

1. 多人开发时缺乏代码审查

目前svn代码管理中，并没有代码审查工具，任何人通过验证后都可以向主分支中进行提交代码，造成代码紊乱，难以控制

git中有gerrit代码审查服务器，任何人提交只有通过代码审查后才可以进入开发分支

2. 开发依赖于svn中央服务器，当svn中央服务器宕机时，本地svn将无法提交及查看日志

当svn服务器挂掉时，将无法进行提交

git是分布式代码管理工具，能进行代码本地提交，不依赖中央服务器

3. 当代码锁时，影响开发

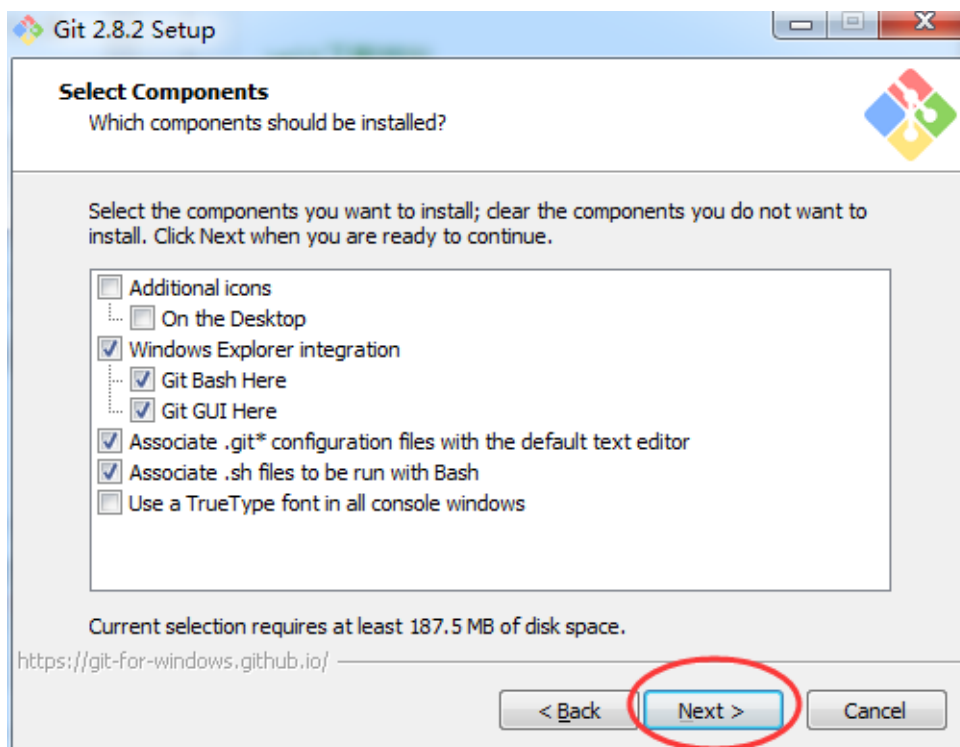
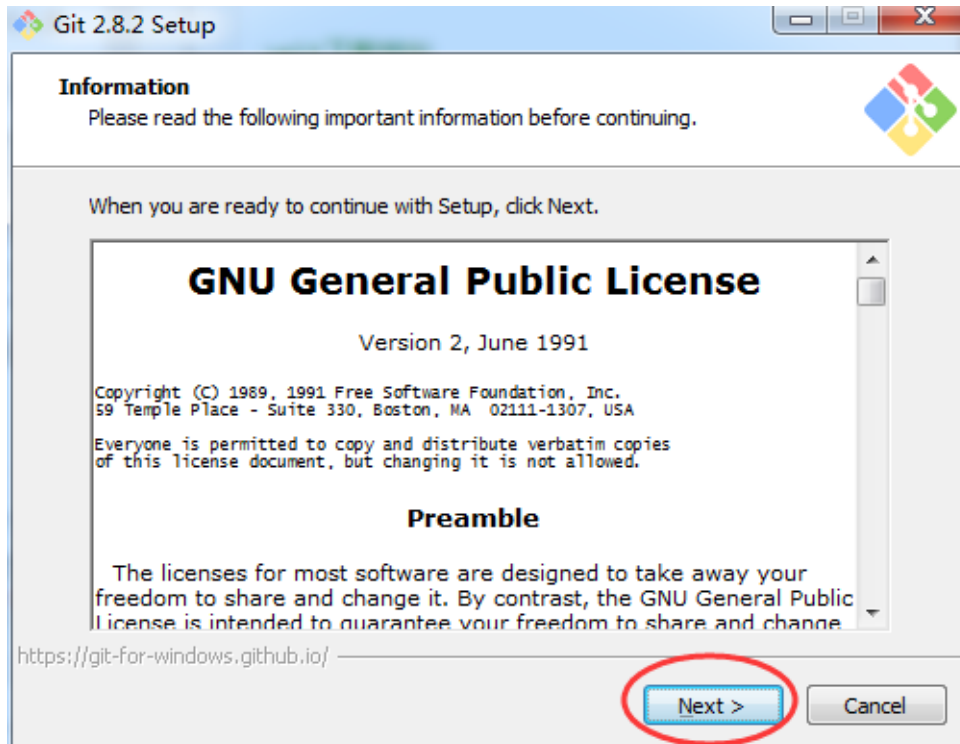
svn只有一个主线，为了不影响版本的发布，每次都需要将svn服务器锁掉，再次期间，研发人员无法进行新特性的提交。

