# FastCAE开发规范

青岛数智船海科技有限公司

|  |  |  |  |
| --- | --- | --- | --- |
| **时间** | **修改纲要** | **修改人** | **生效时间** |
| 2019/11/17 | 制定规范 | 李宝君 | 2019/12/01 |
| 2020/04/23 | 4.7添加结构体使用说明，数组使用说明。 | 李宝君 | 2020/05/01 |
| 2020/07/02 | 添加开发过程规范。 | 李宝君 | 2020/07/02 |
| 2020/07/09 | 添加代码段嵌套要求 | 李宝君 | 2020/07/10 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 总则

随着FastCAE开发团队的壮大，团队之间的协作变的越来越频繁，为了保证协作与交接的顺畅以及保证工程管理与代码质量特制定该规范。该规范将对开发过程中的工程管理与代码编写提出要求，所有参与FastCAE开发的人员必须遵守该规范。产品负责人或开发小组负责人有权对本规范进行修改，并根据该规范对开发人员的编码质量进行评定，评定结果与月度/季度考核挂钩。开发人员可向负责人提出合理建议，经批准后对本规范进行修改。

## 开发环境规范

FastCAE要求跨平台运行，对操作系统的要求为Windows 10和Ubuntu 16.04 LTS。要求保证开发环境干净，杜绝流氓软件，尤其不得安装360系列杀毒防护软件。

Windows下要求使用Visual Studio集成开发环境（以下简写VS），MSBuild编译器。要求开发环境预先安装Qt5.4.2 64bit OpenGL版与VS 2013 with update5 社区版。Qt下载地址<http://download.qt.io/archive/qt/5.4/5.4.2/> VS 2013下载地址 <https://my.visualstudio.com/Downloads?q=visual%20studio%202013&wt.mc_id=o~msft~vscom~older-downloads>。Qt安装路径：C:\Qt\Qt5.4.2（默认路径），全部默认安装即可，大约占用硬盘1.5G。VS安装路径：C:\Program Files (x86)\Microsoft Visual Studio 12.0（默认安装路径），VS只安装基本组件即可，选择组件时可以全部不勾选，大约占用硬盘8G。

Ubuntu下要求使用G++编译器，开发环境要求安装与Windows相同版本的Linux 6位Qt。安装位置为 /opt/Qt/Qt5.4.2

FastCAE还用到了很多其他的第三方库，例如OCC和VTK等，这些库由专人负责编译后交付开发人员使用，如果需要功能拓展对其进行重新编译的，或者要求添加其他第三方库的须向负责人说明，统一添加修改，不得私自改动第三方库。

FastCAE在Windows和Ubuntu下都有一键开发环境配置工具，在安装完必要的开发工具（Qt、VS等）之后，具体的编码环境可通过一键开发环境配置工具完成配置，第三方库也会在自动完成配置。

## 代码工程管理规范

FastCAE通过Git进行代码管理与维护，Git使用要求参考附录一《Git使用规范》

FastCAE工程代码通过qmake进行跨平台管理，其中包含多个工程，工程配置采用qmake工程文件\*.pro文件和\*.pri文件。\*.pro文件包含编译选项信息，\*.pri包含源码文件（头文件源文件等）信息。两类文件信息不得混淆编写，为保证准确性，两类文件中不得使用通配符。

