、编写方式、系统功能等方面。

2 项目概述

2.1 项目前景

近年来，开源软件在操作系统，编程框架，应用软件等方面得到了广泛应用。如 Github，开源中国等软件项目托管平台，以及 Amazon，Oracle 等大型互联网公司都提供了大量的开源软件，以供各大公司使用或二次开发。据统计，80%以上的世界财富 500 强企业投入使用了大量开源软件，并且已经或正在部署开源软件安全检测系统。而软件测试技术对于某些潜在缺陷可能无法排除，因此静态检测技术作为在代码运行之前对代码进行分析的重要方法，对于程序缺陷检测具有重要价值。

2.2 项目功能概述

本次项目主要提供了用户提交项目文件，工具自动分析并给出用户分析报告的功能。用户注册和登入后，上传 C++项目文件、分析工具对**内存泄露**、**内存重复释放**、**使用结构体中未初始化成员**，**整数溢出**，**空指针解引用**，**除 0 模 0**，**继承类指针数组问题**等程序缺陷采用语法分析、语义分析、数据流分析、别名分析等分析方式给出缺陷报告，描述检测到的缺陷类型和其在源代码中的具体位置。

2.3 用户类及其特征

用户：提交项目文件，接受项目缺陷分析报告。

2.4 运行环境

代码运行在 Ubuntu 桌面系统（16.04 版本以上）或 Window10 操作系统上

使用 C++语言进行编写

采用 LLVM+clang（9.0.0 版本）进行编译

使用 Github 进行项目管理

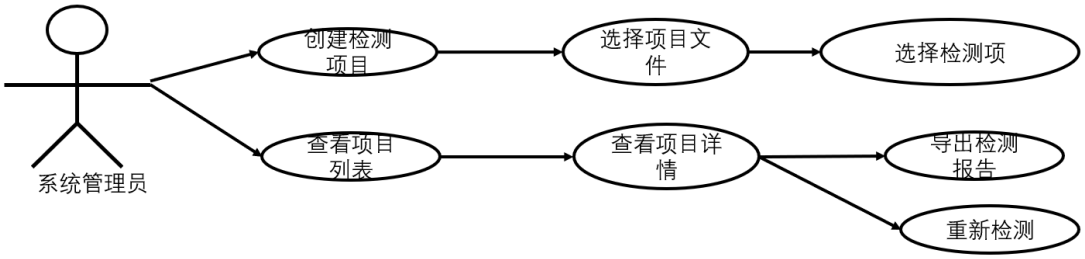
使用 clang-format 进行编码规范

3 功能性需求

普通功能性需求：上传项目文件；选择需要检测的缺陷；查看自己之前检测过的项目；项目重新检测；提交新项目；缺陷检测；导出项目缺陷检测报告等功能。要求对于每一种缺陷的误报率和漏报率都控制在 30%以内。

3.1 功能划分

普通用户用例图模型如下：



如上图所示，用户使用的主要功能有功能，上传项目文件，选择需要开始检测的缺陷对项目进行检测，查看项目详情导出项目检测报告和重新检测的功能。

3.2 功能描述

在此使用事件列表的方式来进行功能描述：

静态分析工具项目事件列表

事件	触发	事件源	行为	响应	事件宿
用户创建检测项目	用户打从命令行输入项目检测命令并输入项目基本信息	用户	在用户的项目列表中添加一个项目，并在其中保存用户输入的基本信息	用户项目列表中的项目	用户
用户上传项目文件	用户从命令行输入项目文件所在地址	用户	Load 项目文件，并在数据库中插入相应信息	上传成功或失败的结果信息	用户
用户选择检测项	用户选择命令行中输入需要检测的检测项,默认为全选	用户	记录用户的选择，并重新展示给用户	展示用户的选项	用户
用户开始检测	用户输入开始检测命令	用户	将检测项信息和项目文件地址传给程序的 Processor 组件进行预处理，展示缺陷检测的进度	缺陷检测的进度	用户
用户查看自己的项目列表	用户输入查看项目列表的命令	用户	从数据库中根据用户账号信息读取用户的项目列表，整理成项目列表展示给用户	用户项目列表	用户
用户查看项	用户输入查看项	用户	从数据库中读取项目详	项目详情信息	用户

目详情	目详情命令		情数据（项目检测进度，项目操作历史等项目数据）整理成项目详情信息，发送给用户		
用户导出检测报告	用户输入导出项目缺陷检测报告命令和本地存储位置	用户	判断项目检测报告是否已经完成。然后从后端数据库中的读取项目缺陷检测报告所在地址，根据地址将相应缺陷加测报告保存	导出成功或失败的消息和缺陷检测报告	用户
用户重新检测	用户输入重新检测的命令，提交项目文件重新并选择需要检测的缺陷	用户	将检测项信息和项目文件地址发送给程序的 Processor 组件进行预处理，展示缺陷检测的进度	缺陷检测的进度	用户

4 非功能性需求

4.1 性能需求

性能：软件并发用户数大于 20 人，点击响应时间小于 2 秒，用户报告生成时间取决于项目的规模和用户选择的需要检测的缺陷类型，10w 行规模的代码预计在可以在 10min 内完成

项目报告的生成。

4.2 安全性需求

要求系统无已知漏洞且能够阻挡一定的网络攻击，防止用户个人信息，项目代码和缺陷报告泄露。

4.3 软件质量属性

1.正确性：在支持在当下主流浏览器环境下能正确地完成软件静态分析的功能；

2.健壮性：无已知系统漏洞，在硬件发生故障、能够对错误输入做出正确的反应和处理；

3.效率：根据项目规模的不同，工具可以在有限的时间内生成报告，报告生成的时间取决于于项目规模的大小和选择的缺陷种类，10w 行以内的项目 10min 以内生成缺陷检测报告，

4.可用性：能够在有限的时间内完成缺陷报告的生成；

5.可理解性：软件功能设计合理，易于操作使用，用户可快速掌握软件操作；

6.可维修性：可以在短时间内改正在运行发现的错误；

7.灵活性： 支持通过二次开发动态增加功能模块

8.可测试性：软件功能易于测试；

4.4 其他非功能性需求

软件是单机形式的命令行软件

采用 C++开发，采用 LLVM+clang（9.0.0 版本）进行编译，使用 Github 进行项目管理，使用 clang-format 进行编码规范。