HUAWEI

华为云技术私享会

华为云社区
bbs.huaweicloud.com

HUAWEI

华为云
技术
私享会

提交代码这点事

庄表伟 华为云软件开发云高级产品经理

华为云
技术
私享会

从个体户到大兵团

提交代码这点事

HUAWEI

这不是一个课程，不介绍基础知识，仅仅是某种逻辑梳理。

前置声明

提交代码这点事

华为云技术私享会

个体户的幸福生活

- 直接登录到服务器，vim或者emacs
- EditPlus的FTP编辑模式，搭配PHP，简直神兵利器
- 在有了用户以后，在自己的本地跑一个Web服务（XAMPP），先简单测试下，然后上传到服务器
- 在Java的Web服务器中，我曾经最喜欢Resin，因为服务器不需要重启
- 如果一个人要搞定太多的代码，只能自己架一个VCS：Version Control Software
  - 当初为何要这么修改：是记在注释里，还是记在提交说明里？
  - 如果没有VCS：改回原来的版本，是一个非常困难的任务

提交代码这点事

华为云技术私享会

软件开发的三大烦恼：

- 需求越来越多——还不断变化
- 人越来越多——水平还不够高
- 代码越来越多——质量还很差

经验总结

提交代码这点事

华为云技术私享会

小组协作

- 工具的选择
  - 用中央服务器，汇集所有人的工作成果
  - 难用到死的VSS（Visual Source Safe），背后的逻辑是什么？
  - SVN（Subversion）的改进之处
    - 版本号
    - 原子提交
    - 分支
    - 存储与网络访问优化
- 开源社区的committer是些什么人？
  - 邮件列表有哪些人？
  - diff与patch
  - code review
  - author与committer

提交代码这点事

华为云技术私享会

小组协作

- 版本与分支
  - 版本的含义
    - 版本号：为一个阶段工作成果命名
    - 隐含质量属性：单双数、小数点、alpha、beta、final
    - 方便收集bug
  - 围绕版本的工作
    - 区分两种代码提交（feature、bugfix）
    - 划分两种工作阶段（特性开发期、发布前准备期）
    - 第三种工作阶段：针对某个版本长期维护，不断打补丁
  - 分支的含义
    - 为并行工作，提供可能
    - 不同的版本，定义不同的分支
    - 为一组（一类）持续的工作，提供意义
    - 批量汇集工作成果（Merge）

工具的演进，与开发模式的演进，往往呈现交互影响的关系。复杂的开发模式，促使新的工具诞生；新工具的引入，促使传统开发模式的革新。

思考与总结

提交代码这点事

git出现之后

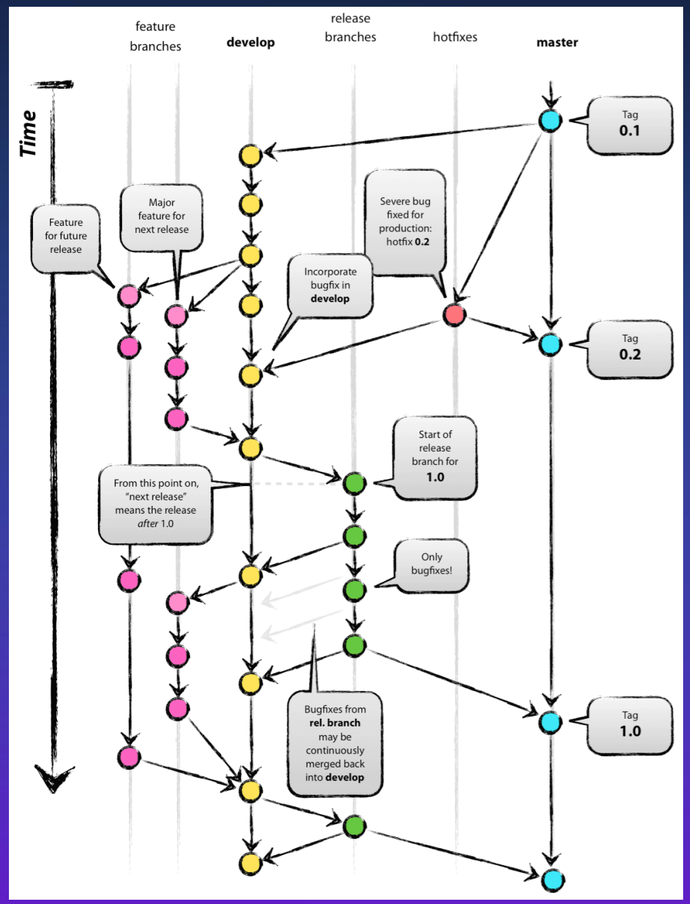
- Git其实并非第一个分布式版本管理系统，却是目前最流行的一个
  - 因为linus大神、因为开源、因为Github……
- Git的设计要点
  - 分布式版本控制系统
  - svn基于revision,也就是delta，git基于状态，每个commit保存了完整的工作区目录
  - svn在单分支上是一条revision的时间线，git则是由commits组成的DAG（无环有向图）
- Git带来的优势
  - 去中心化
  - 更好的分支管理
  - 更多的协作模式

