**Git操作规范培训课程教案**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PPT页码** | **时间** | **教案内容（即讲师培训语录）** | **技巧/道具** | **备注** |
| P1-P2 | 5分钟 | **Git操作规范制定的目的**：  统一规范工程师git使用习惯，方便后续智能机项目的信息查询以及软件代码版本的统一管理。  **该规范适用范围**  适用于软件一部、软件二部、软件三部、南京研发中心  **该培训的目的**：   1. 汇总工程师反馈的常见Git操作问题，讲解解决方式 2. 通过培训，推动TN-MB-094 V1.1 Git使用管理规定的执行   备注：  Git正式服务器：192.168.10.204  练习服务器：172.16.20.164  当前204上已添加权限的人员，在164测试服务器上均有读写权限  即各工程师可以连接164服务器，做修改推送更新等各种练习  如在164上练习，第一次练习时需测试一下权限是否正常：ssh git@172.16.20.164  如提示PTY allocation request failed on channel 0则可以进行后面的测试，如提示要输入164的password，请与我联系，查看权限设置是否有误。  204服务器上S6030原型路径: repo init -u [git@192.168.10.204:mt6572/platform/manifest.git](mailto:git@192.168.10.204:mt6572/platform/manifest.git) -b sw -m S6030\_4.2\_V1.0.xml  则：练习时下载路径为repo init -u git@172.16.20.164:mt6572/platform/manifest.git -b sw -m S6030\_4.2\_V1.0.xml  164测试服务器上可以练习的repositories 有：mt6572 mt6582 mt8389 mt6577\_jb mt6589 mtk |  |  |
| P3 | 5分钟 | **Git相关操作的概览**  是总整体上介绍一个项目从开始下代码 修改 提交的整个循环，所用到的git相关的基本介绍和操作。直观的看就是本页PPT所列：   1. 首先，从下载项目的初始化好的代码开始。   软件开发人员开始一个项目，是从SCM给出一个初始化好的路径开始的。  简述repo及提货单；项目提货单命名规则；204服务器上提货单在仓库的分布情况  例如此路径：repo init -u [git@192.168.10.204:mt6572/platform/manifest.git](mailto:git@192.168.10.204:mt6572/platform/manifest.git) -b sw -m S6030\_4.2\_V1.0.xml  标红部分：repo 全局  git@172.16.20.164:mt6572这里mt6572是一个大的仓库名，这个位置也可能是mtk、mt6589、mt6582等  -b sw这里sw是提货单所在的分支名  -m S6030\_4.2\_V1.0.xml这里xxx.xml就是通常说提货单   1. 代码修改阶段   讲解分支及分支相关的git命令、常见问题解决；合并方式merge和cherry-pick的对比  在本地下载好的一套代码上进行开发，操作分支、合并等基本git命令才可以完成后续的操作。   1. 修改提交相关   Git本地提交常用命令介绍；Git提交格式的要求  所有的修改只有提交才可以将修改真正保存、即使更新也不会丢失；也只有提交的结果才可以git log等查看   1. 修改推送至服务器   TAG简介及TAG命名格式；TAG提交的命令；TAG应带有的信息  不同的项目，我们是通过tag进行管理的。在自己验证好的修改上打版本tag，并将该tag推送至服务器才可以让同项目的其他成员取到你的修改   1. 项目更新、切换及其他常用Git操作   git log、git diff、git checkout、git reflog等常用易用命令简介  因为git是一个分布式的系统，你本地只有从服务器同步了代码，则你本地也是一个完整的代码库。  即使没有网络，你也可以从一个项目切换到同库的另一个项目。没有网络，你也可以完成提交，及才看历史记录等git的大部分的操作。  这一部分主要是讲常用到的一些更新、切换的常用操作。 |  |  |