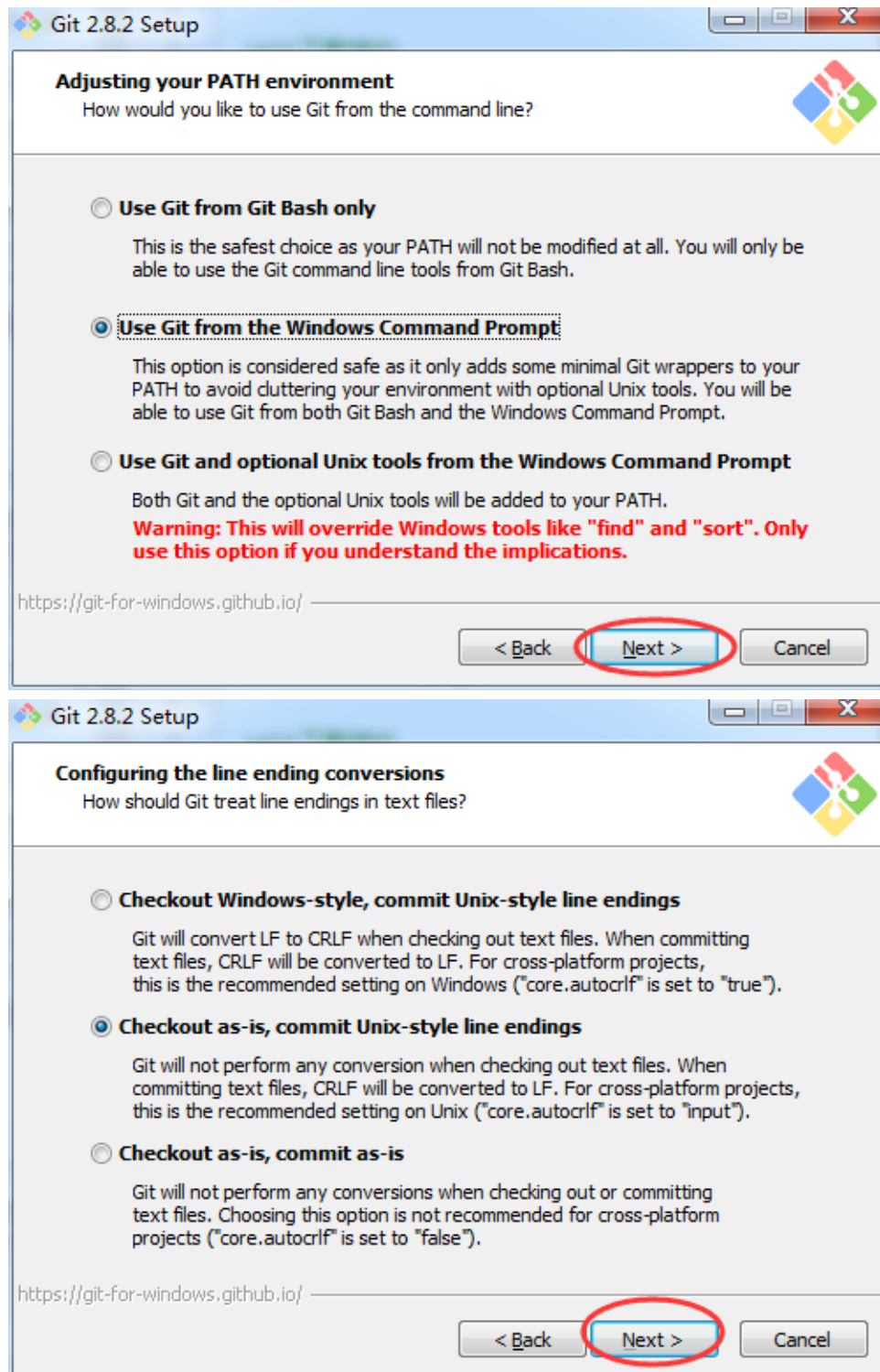
git通过分支进行解决，代码在发布线上发布，开发在研发线上提交

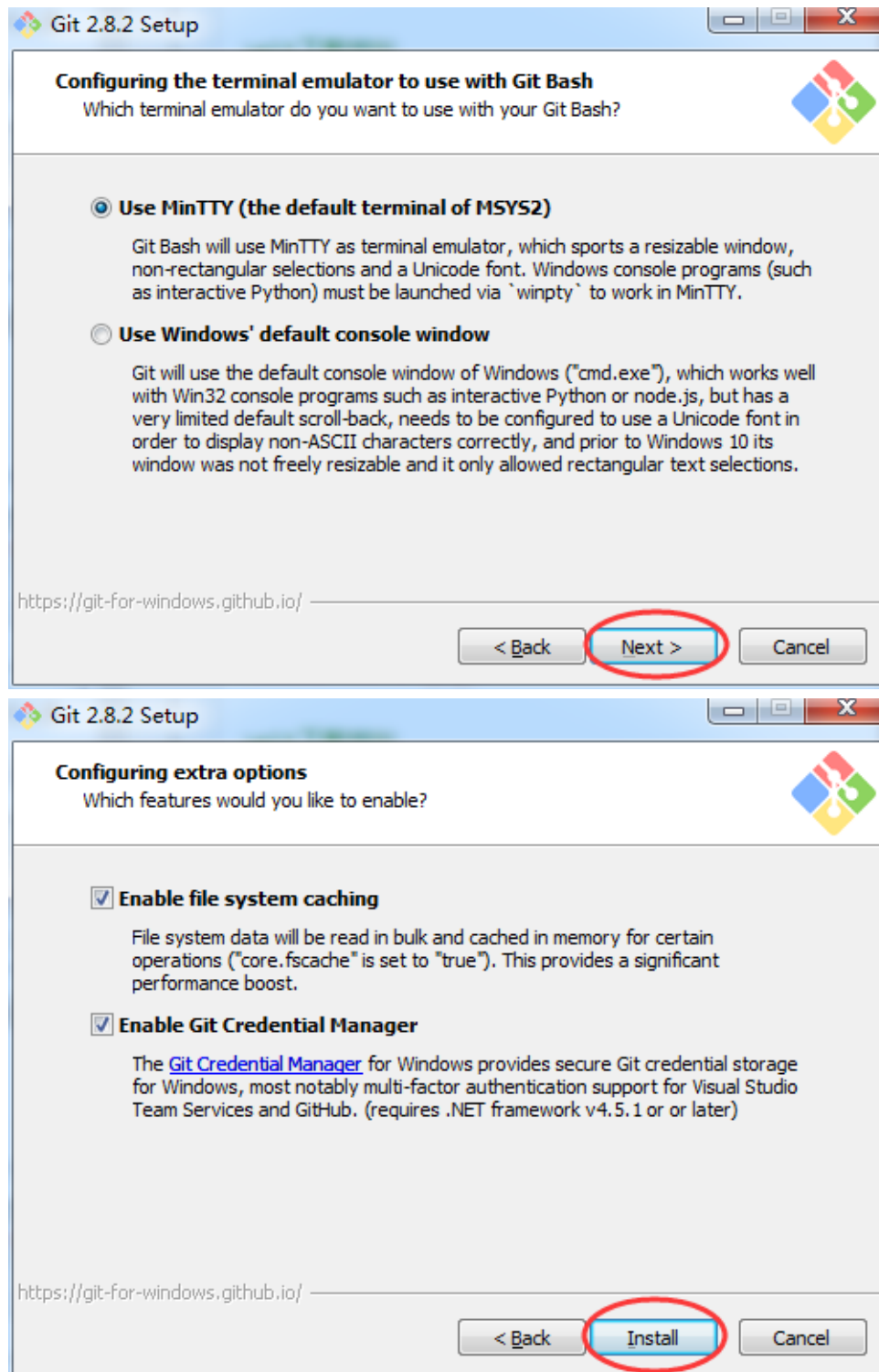
1. windows 安装git

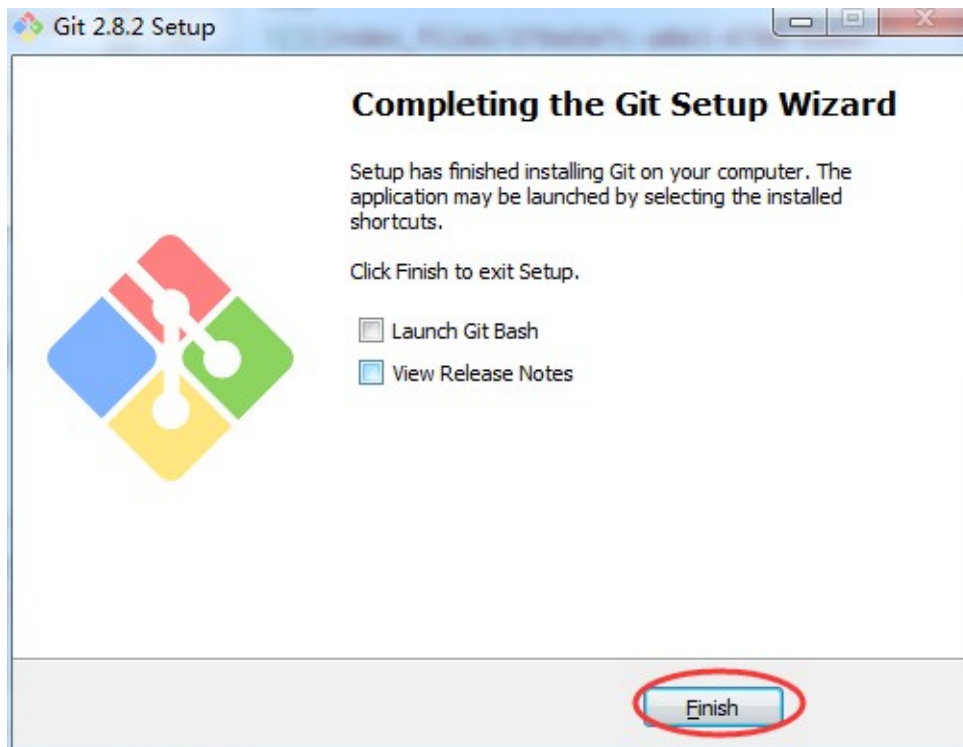
git下载地址

<https://git-scm.com/download/win>



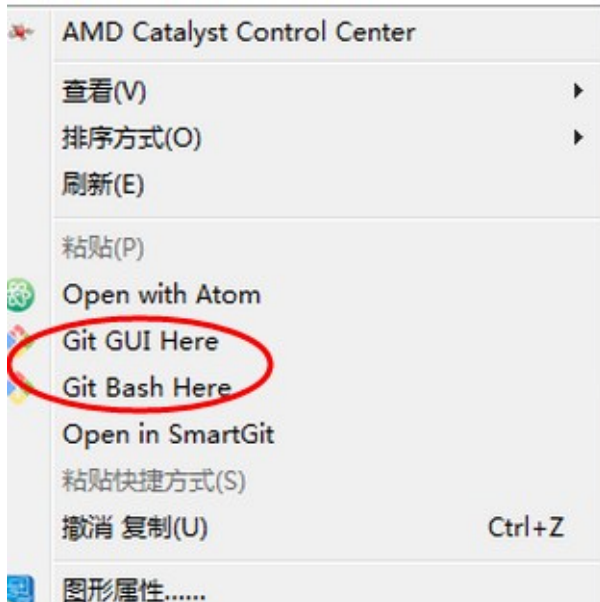




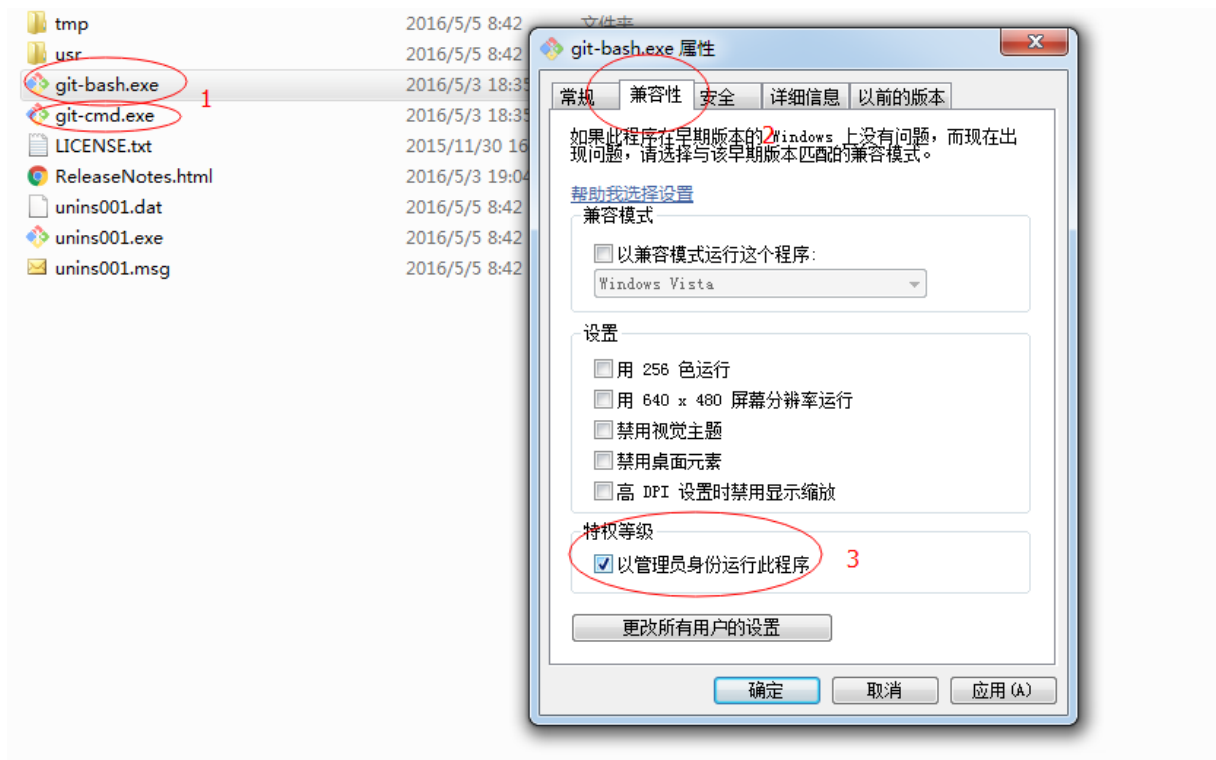


我所使用的安装包Git-2.8.2-64-bit，安装成功后右键会出现

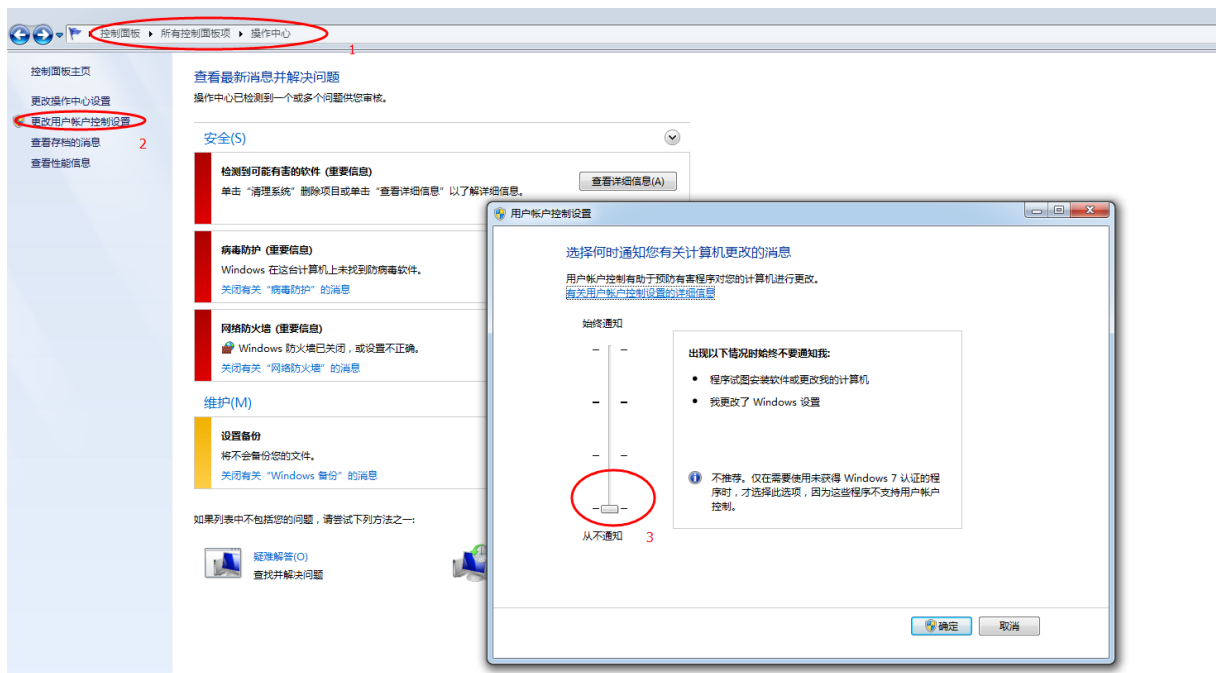
git GUI Here 和 git Bash Here选项



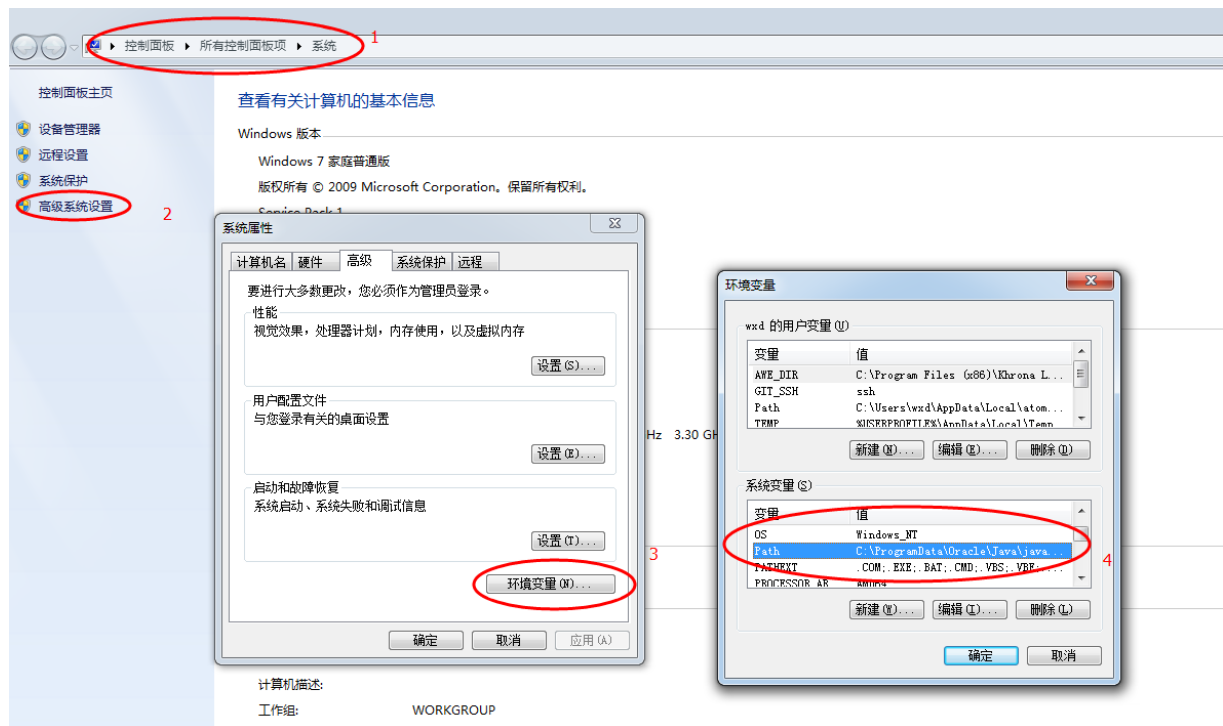
2. 设置git-bash和git-cmd以管理员方式执行



