# 配置管理之gitlab

配置管理（Configuration Management，CM）是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理的目标是记录软件产品的演化过程，确保软件开发者在[软件生命周期](http://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E7%94%9F%E5%91%BD%E5%91%A8%E6%9C%9F" \t "_blank)中各个阶段都能得到精确的产品配置。

## 一、实现的功能：

### 1.1并行开发支持

因开发和维护的原因，要求能够实现开发人员同时在同一个软件模块上工作，同时对同一个代码部分作不同的修改，即使是跨地域分布的开发团队也能互不干扰，协同工作，而又不失去控制。

### 1.2修订版管理

跟踪每一个变更的创造者、时间和原因，从而加快问题和缺陷的确定。

### 1.3版本控制

* 能够简单、明确地重现软件系统的任何一个[历史版本](http://baike.baidu.com/item/%E5%8E%86%E5%8F%B2%E7%89%88%E6%9C%AC" \t "_blank)。
* 产品发布管理：管理、计划软件的变更，与软件的发布计划、预先定制好的生命周期或相关的质量过程保持一致；[项目经理](http://baike.baidu.com/item/%E9%A1%B9%E7%9B%AE%E7%BB%8F%E7%90%86" \t "_blank)能够随时清晰地了解项目的状态。

## 二、管理结构：

* 传统结构采用一个master主分支，一个release版本分支，无数个dev开发分支组成。在dev开发完成之后在release分支上合并。并且保留每一个版本的历史信息，可以随时回退到任意标记的历史版本。以手机制造为例，手机v1.1和手机v2.0。不断开发新版本，但是如果v1.1出现问题可以回退到v1.1的版本并且进行修复。
* 现在的互联网采用的是一个master主分支，少数dev开发分支，新的dev分支及时合并到master分支中。互联网没有采用传统结构，因为保留太多版本在最终合并的时候可能会遇到各种矛盾冲突。而且互联网结构是一直向前的，我们的网页一直在更新，但是不需要回退到以前的版本。

## 三、管理内容

配置管理基本是对以下几点进行管理：代码、文档、环境、软件和配置。

针对代码管理，主要分为版本管理、变更管理、文档管理和发布管理这四个部分。

常用配置管理软件有SVN，vss，git等，我们这里以git为例。