

大话版本控制

笔记本：版本控制

创建时间：2018\9\18 星期二 11:17

更新时间：2018\11\9 星期五 15:32

作者：804790605@qq.com

在团队开发的过程中不可避免的会涉及到版本问题，各个团队会使用各种各样的版本控制程序来维护版本的一致性。那么什么是版本控制？版本控制有哪些应用程序，这些应用程序又有什么区别呢？

（下面请原谅我又进行大量的间歇性copy内容了，看来必须声明：本文章属介绍内容，总结感想性质，非自主技术研究报告性质，不喜勿喷！）

一、概述

首先来看看百度大神：百度百科对于版本控制的定义吧！

版本控制是指对软件开发过程中各种程序代码、[配置文件](#)及说明文档等文件变更的管理，是[软件配置管理](#)的核心思想之一。版本控制最主要的功能就是追踪文件的变更。它将什么时候、什么人更改了文件的什么内容等信息忠实地记录下来。每一次文件的改变，文件的版本号都将增加。除了记录版本变更外，版本控制的另一个重要功能是并行开发。版本控制包括：检入检出控制、分支和合并、历史记录。

个人认为，版本控制的主要目的是：通过使用版本控制(Revision control)（一种[软件工程技巧](#)），籍以在开发的过程中，确保由不同人所编辑的同一档案都得到更新。



看着这样一个版本变化过程，只能在内心里泪流成河了，所以说版本控制的过程也是过程追踪记录，成就达成的过程。

二、常见应用程序

（下面请原谅我又进行大量的间歇性copy内容了，看来必须声明文章属介绍内容总结感想性质，非自主技术研究报告性质）

1、VSS

Visual SourceSafe：微软的版本控制工具，仅支持Windows操作系统。虽然简单好用，但是仅适用于团队级开发，不能胜任企业级的开发工作。

VSS优点：安装、配置、使用均较简单，很容易上手使用；操作简单，容易掌握；权限划分可到文件夹级，有Read、Check-Out & Check-In、Add/Rename/Delete、Destroy四种权限级别。

缺点：权限管理基于文件共享形式，只能从文件夹共享的权限设定对整个库文件夹的权限，而且必须要有可写权限；版本管理和分支管理只能靠人为的手工设置；版本发行时，只能手工挑选对应的版本文件进行发布；安全性不高，基于文件系统共享实现对服务器的访问，需要共享存储目录，这样用户可以对VSS的文件夹执行删除操作。

2、CVS

CVS是一个典型的服务器/客户端软件，有Unix版本的CVS、Linux版本的CVS和Windows版本的CVS。CVS支持远程管理，项目组分布开发时一般都采用CVS。安装、配置较复杂，但使用比较简单，只需对配置管理做简单培训即可。安全性高，CVS服务器有自己专用的数据库，文件存储并不采用“共享目录”方式，所以不受限于局域网。CVS可以跨平台，支持并发版本控制，而且免费。CVS不支持文件改名，只针对文件控制版本而没有针对目录的管理，并且缺少相应的技术支持，许多问题的解决需要自己寻找资料，甚至是研究源代码。但也可以根据自己的需要进行编程。

相对功能单一、简陋，适用于几个人的小型团队，在数据量不大的情况下，性能可以接受。

3、SVN

SVN (Subversion) 是一种版本管理系统，其前身是CVS。SVN是根据CVS 的功能为基础来设计的，它除包括了CVS的大多数特点外，还有一些新的功能，如：文件目录可以方便的改名、基于数据库的版本库、操作速度提升、权限管理更完善等。

SVN是Subversion的简称，是一个开放源代码的版本控制系统，是集中式管理的版本控制器，相较于RCS、CVS，它采用了分支管理系统，它的设计目标就是取代CVS。互联网上很多版本控制服务已从CVS迁移到Subversion。说得简单一点SVN就是用于多个人共同开发同一个项目，共用资源的目的。SVN的特点是简单，只需要一个能放代码的地方就OK了。SVN只有一个单一的集中管理的服务器，保存所有文件的修订版本，而协同工作的人们都通过客户端连到这台服务器，取出最新的文件或者提交更新。



4、ClearCase (闭源集中式)

IBM公司的Rational ClearCase是一款重量级的软件配置管理软件，为大中型软件开发企业提供了版本控制、工作空间管理、平行开发支持以及版本审计，可以为拥有上千开发者的大型项目提供全面配置管理支持。

ClearCase提供了全面的配置管理——包括版本控制、工作空间管理、建立管理和过程控制，而且无须软件开发者改变他们现有的环境、工具和工作方式。

ClearCase包括两套：ClearCase LT和ClearCase (MultiSite)。前者可以用于在同一个局域网的开发小组，适合于中小型开发组织；ClearCase (MultiSite)则适应于分布于不同地理位置、不同局域网的开发小组，适合于大型的开发组织。

优势:

增加团队效率——通过对并行开发的支持来实现，包括图形比较和归并、标签、版本目录结构。

增加个人效率 ——通过自动的工作空间管理来实现，如：直接的版本访问、消除了拷贝文件上的时间的浪费。

简单的维护和提高对客户的支持——通过快速准确的重建先前的版本来实现。

快速准确的产品发布 ——通过保证构造的准确性和对软件的每一个元件进行版本控制来实现。

减少错误发生 ——通过事件发生以后对每一个元件的变更进行追踪来实现。

硬件资源的优化 ——通过分布式构造、减少文件拷贝、可用对象的共享等功能来实现。

提高项目协调和编制 ——通过文件注释和开发周期阶段变更的自动关联来实现。

提高产品质量 ——通过灵活的进程控制，和图形接口定制，使得软件开发在实际中保持一致。

更加有效的团队扩展——通过减少系统管理和维护的负担来实现。

支持分布式结构使得团队成长——通过Client/Server结构进行多点复制和及时的对象版本的更新来实现。

使用配置管理工具而降低风险——由于它不干扰软件程序员的工作，所以可以使用常用的工具和文件系统接口。

增加了软件的安全性和保护性——通过使用分布式的存储结构，所有的软件资源会随时更新、在硬盘或网络出现错误时那些被ClearCase存储的版本信息会立刻恢复。

减少培训和实现成本 ——ClearCase通过采用透明结构以及和标准开发工具进行集成来实现。

强有力的开发和维护 ——通过和其它工具（如：缺陷追踪）、系统、结构进行集成。

支持不同种类的开发 ——通过兼容不同平台的软件配置管理系统，如：Windows NT、UNIX、和一些Client端的软件，如：Windows 95、Windows NT、Windows 3.1和Windows for Workgroups。

缺点：ClearCase 太贵，易用性差，培训费用很贵，没有培训，很难上手使用。能用上这个的都是有钱的主，以前在交通银行工作的时候有用到过。说实话对于开发来说就一般般，但好在他是IBM解决方案中的一种，配合其他使用，可以让项目管理更方便。

5、StarTeam (闭源集中式)

StarTeam属于高端的工具，在易用性，功能和安全性等方面都很不错。StarTeam的用户界面同VSS的类似，它的所有操作都可通过图形用户界面来完成，同时，对于习惯使用命令方式的用户，StarTeam也提供命令集进行支持。而且StarTeam的随机文档也非常详细。StarTeam还提供了流程定制的工具，用户可跟据自己的需求灵活的定制流程。与VSS和CVS不同，VSS和CVS是基于文件系统的配置管理工具，而StarTeam是基于数据库的。StarTeam的用户可根据项目的规模，选取多种数据库系统。StarTeam无需通过物理路径的权限设置，而是通过自己的数据库管理，实现了类似Windows的域用户管理和目录文件ACL控制。StarTeam完全是域独立的。这个优势可以为用户模型提供灵活性，而不会影响到现有的安全设置。StarTeam的访问控制非常灵活并且系统。您可以对工程、视图、文件夹一直向下到每一个小的item设置权限。对于高级别的视图（view），访问控制可以与用户组、用户、项目甚至视图等链接起来。StarTeam是按license来收费的，比起VSS，CVS来，企业在启动StarTeam进行配置管理需要投入一定资金。

优点：权限设置功能强大方便。StarTeam的图形化界面，能够使初学者易于接收，而且其缺陷控制功能的功能（基于数据库的Change Request），是相应工具中独树一帜的。

缺点：不支持并行开发，不能很好解决Merge的问题；不支持分支的自动合并，需要手动来处理；速度慢，一定程度上影响开发效率；故障恢复困难，需要有专职管理员维护；没有中文版本；另外，StarTeam集成度较高，移植过程复杂，需要的管理负担大，需要完善的备份计划。

6、GIT（开源分布式）



6.1、GIT

GIT 是一款免费的、开源的、分布式的版本控制系统。旨在快速高效地处理无论规模大小的任何软件工程。与常用的版本控制工具 CVS, Subversion 等不同，它采用了分布式版本库的方式，不必服务器端软件支持，使源代码的发布和交流极其方便。每一个GIT克隆都是一个完整的文件库，含有全部历史记录和修订追踪能力。其最大特色就是“分支”及“合并”操作快速、简便。支持离线工作，GIT是整个项目范围的原子提交，而且GIT中的每个工作树都包含一个具有完整项目历史的仓库。

GIT 本来是面向 Linux 操作系统开发的软件。在 Linux 平台上使用GIT非常简单，都是命令行模式。但对windows以及中文的支持不是很好。

Git每一个终端都是一个仓库，客户端并不只提取最新版本的文件快照，而是把原始的代码仓库完整地镜像下来。每一次的提取操作，实际上都是一次对代码仓库的完整备份。

6.2、GitLab

GitLab 是一个用于仓库管理系统的开源项目。使用Git作为代码管理工具，并在此基础上搭建起来的web服务。可通过Web界面进行访问公开的或者私人项目。它拥有与Github类似的功能，能够浏览源代码，管理缺陷和注释。可以管理团队对仓库的访问，它非常易于浏览提交过的版本并提供一个文件历史库。团队成员可以利用内置的简单聊天程序 (Wall)进行交流。它还提供一个代码片段收集功能可以轻松实现代码复用。