或者



3. 设置环境变量(使用git bash不需要设置)



****选择编辑在末尾加入git安装目录下的执行文件路径 C:\Program Files\Git\bin;C:\Program Files\Git\usr\bin****

4. windows 生成ssh 秘钥

- 右键打开git bash here



- 输入ssh-keygen命令一路默认执行，执行成功后便会在用户的家目录下的.ssh目录中生成公钥文件(id_rsa.pub)和私钥文件(id_rsa)

```
wxd@wxd-PC MINGW64 ~/Desktop
$ ssh-keygen.exe
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/wxd/.ssh/id_rsa):
Created directory '/c/Users/wxd/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/wxd/.ssh/id_rsa.
Your public key has been saved in /c/Users/wxd/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1N/5J/w+UFkSWEA4C8/YFCiuRucSW2vUTzP3oVoZenE wxd@wxd-PC
The key's randomart image is:
+---[RSA 2048]---+
|                |
|  .+O+O.         |
|  .O.+ . . .     |
|  .oB o   .o     |
|  .O...=. . O    |
|  . = .S . = E   |
|  + = . +..O.    |
|  . = . O.+O=..  |
|  . O   .O. . =  |
|  .                |
+---[SHA256]-----+
wxd@wxd-PC MINGW64 ~/Desktop
$
```

私钥

公钥

5. linux安装git