| P4-P8 | 10分钟 | **repo及提货单的介绍**   1. repo一个工具，封装简化多个git仓库的管理；   repo在我们204服务器上也有一个repo仓库，主要是存放repo命令  备注：搭建智能机开发环境PPT所说的/usr/bin/repo，必须检查的路径：REPO\_URL=‘git@192.168.10.204:tools/repo.git‘ 就是204服务器上的repo仓库路径   1. 提货单是一个从多个git仓库中获取指定版本的配置文件   提货单也是由一个专门的git仓库来管理，这个仓库在隐藏的.repo下面， manifest.git   1. repo的使用过程，概况来说是PPT中所列4条：   克隆提货单仓库、克隆提货单、同步仓库、检出文件（PPT里较详细，这里不详写）   1. 从已有项目切换到新项目，路径获取方式介绍，切换报错常用处理   研发项目汇总表是由配置组整理更新  研发项目汇总表的SVN路径：https://172.16.5.199/svn/PROJECTS/跨项目使用/项目信息/研发项目汇总表.xlsx  这个汇总表里的每个项目包含以下信息：项目名、项目路径、平台、基线、开发人员、SWPM、SCM等  大家通过查到的项目路径可以自行下载、切换项目。如与异常情况，可咨询配置组同事   1. 提货单命名规则   参看PPT即可   1. 204服务器上各项目提货单分布简介   图中标价为同色的，即同一个库的项目可相互切换，如mtk下9201、9301项目可相互切换  图中标记为异色的，即不同库的项目是隔离的，如mtk下的9201，不能切换到mt6589下的9111  初始操作git可以查看这个图，能更直观的看各项目的分布情况。  如果较熟练之后，自己就知道如果快速切换项目。 |  |  |
| P9 | 3分钟 | **Git状态转换图**  先看这个图，便于后面讲解git相关的修改和提交。如果之前熟悉SVN等工具的使用，看这个图，大家可以跟之前的知识对比，能更快的掌握该工具的使用。   1. git的三个状态：untrack/unstage、index/staging、本地库/staged 。即可理解为未跟踪、缓存区、已提交 2. 经常操作的git add就是添加到缓存区、git commit是叫缓存区的内容提交，add+commit就是一个完整的本地提交、提交完成会有对应的SHA1值生成 3. 本地与服务器间交互是通过git pull、git push |  |  |
| P10-P14 | 10分 | **查看git各命令的使用说明：git xxx --help**  **如：git log --help、git branch --help**  这部分主要是代码修改阶段常用到的，具体的知识点详看PPT即可，这里不复制了  Git分支相关内容的介绍   1. 使用git分支的情况、分支命名格式（强调点1）   每个项目在提交前新建的本地分支必须统一：TAG名\_branch   1. Git中有关分支的命令、及使用情况 2. Git分支合并两种方式的对比：merge VS cherry-pick，简言之：线合 vs 点合 3. 对应git操作比较熟悉的同事：git reflog+git cherry-pick，介绍该方式的使用及好用之处 4. 订单合并原型，常见两种情形的介绍 |  |  |
| P15-P16 | 5分 | **Git提交相关操作的介绍**  这部分为便于理解每个命令都会操作一下，便于理解   1. 查看状态 2. 往stage缓存区添加文件 3. 从stage缓存区移除文件 4. 从git删除文件 5. 查看差异 6. 回退改动 7. 提交命令及提交格式（重点强调2）   提交日志规则：<tag名称><本次提交类别><详细修改记录> |  |  |
| P17 | 5分钟 | **同步服务器代码**  repo sync 全局同步  repo sync . 同步本地当个仓库  repo sync [remote path] 同步[remote path]到本地  这3种同步命令的区别及适用情况  常见同步报错情况及解决方式介绍，详见PPT  且以上操作都可以在164测试服务器上试下 |  |  |
| P18 | 5分钟 | **查看已提交的日志**  git log 简介各参数使用情况：-p、--stat、[version1]..[version2]、--gerp、--author及两日期之前等等  git diff 简要说明与git log的不同：git diff主要是显示差异，git log主要是显示日志  git reset回退 加不同参数的情况  git revert简要说明与git reset的不同，以及主要适用回退一条的情况  各操作都可在164测试服务器上试下 |  |  |
| P19-P20 | 5分钟 | **提交推送至服务器：升级项目tag并提交到服务器**   1. 升级tag命令、推送升级后的tag到服务器 2. Tag命名方式简要介绍：主要是配置管理员打tag，工程师做必要的了解，方便项目的某些操作   PPT中已详细列出各种类型的tag及举例，此次详看PPT即可。 |  |  |
| P21-P23 | 5分钟 | 1. 版本编译发布：配置管理员编译版本、软件工程师编译版本，两种情况各自需注意的地方 2. 本次培训重点回顾 3. Git操作规范的其他说明点 |  |  |