V·T·E		Version control software
Years, where available, indicate the date of first stable release. Systems with names *in italics* are no longer maintained or have planned end-of-life dates.		
Local only	**Free/open-source**	RCS (1982) · SCCS (1972)
	Proprietary	PVCS (1985) · *QVCS* (1991)
Client–server	**Free/open-source**	CVS (1986, 1990 in C) · CVSNT (1998) · QVCS Enterprise (1998) · Subversion (2000)
	Proprietary	AccuRev SCM (2002) · ClearCase (1992) · *CMVC* (1994) · Dimensions CM (1980s) · *DSEE* (1984) · Endevor (1980s) · Integrity (2001) · Panvalet (1970s) · Perforce (1995) · Software Change Manager (1970s) · StarTeam (1995) · Surround SCM (2002) · Synergy (1990) · Team Concert (2008) · Team Foundation Server (2005) · Visual Studio Team Services (2014) · Vault (2003) · *Visual SourceSafe* (1994)
Distributed	**Free/open-source**	*ArX* (2003) · BitKeeper (1998) · *Codeville* (2005) · Darcs (2002) · *DCVS* (2002) · Fossil (2007) · Git (2005) · *GNU arch* (2001) · GNU Bazaar (2005) · Mercurial (2005) · Monotone (2003) · *SVK* (2003) · Veracity (2010)
	Proprietary	*TeamWare* (1990s?) · Code Co-op (1997) · Plastic SCM (2006) · Team Foundation Server (2013) · Visual Studio Team Services (2014)

提交代码这点事

GitFlow简介

- http://nvie.com/posts/a-successful-git-branching-model/
- 图灵社区：《基于git的源代码管理模型——git flow》
- 阮一峰：《Git分支管理策略》
- 本质上：这是一群人，在一个仓库上工作时的协作模型



提交代码这点事

华为云技术私享会

Github出现之后

- fork的本质
  - 降低个人参与开源的门槛
  - 社区分裂常态化->松而不散的大社区
- pull raequest的本质
  - 减少committer进行code review的工作量
  - 特化一部分mailist的工作
- 集成issue管理
  - 一个恰当的复杂度
  - label与milestone
  - 与代码的关联关系

提交代码这点事

华为云技术私享会

Github出现之后（续）

- 与CI集成之后
  - 如何判断一个代码是否合格？
  - 人与工具的分工
- Github Flow
  - upstream
  - feature branch
  - online edit PR
- Gitlab对Github的改进
  - fork vs. protect branch
  - 改进合入流程

以下复杂问题，很多公司也许不会遇到

提交代码这点事

提交代码这点事

华为云技术私享会

更多复杂的问题

- 如何管理更大的仓库
  - 二进制文件：Package Management Server、Git-LFS、Git-Annex
  - 分仓方案：git submodule、git subtree、repo
  - 权限管理：是否应该（能够）支持分目录权限
- 如何让更大的团队，流畅协作
  - CI的效率，成为瓶颈
  - code review的能力与效率，成为瓶颈
  - 分级合入，成为唯一选择
- 如何防止低级错误
  - 流程，更多的流程
  - 规范，更加严格的规范
  - 培训，附带考试的培训
- 如何管理更多需求、缺陷、任务
  - issue爆炸
  - 分类、标签、里程碑、优先级、重要性、紧急程度、所属版本、所属分支...
  - 需求分解、E2E追溯与反向卷积

人、工具、流程，是项目管理的三要素，其中，人是决定因素。

深入思考

提交代码这点事

华为云技术私享会

人的责任

- 提交者的责任
  - 提交足够好的代码：什么样的代码，算是好代码？
  - 不犯低级错误：什么样的错误，是低级错误？
- 审核者的责任
  - 作为架构与质量的看护者：如何守住？
  - 作为技能与品位的拥有者：如何传承？
- 管理者的责任
  - 正确判断，是人的问题？还是工具的问题？还是流程的问题？
  - 以及，如何改进？
- 工具提供者的责任
  - 对工具的价值，有清醒的认识：既可载舟、亦可覆舟
  - 对工具的边界，有合理的认识：既非无能，也非万能

一些感悟与总结：关于打磨

工具特性

用户体验

架构品质

团队磨合

个人技能

THANK YOU