```
yum install -y git
```

6. linux 生成ssh 秘钥

输入ssh-keygen命令一路默认执行，执行成功后便会在用户的家目录下的.ssh目录中生成公钥文件(id_rsa.pub)和私钥文件(id_rsa)

1. git 密钥设置

1.1 生成git ssh公钥

见安装git一节

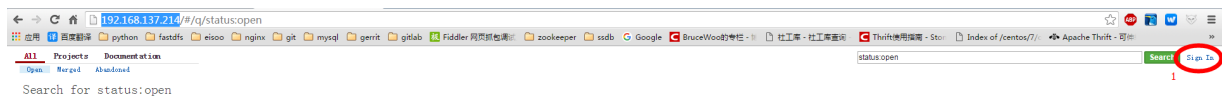
默认用户家目录下的.ssh目录下的id_rsa.pub为公钥文件

1.2 添加ssh公钥

1.2.1 在gerrit服务器中添加ssh公钥

gerrit服务器地址: <http://192.168.137.214/>

- 登录



- 域账号登录

Sign In to Gerrit Code Review at 192.168.137.214

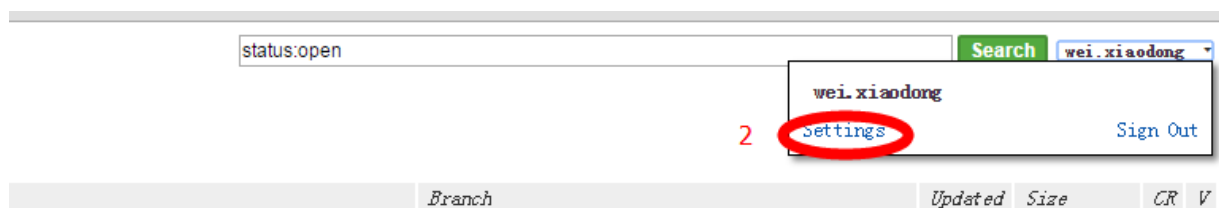
Username

Password

☐ Remember me

[Cancel](#)

