Atitit cvs git 使用总结

目录

[1. Vcs版本控制系统目的：： 1](#_Toc21144)

[1.1. 3、基本概念 1](#_Toc7430)

[2. Vcs的一般功能 2](#_Toc19341)

[3. GUI工具 2](#_Toc30898)

[3.1. Ide 的git插件，方便快捷 2](#_Toc24716)

[3.2. Git gui等根据tortoi gui工具等，优点是语言ide无关性 2](#_Toc20178)

[4. Pull clone http vs git模式 2](#_Toc30578)

[5. Commit提交类型 2](#_Toc13419)

[5.1. fix:  bug修复 3](#_Toc25959)

[5.2. feat: 功能增加 3](#_Toc13679)

[5.3. chore:, docs:, style:, refactor:, perf:, test:, 3](#_Toc3156)

[5.4. Refect重构 3](#_Toc8920)

[6. 插件git commit template 3](#_Toc32029)

# Vcs版本控制系统目的：：

同步文件的最新的修改

　多人协作开发或者修改，共享数据

　　错误恢复

　　多功能的并行开发（分支功能、特性-合并操作）

如果不使用vcs，那么就需要手动控制版本，需要打版本标记比如数字版本 v1 v2 v3等，日期版本 0511,0611等。。年号版本 ，，以及其他keyword。。当冲突合并的时候人工对比版本号判断最新版本。。

## 3、基本概念

　　repository——存放所有文件及历史信息

　　checkout ——取出或者切换到指定版本的文件

　　version ——记录表示一个版本（编号或者其他代码），某个特定状态下的资源

　　tag ——记录标识一个主要版本（1.0 2.0 3.0）

# Vcs的一般功能

基本的版本管理与同步

分支管理

单文件对比

# GUI工具

## Ide 的git插件，方便快捷

，缺点是多语言可能使用不同的ide插件操作不同

## Git gui等根据tortoi gui工具等，优点是语言ide无关性

# Pull clone http vs git模式

http更加简单

# Commit提交类型

## fix:  bug修复

## feat: 功能增加

1. ****BREAKING CHANGE:**** a commit that has the text BREAKING CHANGE: at the beginning of its optional body or footer section introduces a breaking API change (correlating with **[MAJOR](http://semver.org/" \l "summary)** in semantic versioning). A breaking change can be part of commits of any type. e.g., a fix:, feat: & chore: types would all be valid, in addition to any other type.

## chore:, docs:, style:, refactor:, perf:, test:,

## Refect重构

# 插件git commit template