

开发环境报告

1、项目代码管理

项目的代码管理采用 `git-flow` 流程进行

基本介绍

`git-flow` 是一个 `git` 的拓展集，为项目的规范化管理提供了一个很好的解决方案。

安装

安装前需要一个可以工作的 `git`

OSX

```
$ brew install git-flow 或 $ port install git-flow
```

Linux

```
$ sudo apt-get install git-flow
```

开始

使用 `git-flow`，先初始化一个现有的git库

```
git flow init
```

特性

增加新特性

新特性的开发是基于 'develop' 分支的，通过下面的命令开始开发新特性：

```
git flow feature start MYFEATURE
```

完成新特性

完成开发新特性，这个动作执行下面的操作。

```
git flow feature finish MYFEATURE
```

发布新特性

发布新特性分支到远程服务器，所以，其它用户也可以使用这分支。

```
git flow feature publish MYFEATURE
```

取得一个发布的新特性分支

取得其它用户发布的新特性分支，并签出远程的变更。

```
git flow feature pull MYFEATURE
```

作一个release版本

开始准备release版本

开始准备release版本，使用 `git flow release` 命令。

```
git flow release start RELEASE [BASE]
```

创建 `release` 分支之后立即发布允许其它用户向这个 `release`分支提交内容是个明智的做法。命令十分类似发布新特性：

```
git flow release publish RELEASE
```

完成 release 版本

完成 `release` 版本是一个大 `git` 分支操作。它执行下面几个动作

```
git flow release finish RELEASE
```

紧急修复

开始 git flow 紧急修复

像其它 `git flow` 命令一样, 紧急修复分支开始自：

```
git flow hotfix start VERSION [BASENAME]
```

完成紧急修复

当完成紧急修复分支，代码归并回 `develop` 和 `master` 分支。相应地，`master` 分支打上修正版本的 `TAG`。

```
git flow hotfix finish VERSION
```

参考：http://danielkummer.github.io/git-flow-cheatsheet/index.zh_CN.html;;

2、Bug管理系统

系统名称：

EasyBug

系统简介：

EasyBug是一个在线的Bug管理系统，相比别的Bug管理系统更加轻便快捷，项目的创建与成员的绑定都是自动化，核心流程比较简单，成员分工明确，集成了主要的截图功能，报表丰富，特别适合中小型项目的开发。

系统使用：

在线系统，注册账号之后即可使用（唯一不足的是需要“翻墙”）。图形化界面，操作容易便捷。

项目的创建与管理

进入“我的项目”页面，点击创建项目或者管理已有的项目

成员的绑定与管理

进入“我的团队”页面，即可看到相关的信息，在此页面中可以增删查改相关的人员，并且可以改动人员的权限。绑定成员后，系统会自动通知相关的人员，且为每个成员初始化一组账号与密码

Bug的管理与报告

每个成员都可以上传检测到的Bug，系统还配置了丰富的图像处理工具，并且会根据Bug的类别来通知相关的负责人员，分工明确

EasyBug网址：<http://easybug.org/>