为方便开发人员进行开发编译，FastCAE的工程管理提供了多个批处理工具，在对工程修改的过程中，开发人员需要及时更新同步批处理工具，保证其他人员的开发效率。

## 代码规范

FastCAE主要采用C++语言编写，C/Python作为辅助有少量的使用，因此，代码规范主要针对C++编码提出规范性要求。

1. 为保证程序的模块化，头文件中包含尽量少的包含其他头文件，尤其是自己定义的头文件，尽量采用前置声明的方式。
2. 遵循MVC原则，尤其必须遵守数据与视图分离的原则，两者不得混淆编写。
3. 有效代码中包括原始Ui文件中不得出现中文字符。
4. 成员变量在声明的时候就要进行初始化，静态变量除外。
5. 习惯使用命名空间，且尽量不要使用using namespace XXX。
6. 所有的成员变量必须通过成员函数访问，不得出现公有的成员变量。
7. 代码中主要数据表示一律使用类，不得使用结构体；数组使用注意内存问题，数量较多时尽量使用动态数组。
8. 同类型的成员函数和成员变量放到一起，不要出现多个相同的成员属性声明。
9. 在头文件中成员按照先成员函数再成员变量的顺序编码，成员函数中按照public protected private的顺序编写。
10. 不涉及到信号槽的类不要继承QObject，也不要声明Q\_OBJECT宏。
11. 要求成员变量与普通变量能够明显区分，例如可在成员变量前面加前缀“m\_”或者“\_”
12. 命名规范不做特别要求，建议使用匈牙利命名法或者驼峰命名法。
13. 编码中采用英文名称，禁止拼音命名。
14. 编码要求有适当的清晰的缩进。
15. 函数功能注释要求写在函数声明的前面（这样VS可以提示）。
16. 一个文件的长度原则上不要超过2000行，一个代码段的纵向原则上不要超过两屏显示横向不超过一屏显示（横屏）。代码段嵌套原则上不得超过四层。

## 开发过程规范

FastCAE开发过程提倡迭代开发与敏捷开发的方式，简单的功能模块（一周以内可完成）的开发人员可自行安排。时间较长或者改动较大的功能，需要有小组负责人或者总体负责人完成开发计划以及软件原型之后交由开发人员进行开发。

* 代码位置

FastCAE使用行云进行代码托管，代码上传与下载使用Git，地址为<https://codeup.teambition.com/5c669f8a3ca32d0001d8bbd7/DISOSoft/FastCAE.git> 。

* 分支新建与合并

Master分支是主分支，也是版本发布的分支。任务量较大或者时间较长，且相关性较小的项目采用分支开发的方式，分支由产品或项目负责人进行创建与合并，需要特别注意因为分工造成的代码交叉修改的情况。Master分支需要及时合并到各其他分支，保证功能性。

* 缺陷修复

FastCAE缺陷管理列表地址：<https://www.teambition.com/project/5ed9dc3c289d7400232f5ac9/tasks/scrum/5ed9dc3c289d7400232f5aea> 修改缺陷时需要注意测试版本，若为发布版本，则需要在master分支修改；否则则在对应的分支修改，不可以跨分支修改。

* 版本发布

版本发布由项目或产品负责人负责，发布的版本要求源码或者安装文件，发布之前必须经过测试人员测试。

## 结语

产品开发过程漫长而艰辛，需要开发人员投入大量的心血，制定这个规范并非约束，而是希望规范开发的过程，减少不必要的错误，提高开发效率。产品开发不是一锤子买卖，需要每一个开发人员真正站在用户的角度去开发，经过长时间不断的迭代完善才能形成经得住考验的产品，希望每一个参与到产品开发的同仁们能够发挥各自的优势与特长，积极献计献策，大家一起做好这个产品。

附录一

**Git使用规范**

* 开发人员必须使用自己的账户密码进行登录，禁止使用他人账号进行任何操作。
* 每次提交代码之前必须先更新代码且确保更新完成后代码能够编译通过。如果更新操作的文件与本地工作副本修改的代码是同一文件，应调试代码，确保他人更改的代码部分不影响本地修改部分的代码逻辑。
* 出现冲突及时解决，不要进行相互覆盖。
* 代码进行整体提交，一个项目一次性整体提交，不得分多次提交或单个提交，以免发生提交后在他人副本上产生编译不过的问题。
* 提交代码应写合理清晰的log信息，确保其他成员在观看log后就知道本次提交的内容改动。
* Git的版本控制应包含：源代码、UI文件、脚本文件、资源文件、环境配置说明文件（如果需要，如依赖的库有哪些、这些库配置的相对路径等）等。严禁纳入版本控制的文件包括：中间编译文件、平台相关工程文件（\*.vcproj \*.opensdf \*.sdf \*.suo \*.vcproj.filters \*.vcoroj.user）等。
* 确保检出的代码能够在按照使用说明进行配置后，能够编译通过正常运行。
* 以下情况禁止上传代码：

代码不完整的。

代码编译不通过的。

与服务器版本存在冲突的。

影响全团队成员开发的。