- 进入用户设置



- 进入ssh key设置

The screenshot shows the Gerrit web interface. At the top, there are tabs for 'All', 'My', 'Projects', 'People', 'Plugins', and 'Documentation'. Below these are sub-tabs for 'Open', 'Merged', and 'Abandoned'. The main heading is 'Settings'. On the left sidebar, the 'SSH Public Keys' option is highlighted with a red circle and labeled with a red '3'. The main content area shows a table of user information:

Username	wei.xiaodong
Full Name	wei.xiaodong
Email Address	
Registered	Apr 18, 2016 2:12 PM
Account ID	1

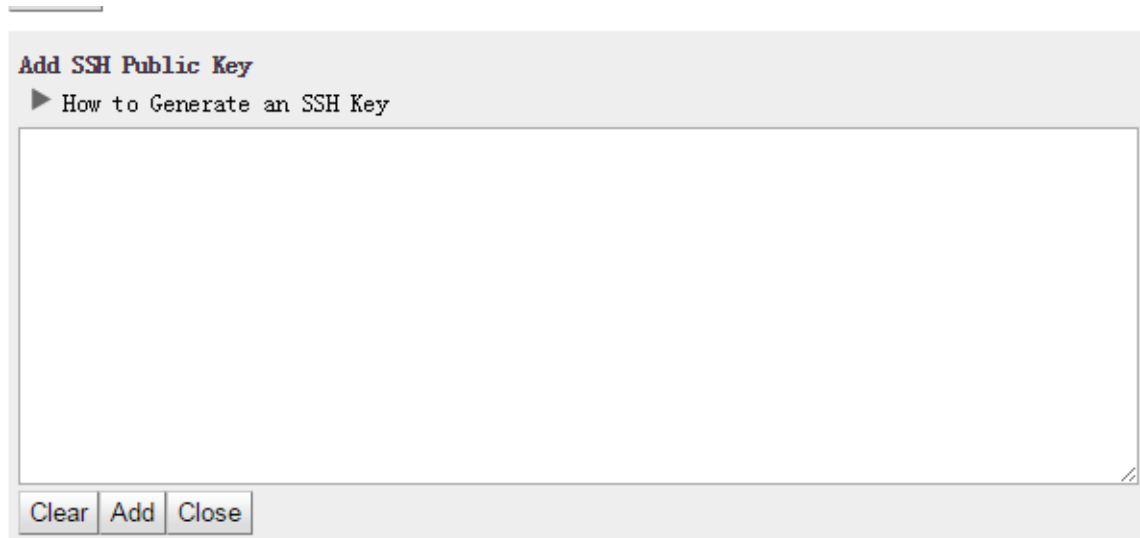
Below the user information, there is a table of SSH Public Keys. The 'Add Key ...' button is highlighted with a red circle and labeled with a red '4'.

Status	Algorithm	Key	Comment
<input type="checkbox"/>	ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAQ...TeXr1C17pF	root@localhost.localdomain
<input type="checkbox"/>	ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAQ...EW89fzB9QP	root@localhost.localdomain
<input type="checkbox"/>	ssh-rsa	AAAAB3NzaC1yc2EAAAABIwAAAQEA2F...BQU9gPPw==	root@localhost.infoworks

Below the table, there is a 'Server Host Key' section showing a fingerprint and an entry for the known_hosts file.

Server Host Key
Fingerprint:
6b:c1:3f:f4:9a:23:34:1e:35:03:ac:5a:4d:45:a7:0e
Entry for ~/.ssh/known_hosts:
[192.168.137.214]:29418 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCBiYtXkR...RV1s+ulw==

- 将生成的id_dsa.pub文件中的内容复制到key文本框里，点击add即可完成gerrit中ssh key的添加



2. git初始化设置（windows需右键打开git bash here）

这是一次性工作，即这些设置会在全局文件（用户主目录下的.gitconfig文件）或系统文件（/etc/gitconfig）做永久保存。

2.1 配置用户的姓名和邮箱地址

```
1. git config --global user.name "wei.xiaodong"      配置用户姓名
2. git config --global user.email "wei.xiaodong@www.com" 配置用户邮箱地址
```

2.2 设置git命令别名，以便使用简洁的命令操作

```
1. git config --global alias.st status
2. git config --global alias.ci commit
3. git config --global alias.co checkout
4. git config --global alias.br branch
```

2.3 开启git命令颜色显示

```
1. git config --global color.ui true
```

2.4 设置git编码

```
1. git config --global core.quotePath false
2. git config --global i18n.logOutputEncoding utf8
3. git config --global i18n.commitEncoding utf8
```

2.5 设置windows下不使用软连接

```
1. git config --global portable.windowsNoSymlinks true
```

2.6 解决windows下文件名过长无法新建的问题

```
1. git config --global core.longpaths true
```

2.7 解决LF will be replaced by CRLF

windows

```
1. git config --global core.autocrlf true
2. git config --global core.autocrlf input
```

把 `core.autocrlf` 设置成input来告诉 Git 在提交>时把CRLF转换成LF，签出时不转换

linux

```
1. git config --global core.autocrlf false
2. git config --global core.autocrlf input
```

1. 使用repo

(windows需右键打开git bash here)

