



编码实现过程定义

江西博微新技术有限公司

(版权所有，翻版必究)

变 更 记 录

版本	修改条款	修改内容	修改人/日期	批准人/日期
1.0.0	初版		刘献/20120329	
1.0.1	增加内容	增加 5.1.6 编译联调” 章节	刘献/20120410	
1.0	试用版	评审通过	刘献/20120601	何贺/20120601
1.1.0	删除内容	删除 7、8、9 章节的参考文献、裁剪、度量、评审等	刘献/20120714	
1.1.1	修改内容	修改 5.1.7 说明模块检测记录提交时机	刘献/20120724	
1.1.2	修改内容	修改 5.1.8 代码抽查的频率和提交时机	刘献/20120724	
1.1.3	修改内容	修改 5.1.7 模块测试时机说明	刘献/20120725	
1.1.4	修改内容	修改 5.1.4 将进度计划的具体要求放置到项目策划过程域做统一说明	刘献/20120729	
1.1.5	增加内容	增加 5.1.6 的 C++ 静态代码检查工具	刘献/20120729	
1.1	正式版	评审通过	刘献/20120730	何贺/20120730
1.2.0	增加内容	在 5.1.7、5.1.10、5.1.11、6 节中增加内存检查相关内容,与测试过程定义中集成测试入口准则对应; 在 5.1.8 节中规范 checkStyle、findbugs 检查结果填写,对代码抽查流程细化。	陈昌艾/20131108	
1.3	正式版	评审通过	熊蔡鹏/20132230	何贺/20131202
1.4	修改内容	调整对代码抽查的描述,整理为每周对已完成的功能进行抽查	熊蔡鹏/20140804	
1.5	修改内容	【活动 1】: 增加在制定项目编码计划阶段,需注意明确代码抽查时间节点。 【活动 5】: 编码阶段结束前,开发组长需运用代码检查工具对项目代码进行整体检查,并对检查结果进行记录,	胡伟/20140910	

		发送至项目经理、部门质量检查人员及 项目组 QA 人员。		

目录

1. 目的	4
2. 适用范围	4
3. 术语定义	4
4. 角色与职责	4
5. 过程描述	5
5.1. 【过程元素 1】编码实现	5
5.1.1. 活动图	5
5.1.2. 入口准则	6
5.1.3. 输入	6
5.1.4. 【活动 1】制定编码实现计划	6
5.1.5. 【活动 2】代码实现	6
5.1.6. 【活动 3】编译联调	6
5.1.7. 【活动 4】执行模块测试	7
5.1.8. 【活动 5】代码抽查	7
5.1.9. 【活动 6】编写部署手册	9
5.1.10. 出口准则	9
5.1.11. 输出	9
6. 工具	9

1. 目的

本文档主要是规范江西博微新技术有限公司研发部的编码实现过程。

2. 适用范围

本规范主要适用于江西博微新技术有限公司研发部软件开发周期中的编码实现阶段。

3. 术语定义

术语	定义
编码计划	对《项目进度计划》中编码实现部分内容的细化。
静态代码分析	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。
静态代码分析工具	具有通用目的和专业化的，用于检测特定类型错误，代替程序员进行静态代码分析的程序。

