# JIRA基础培训

1.创建一个issue，创建一个subtask，添加附件，创建工时log work。

2.导入cvs: save the configuration

3.快捷键使用：右上角--->帮助

4.仪表盘--->管理仪表盘--->创建新面板；添加小工具

5.更改语言：右上角--->用户信息

6.冲刺：关注时间；看板：关注任务完成情况

7.看板创建：1.创建项目本身自带；2.在看板按钮新建

8.看板--->配置--->卡片布局。

9.

# JIRA管理员培训

1.创建项目--->创建3个问题类型--->添加3个问题类型到一个问题类型方案--->关联到创建的项目

2.自定义字段（级联）--->新建界面，添加到界面--->添加界面到界面方案，关联到项目

3.新建状态--->新建工作流---->添加工作状态到工作流

4.项目--->查看所有问题和过滤器--->工具--->批量操作

5.创建用户，添加到用户组，创建一个角色，把用户和用户组添加到这个权限角色。

6.项目--->创建项目--->使用共享项目创建；创建出来的项目工作流和模板项目一样。

7.

# BITBUCKET日常使用培训

1.创建一个仓库（右上角头像查看资料），加入内容，导入一个外部仓库。

2.fork一个库，差异化修改，提交合并请求

3.Sourcetree创建仓库，创建一个developer下的feature分支，工作流初始化仓库

# BITBUCKET管理员培训

1.创建项目权限（IT人员有）--->设置---->项目权限管理（用户或组设置）；添加仓库--->仓库权限设置--->设置分支权限；hooks下禁止强制推送

2.

3.

# Bamboo日常使用与管理员培训

1.task顺序执行，由多个task组成job；持续集成要素、原则

2.创建project--->创建plan--->添加task，加入maven3.x插件（mvn，不选底下的复选框），加入-Dmaven.test.skip=true clean package--->右上角actions的configuration job里面的artifacts添加.jar附件（记得配置仓库）。

3.定时编译：configuration--->plan configuration--->Triggers设置时间或者仓库。触发后通知指定的人。

4.创建一个bug给jira：点击工程名，右上角application links页面后点击add link到JIRA--->all builds页面下点击build，右上角actions下create issue创建问题到jira

5.系统设置界面--->删除，移动plan，批量配置权限，创建user，仓库配置，备份；agents（构建编译的环境）

# Git入门使用

1.git有三个配置文件位置：、etc/config;~/.config;工作目录下

2.帮助<git help commit>

3.从当前目录初始化<git init>

4.<git remote add origin url>；创建key<ssh-keygen -t rsa>

5.创建忽略文件<vim .gitignore>，没有被提交的未被版本控制的文件

6.<git add -u>添加所有被版本控制跟踪的文件到暂存区，不能使用-a（这个会提交所有的文件）

7.<git diff>工作区和暂存区比较；<git diff --cached> head和暂存区比较

8. <git remote set-url origin ssh://git@192.168.10.52:7999/~test/cang1.git>重新连接一个新的仓库。

9.移除文件：<rm filename>移除；<git rm filename>

10.创建分支：<git checkout -b dev>移除分支<git branch -d dev>删除远程分支<git push origin :dev>；提交远程分支<git push origin dev:dev>

11.<git tag 1.2> <git checkout 1.2>;删除<git tag -d 1.2>；推送远端<git push origin tag 1.1><git push origin --tags>;删除远端游离分支<git push origin :refs/tags/1.2>;获取远端tags<git fetch origin tag 1.2>

12.修改过的文件恢复<git checkout readme.txt>;修改过提交到暂存区的文件恢复<git reset filename>;提交到本地仓库的修改过文件恢复<git reset --soft commitid>(--hard重置到先前状态；--soft恢复到暂存区，文件没变)；推送到远程仓库，撤销恢复1.先<git reset --hard

commitid>还原到先前版本，<git push -f>(不建议使用-f选项) 2.反转提交<git revert commitid>，提交<git push>；<git log>,<git reflog>

13.查看当前所连接的远程仓库<git remote -v>或者通过.git文件夹下的config文件查看

# Git客户端sourcetree，git-flow工作流培训

1.git-flow有两个永久分支，具有权限控制（master稳定对外发布和develop）；辅助分支feature（新功能）release、hotfix

2.工具--->选项--->一般下的SSH客户端选择“OpenSSH”选项

3.克隆/新建--->本地工作副本；仓库--->Git工作流--->初始化仓库、建立（完成）新功能、建立（完成）新的发布版本、建立（完成）新的修复补丁

4.

5.

6.

7.

8.