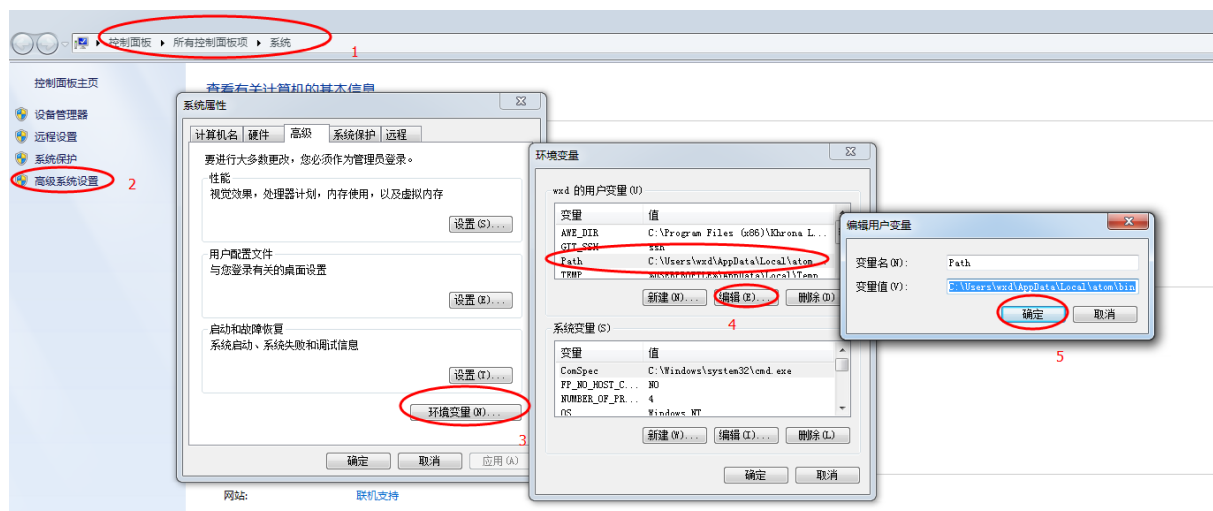
由于我们项目之间的相互依赖太重，所以选择使用repo工具进行管理维护

1.1 下载repo工具

repo文件存放目录 [AnyShare://CMC产品线/Dorado/pShare/应用小组/5.常用软件/repo](#)

1. linux:
2. 将repo文件加入到环境变量中:
3. `cp repo /bin/` (windows请在系统环境变量path中添加)
- 4.
5. 增加repo可执行权限:
6. `chmod +x repo`

1. windows:
2. 将repo文件加入到用户环境变量中:



1.2 使用repo工具获取项目清单

gerrit上的使用(需要将wei.xiaodong账号替换成自己账号)

1. `repo init -u ssh://wei.xiaodong@192.168.137.214:29418/manifest -m platform.xml --repo-url=ssh://wei.xiaodong@192.168.137.214:29418/repo-cmd --repo-branch=master`

gitlab上的使用，暂且不用

1. `repo init -u git@192.168.137.91:root/manifest.git -b gerrit -m platform.xml`

repo init 命令是在当前目录下初始化repo仓库，并获取项目清单，-u参数，指定项目清单仓库，-b参数，指定分支（默认master分支），-m参数指定清单文件名，默认default.xml