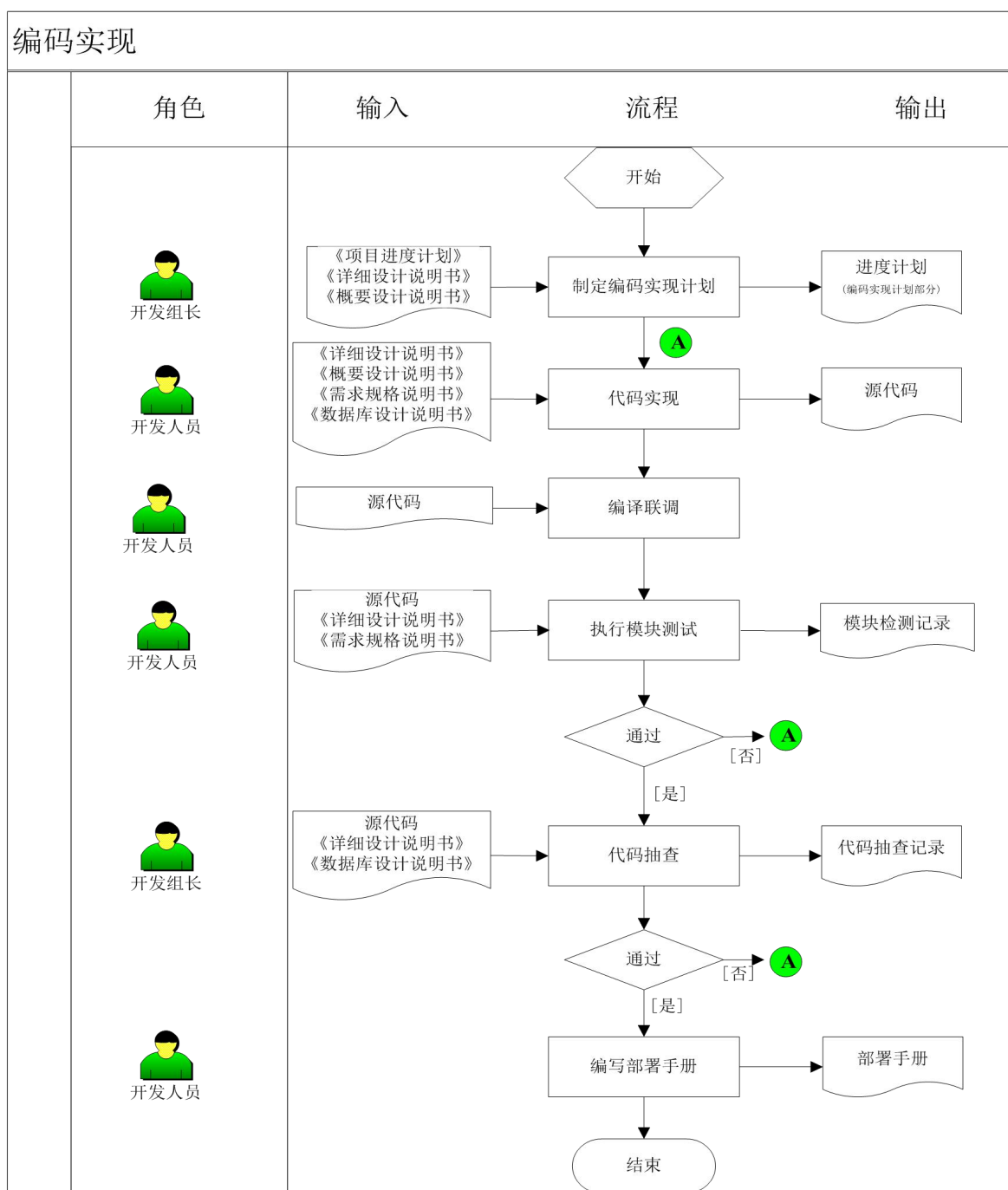
4. 角色与职责

角色	职责
开发组长	负责制定项目组具体的编码实现计划，指导开发人员开展具体的开发工作，并对项目组所编写代码进行代码抽查。
开发人员	依照《详细设计说明书》、《需求规格说明书》等文档进行系统的编码实现，并对所完成内容执行模块测试。

5. 过程描述

5.1. 【过程元素 1】 编码实现

5.1.1. 活动图



5.1.2. 入口准则

通过评审的《需求规格说明书》。

通过评审的《概要设计说明书》、《详细设计说明书》和《数据库设计说明书》。

5.1.3. 输入

《项目进度计划》

《详细设计说明书》

《概要设计说明书》

《需求规格说明书》

《数据库设计说明书》

5.1.4. 【活动 1】制定编码实现计划

开发组长根据项目进度要求、人员情况及项目设计内容制定具体的编码实现计划, 在制定项目编码计划阶段, 需注意明确代码抽查时间节点。

5.1.5. 【活动 2】代码实现

开发人员按照编码实现计划的安排, 依照《详细设计说明书》、《概要设计说明书》、《数据库设计说明书》和《需求规格说明书》, 利用对应的开发工具编写代码, 实现《详细设计说明书》中所设计的功能, 并遵守对应开发语言的代码编写规范。开发组长对代码实现过程中遇到的问题及时给予指导和支持。

5.1.6. 【活动 3】编译联调

开发人员在完成代码实现后, 需要进行以下操作才允许将所编写的代码签入到团队源代码管理工具中进行统一管理:

1. 获取项目所有代码，使用开发工具进行编译，纠正编译器所报的错误问题，还需要处理诸如：已赋值未使用的变量、使用未赋值的变量等警告内容，直至编译通过。
2. 对所编写的源代码执行代码自查，并使用如下静态代码分析工具进行辅助分析：CppCheck(C++)、VS 集成的代码分析工具(.net) ,Check Style(Java) ,主要检查所编写的代码是否符合编码规范。

5.1.7. 【活动 4】执行模块测试

在功能完成之后，开发人员根据《需求规格说明书》和《详细设计说明书》执行模块测试，模块测试主要对照需求检测功能是否正常和正确，并将测试中发现的问题进行修正，记录到《模块检测记录》中。建议项目组内的开发人员对完成的模块进行交互检测，以尽可能多地发现问题。

1. 此外产品在功能完成后对于在模块代码实现后，需要使用 vld 工具（目前使用的工具 vld-2.2.3-setup.exe）检查内存泄露问题，并检查结果记录到《内存检查记录》中。在系统集成测试时，需向测试人员提交所要测试模块的《内存检查记录》。命名如“QR731803003 内存检查记录(项目编号_项目版本号_模块名称_编写者拼音首字母_配置项创建日期).txt”

5.1.8. 【活动 5】代码抽查

开发组长依照代码抽查原则和项目相关的文档对开发人员所实现的代码进行抽查，抽查原则有：

- 1、层次结构及命名：接口、类、方法所处层、包是否划分合理，命名是否规范、是否能反映它们代表的事物

- 2、可读性、可维护性：注释是否充分、是否存在过度设计的代码或冗余代码、是否符合 6 大基本原则
- 3、核心算法：框架代码、模块或系统级接口、业务计算、关键命令功能、关键功能、关键模块、项目组自己定义内容等代码；
- 4、安全：数据是否需要加密，密钥的管理、是否正确使用了日志记录或监控
- 5、其他：流或连接是否有忘记关闭、是否使用了不必要的同步、锁、异常的正确处理等
- 6、系统中的业务流程、关键时序所对应的代码；
- 7、出错最多的模块；
- 8、出错最多的人员；
- 9、新加入人员前三个月内编写的代码；
- 10、项目初期每个人编写的前 2000 行代码；

checkStyle、findbugs 代码检查工具检查的问题分别合并为 checkStyle、findbugs 检查两项，此两项的检查时间不能超过 0.5 小时，且必须检查。开发组长**每周**对本周完成的功能模块，进行代码抽查，将检查了的所有检查点及检查时间记录到《代码抽查记录》，对于有问题的检查点需记录问题，并对发现的问题进行分析，根据具体问题确定处理办法，指导开发人员进行修改，开发人员修改完成之后，由开发组长对抽查发现的问题进行复查，确认所有的问题都处理完毕，记录解决时间及解决耗时。对于没有问题的检查点在《代码抽查记录》问题栏中记录无。

在系统进行功能验证时，需向验证人员提交所要验证模块的《模块检测记录》及《代码抽查记录》。

建议开发组长通过 SVN 代码库历史版本对比,对新增、修改的代码进行抽查。

编码阶段结束前,开发组长需运用代码检查工具(如 checkstyle/findbugs)对项目代码进行整体检查,并对检查结果进行记录,发送至项目经理、部门质量检查人员及项目组 QA 人员。

5.1.9. 【活动 6】编写部署手册

开发人员在系统所有功能全部完成之后,需要编写系统部署手册,说明项目所依赖的中间件、组件、数据库等内容,并以图片的形式说明安装步骤,测试人员将依照该文档进行独立部署,该文档也将移交给售前人员,指导进行项目部署。

5.1.10. 出口准则

通过模块测试并输出《模块检测记录》

通过代码抽查并输出《代码抽查记录》

产品方面,需达到零内存泄露以及零 GDI 泄露并输出《内存检查记录》

5.1.11. 输出

源代码、《模块检测记录》、《代码抽查记录》、《系统部署手册》。

产品方面,要求输出《内存检查记录》。

6. 工具

Microsoft Word

Microsoft Project

Power Project

Microsoft Visual Studio

Eclipse

CppCheck

Check Style

vld