6.3、GitHub

gitHub是一个面向开源及私有软件项目的托管平台，因为只支持git作为唯一的版本库格式进行托管，故名gitHub。gitHub于2008年4月10日正式上线，GitHub可以托管各种git库，并提供一个web界面，但与其它像 SourceForge或 Google Code这样的服务不同，GitHub的独特卖点在于从另外一个项目进行分支的简易性。还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。

6.4、码云

码云是开源中国社区2013年推出的基于 Git 的完全免费的代码托管服务，这个服务是基于 Gitlab 开源软件所开发的，我们在 Gitlab 的基础上做了大量的改进和定制开发，目前已经成为国内最大的代码托管系统，致力于为国内开发者提供优质稳定的托管服务。

码云的主要功能是除了提供最基础的 Git 代码托管之外，还提供代码在线查看、历史版本查看、Fork、Pull Request、打包下载任意版本、Issue、Wiki、保护分支、代码质量检测、PaaS 项目演示等方便管理、开发、协作、共享的功能。

7、Mercurial（开源分布式）

Mercurial 是一种轻量级分布式版本控制系统，采用 Python 语言实现，易于学习和使用，扩展性强。其是基于 GNU General Public License (GPL) 授权的开源项目。

相对于传统的版本控制，具有如下优点：

更轻松的管理。传统的版本控制系统使用集中式的repOSitory，一些和repository相关的管理就只能由管理员一个人进行。由于采用了分布式的模型，Mercurial 中就没有这样的困扰，每个用户管理自己的repository，管理员只需协调同步这些repository。

更健壮的系统。分布式系统比集中式的单服务器系统更健壮，单服务器系统一旦服务器出现问题整个系统就不能运行了，分布式系统通常不会因为一两个节点而受到影响。

对网络的依赖性更低。由于同步可以放在任意时刻进行，Mercurial 甚至可以离线进行管理，只需在有网络连接时同步。

简单易学、易于使用；轻量级，运行快速；可扩展性，易于根据用户需求自行定义、扩展。

8、Monotone（开源分布式）

Monotone是一个免费的分布式版本管理系统。提供了简单的文件事务版本存储，可离线操作，高效的点对点同步协议，支持历史版本敏感的合并操作、轻量级分支处理以及集成代码评审和第三方测试工具。使用加密的版本命令方式和客户端 RSA 认证，很好的支持国际化，不依赖第三方工具，支持跨平台。可运行在Linux，Solaris，Mac OSX，Windows和其他Unixes上，遵循GPL协议。

（目前Google Code支持SVN、Git、Mercurial三种方式）

三、各个应用程序比较

1、CVS与SVN比较

比较项目		CVS	SVN
权限控制	是否依赖系统帐号	依赖	不依赖
	可否对分支授权	否	是
	是否支持LDAP认证	否	是
	图形化帐号管理	否	是(集中管理平台)
	用户可否获取忘记口令，修改口令	否	是(集中管理平台)
目录，文件名变更		否	是
分支	创建分支时间	耗时*	快
管理	分支可见、查询	难	易
二进制文件	二进制优化	否	是
	二进制文件标识	手工	自动

	二进制文件（图形文件）被破坏	易破坏	不易破坏
事物	原子提交	否	是
处理	修改提交说明	单个文件	是
换行	可否指定换行符类型	否	是
符	检查换行符设定，避免跨平台开发带来的混乱	否	是
功能扩展		CVSROOT	hooks 脚本
网络	网络带宽占用	高	低
带宽	脱机命令	否	部分

2、CVS、SVN、GIT比较

名称	版本库模型	并发模式	历史模式	变更范围	网络协议	原子提交性	部分克隆
CVS	Client-server	Merge	Changeset	File	Pserver,ssh	No	Yes
SVN	Client-server	3-way merge, recursive merge, octopus merge	Changeset and Snapshot	Tree	custom (svn), custom (svn) over ssh, HTTP and SSL (usingWebDAV)	Yes	Yes
Git	Distributed	Merge or lock	Snapshot	Tree	custom, custom over ssh, rsync, HTTP/HTTPS, email, bundles	Yes	No

四、概述SVN

见后续

五、概述GIT

见后续

参考资料：

百度百科：<https://baike.baidu.com/item/%E7%89%88%E6%9C%AC%E6%8E%A7%E5%88%B6/3311252?fr=aladdin>

论坛：版本控制总结：<https://www.cnblogs.com/yanghongliang/p/5750306.html>

SVN和Git、GitHub、GitLab、码云之间有什么区别：<https://baijiahao.baidu.com/s?id=1605566374654856802&wfr=spider&for=pc>

Git，Github和Gitlab简介和使用方法：<https://blog.csdn.net/ylgwhyh/article/details/52134338>

Git详解及 github与gitlab使用：<https://www.cnblogs.com/clsn/archive/2017/11/30/7929958.html>

Atlassian为Git做了一个10年的纪录图：<https://www.atlassian.com/git/articles/10-years-of-git/>

GIT教程：<https://www.atlassian.com/git/tutorials>