一、目的 2

1.0不重复造轮子 2

1.1技术沉淀 2

1.2提升公司软实力 2

二、PisenKit维护规范 2

2.0原则问题 2

2.1命名规范 2

2.2版本管理 3

2.3子模块的用法 3

# 目的

## 1.0不重复造轮子

项目之间、团队之间可以共享代码，减少不必要的重复劳动，提高开发效率。

## 1.1技术沉淀

让每一次的项目开发都有收获！养成经常思考、整理、优化代码的习惯，沉淀的不仅仅是代码量，更重要的是技术和经验！

## 1.2提升公司软实力

该公共库版权属于品胜公司，由iOS开发团队共同打造，待时机成熟后可以共享到开源平台，提升公司iOS开发团队的形象。

# 二、PisenKit维护规范

识别可加入基础库的代码 -> 规范化代码并放入恰当的位置 -> 剥离可配置的信息 -> 通过白盒测试 -> 加入develop分支 -> 某个时间节点（如项目完成）统一合并到master

## 2.0原则问题

为了不让PisenKit成为一个简单的代码容器，同时保证在代码不断增加的过程中不会影响库的稳定性、易用性、扩展性，因此在加入代码时务必遵守以下原则：

1. **业务无关性：**保证在不修改任何源码的情况下直接适用于别的项目。
2. **代码原创性：**鼓励创造更好的轮子。禁止直接将第三方代码甩进去的流氓行为！允许将常用的基础功能代码在规范化（命名、格式化、统一用法、注释）后加入基础库，或者只添加适配器代码调用第三方库。
3. **用法统一性：**将封装好的功能提供统一风格的对外接口。比如参数配置的方式采用Delegate、Block、单例对象、Category重新同名方法、链式编程等。
4. **功能独立性：**封装的功能尽量单一，减少对平级功能的依赖，目的是保证功能块能随时被更好的方案替换。
5. **功能可靠性：**可以通过编写白盒测试用例来确保功能的可靠性。

## 2.1命名规范

1. 目录结构

Demo：对基础库进行单元测试的demo项目

Documents：基础库维护的规范文档

PSKSingleton：存放单例类

PSKUtils：存放静态类，一般一个方法就是一个独立的小功能。

基础库结构目录：



1. 类名

a. 自定义类命名规范：PSKXxxxx

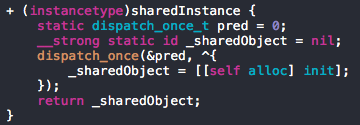
b. category类命名规范：SYSTEM\_CLASS (PisenKit)

c. 封装静态方法的类：PSKXxxxUtil、PSKXxxxHelper

d. 单例类：PSKXxxxManager

1. 方法名

a. 创建单例方法：

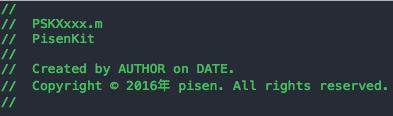


b. 私有方法前面加下划线’\_’

c. category方法前必须有前缀’psk\_’

1. 文件头注释

必不可少的要素：类名、包名、作者、创建日期、版权信息。如，



## 2.2版本管理

master：规范、稳定、可靠的代码库，在master上打tag

develop：当前项目的开发分支，待项目完成后合并到master

## 2.3子模块的用法

添加子模块

git submodule add http://git.pisendev.com/iOS.PisenKit.git TARGET\_PATH

初始化子模块

git submodule init

git submodule update

更新子模块

git submodule foreach git pull --rebase