1.3 使用repo获取项目

```
1.    repo sync
```

`repo sync` 命令可以将项目清单列表中的项目全部同步到本地或使用该命令获取最新的项目提交

1.4 开始一个项目

```
1.    repo start develop --all
```

`repo start` 开始一个分支，分支名与项目清单中的分支对应，`--all`参数作用于所有项目，也可指定项目列表

1.5 添加修改的项目

```
1.    repo forall -pv -c <git 命令>
```

`repo forall` 表示对所有项目进行迭代，`-p`参数，显示项目名，`-v`参数显示详细信息，`-c`参数指定git 命令

例如：`repo forall -pv -c git add .` 表示对所有项目执行添加操作

`repo forall -pv -c git commit -m "提交信息"` 表示对所有项目进行本地提交

1.6 项目提交推送到代码审核服务器上审核

```
1.    repo upload <project>
```

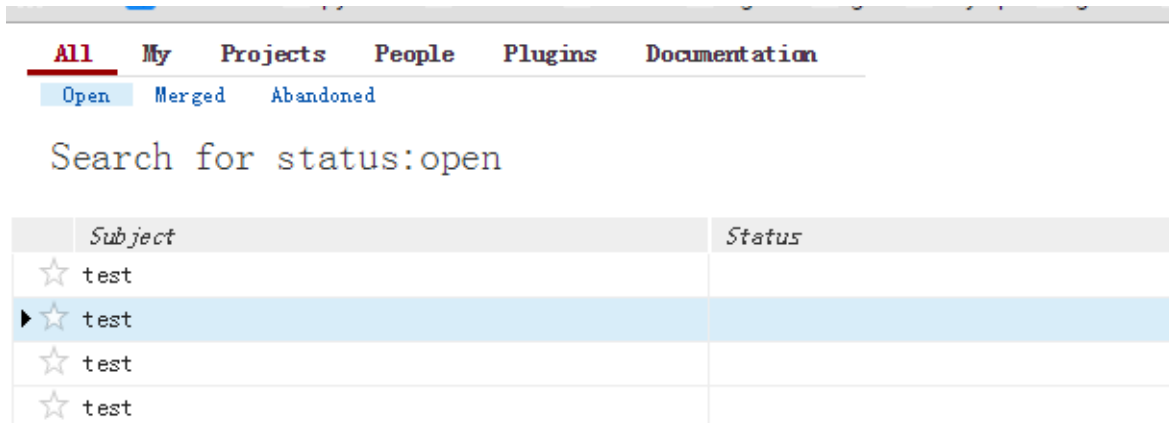
`repo upload` 项目名表示将该项目中的提交推送到代码审核服务器上

例如：`repo upload API` 表示将API项目下的提交推送到代码审核服务器上

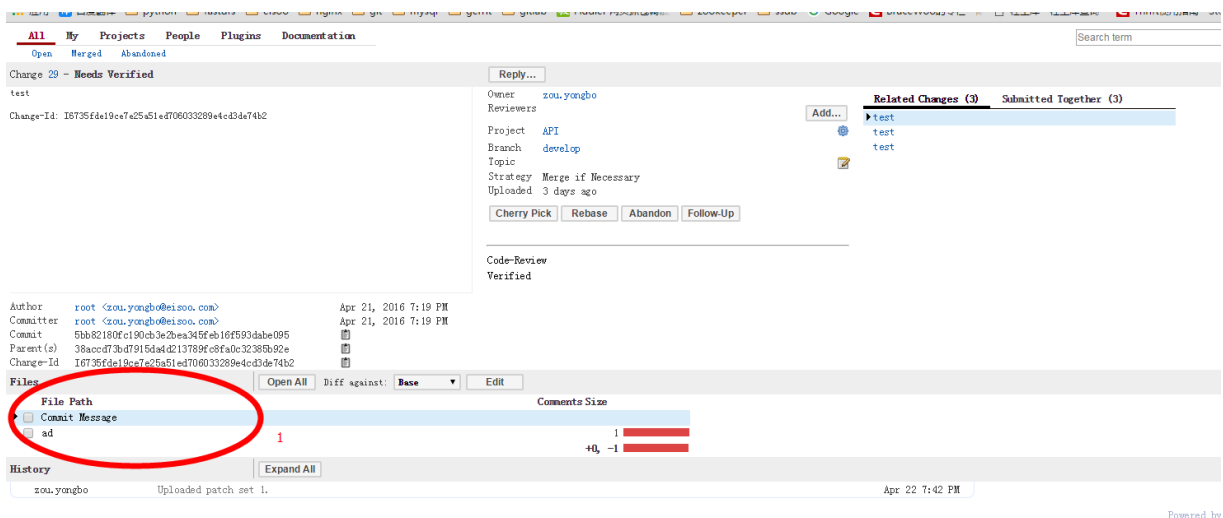
1. 查看需要评审的代码

1. gerrit代码审查服务器地址: <http://192.168.137.214/>

审核人登录gerrit服务器后, 在ALL标签下open选项卡中即可看到所有需要评审的项目, 点击subject列中的连接即可进入



点击filepath即可看到每个文件的详细修改



2. 接受代码提交

代码审查完毕后, 进行回复, **Code-Review +2**表示代码完成的很好, 可以直接归入, **Verified +1**表示验证通过, 可以提交, 代码修改将自动同步到gitlab库中

The screenshot displays the Gerrit web interface for a code review. At the top, there are navigation tabs: All, My, Projects, People, Plugins, and Documentation. Below these are sub-tabs: Changes, Drafts, Draft Comments, Edits, Watched Changes, Starred Changes, and Groups. The main content area shows a change titled 'test' with a status of 'Needs Verified'. A 'Reply...' button is circled in red. Below the change title, there is a table with columns for 'Changes (3)' and 'Sub'. The table contains one row with a score of '+2' for 'Code-Review' and '+1' for 'Verified', both by 'wei.xiaodong'. The 'Submit' button is also circled in red. Below the change details, there are buttons for 'Cherry Pick', 'Rebase', 'Abandon', 'Follow-Up', and 'Submit'. The 'Submit' button is circled in red. Below the buttons, there is a table with columns for 'Code-Review' and 'Verified'. The table contains one row with a score of '+2' for 'Code-Review' and '+1' for 'Verified', both by 'wei.xiaodong'. The 'Abandon' button is circled in red.

Change 29 - Needs Verified

test

Change-Id: I6735fde19ce7e25a51ed706033289e4cd3de74b2

Author root <zou.yongbo@eisoo.com> Apr 21, 2016 7:19 PM

Committer root <zou.yongbo@eisoo.com> Apr 21, 2016 7:19 PM

Commit 5bb82180fc190cb3e2bea345feb16f593dabe095

Parent(s) 38acc073bd7915dad213789fc8fa0c32385b92e

Change-Id T6735fde19ce7e25a51ed706033289e4cd3de74b2

Project API

Branch develop

Topic

Strategy Merge if Necessary

Updated in the future

Cherry Pick Rebase Abandon Follow-Up Submit

Code-Review +2 wei.xiaodong 2

Verified +1 wei.xiaodong 3

3. 拒绝提交

The screenshot displays the Gerrit web interface for a code review. At the top, there are navigation tabs: All, My, Projects, People, Plugins, and Documentation. Below these are sub-tabs: Changes, Drafts, Draft Comments, Edits, Watched Changes, Starred Changes, and Groups. The main content area shows a change titled 'test' with a status of 'Needs Verified'. A 'Reply...' button is circled in red. Below the change title, there is a table with columns for 'Changes (3)' and 'Sub'. The table contains one row with a score of '+2' for 'Code-Review' and '+1' for 'Verified', both by 'wei.xiaodong'. The 'Submit' button is also circled in red. Below the change details, there are buttons for 'Cherry Pick', 'Rebase', 'Abandon', 'Follow-Up', and 'Submit'. The 'Abandon' button is circled in red. Below the buttons, there is a table with columns for 'Code-Review' and 'Verified'. The table contains one row with a score of '+2' for 'Code-Review' and '+1' for 'Verified', both by 'wei.xiaodong'. The 'Abandon' button is circled in red.

project API

Branch develop

Topic

Strategy Merge if Necessary

Uploaded 7 minutes ago

Cherry Pick Rebase Abandon Follow-Up

Code-Review

Verified

Abandon 2

1. git 分支管理

Created with Raphaël 2.1.4git分支协作图

Created with Raphaël 2.1.4develop(研发线) develop(研发线) master(版本发布线) master(版本发布线) at和st人员 develop线上的代码合并到master分支 st人员测试发布版本 master线 bug修复向develop合并

2. 协作开发

Created with Raphaël 2.1.4git研发协作图（针对develop分支）

Created with Raphaël 2.1.4developer(研发人员) developer(研发人员) gerrit(代码审核服务器) gerrit(代码审核服务器) gitlab(代码管理服务器) gitlab(代码管理服务器) 使用repo工具获取代码 研发人员完成一项特性 gerrit代码审核审核通过，自动同步 at和st人员测试

3. 发布线

Created with Raphaël 2.1.4git发布版本（针对master分支）

Created with Raphaël 2.1.4develop(研发线) develop(研发线) master(版本发布线) master(版本发布线) 向master分支合并 st人员测试 bug修复合并到develop分支上