在 Windows 下用 NetBeans 做 PHP 开发，首先想到的是 NetBeans 的插件：NBGit。  
评价：能用；若需没有的功能，可以自定义菜单调用自定义 bat 脚本；开发不活跃，使用没有信心。

第二个则是：TortoiseGit，SVN 小乌龟的 Git 版本。  
评价：该有的功能基本都有了，还是不错的。

另外，TortoiseGit 只是 GUI 工具，使用它需要先安装 msysGit，这是正宗的 Git 之 Windows 版本。msysGit 有个简单的 GUI 工具，及简单的 Explorer 集成；但它自带的 Bash 非常好用，深得 Linux 的真传。

**选择：msysGit。**  
理由：  
NBGit 不用说，功能都不完善，还需要自己定制 bat 脚本（若此，则它同样要依赖 msysGit）；开发不活跃，很可能 NetBeans 下个版本更新就不能用了；况且，我们还有别的项目，不使用 NetBeans。

TortoiseGit 从功能上说是完善的，但它只是功能的堆砌而已，使用时完全体会不到 GUI 带来的便利。相反，它让人感觉很繁琐，一个劲地点鼠标，点来点去全是跟菜单打交道，远离了 Git 命令、远离了 Git 输出提示、远离了真相。

msysGit 的 Bash 非常好用；加上 Git 强大的 alias 功能，我们完全可以自定义一个 $ git go，使得 90% 的情况下只需要这一个命令，即使是不熟悉命令行的 Windows 用户也会觉得很好玩；因为 NBGit、TortoiseGit 都需要 msysGit 做底层，我们直接用底层工具也避免了上层 GUI 带来的额外的 bug。

需要的配置：

**1、**C:\Program Files\Git\etc\git-completion.bash：

**alias ls='ls --show-control-chars --color=auto'**

说明：使得在 Git Bash 中输入 ls 命令，可以正常显示中文**文件名**。

**2、**C:\Program Files\Git\etc\inputrc：

**set output-meta on**

**set convert-meta off**

说明：使得在 Git Bash 中可以正常**输入**中文，比如中文的 commit log。

**3、**C:\Program Files\Git\etc\profile：

**export LESSCHARSET=utf-8**

说明：$ git log 命令不像其它 vcs 一样，n 条 log 从头滚到底，它会恰当地停在第一页，按 space 键再往后翻页。这是通过将 log 送给 less 处理实现的。以上即是设置 less 的字符编码，使得 $ git log 可以正常显示中文。其实，它的值不一定要设置为 utf-8，比如 latin1 也可以……。还有个办法是 $ git –no-pager log，在选项里禁止分页，则无需设置上面的选项。

**4、**C:\Program Files\Git\etc\gitconfig：  
**[alias]  
go =** “! bash -c \”git pull && git add .; if [ \\\"$\*\\\" == \\\"\\\" ]; then git commit -a; else git commit -am \\\”$\*\\\”; fi; git push origin master:your-id;\”"

说明：强大的 alias，有了这个，我们 90% 的情况下只需要输入 $ git go 这一个命令，免去了先拉后提交再推的繁琐步骤。

两种用法：  
$ git go  
或  
$ git go aaa 修订说明

命令后带修订说明时，会直接提交。需要注意的是，在“修订说明”之前，有还个“aaa”，这是个 bug，参数中的第一个会被忽略，所以随便写一个凑数的……。

若命令行里没有提供修订说明，则会自动弹出一个编辑器，等待输入。默认的编辑器是 Vim。Vim 的使用是很简单的，首先要明白它有两个模式，一个是命令模式、一个是输入模式。Vim 启动的时候默认的是命令模式，需要先按’i'键，进入输入模式；然后就正常编辑；编辑完成之后，将输入法切换回英文状态，按 Esc 重新进入命令模式；此时按 ‘**(Shift):wq**‘ 并回车，w 表示写入保存、q 表示退出。完毕！

若实在不习惯 Vim，也可以设置为其它编辑器：

$ git config --global core.editor "notepad"

其中 notepad 可以替换为更好用的 wordpad、notepad++ 等（不过它们在命令行里无法直接访问，得先设置 PATH 变量）。

以上 alias 是为 Windows 定制的，Linux 下可以写得更优雅，不过鉴于使用上没分别，就保持一致吧～。

**[gui]  
encoding = utf-8**

说明：我们的代码库是统一用的 utf-8，这样设置可以在 git gui 中正常显示代码中的中文。

**[i18n]  
commitencoding = GB2312**

说明：如果没有这一条，虽然我们在本地用 $ git log 看自己的中文修订没问题，但，一、我们的 log 推到服务器后会变成乱码；二、别人在 Linux 下推的中文 log 我们 pull 过来之后看起来也是乱码。这是因为，我们的 commit log 会被先存放在项目的 .git/COMMIT\_EDITMSG 文件中；在中文 Windows 里，新建文件用的是 GB2312 的编码；但是 Git 不知道，当成默认的 utf-8 的送出去了，所以就乱码了。有了这条之后，Git 会先将其转换成 utf-8，再发出去，于是就没问题了。

**以上，给 Windows 下的同事在 Git Bash 里推代码就比较完美了。不过仍然有 3 个问题：**

1、上面的 alias $ git go 有 bug，代码修订说明之前要输入一串字符凑数；

2、$ git diff，如果代码里有中文，会显示乱码；

3、$ git checkout 有时候需要修改/增删很多文件，如果某些文件被占用，会被 Windows 拒绝，导致失败，甚至可能造成版本库出现无法修复的问题。

这 3 个都是可承受的问题，前两个应该有办法解决；第 3 个归功于文件系统，只能尽量避免 checkout，实在需要的时候先注销一次，就不会有问题了。

**【TIP】**该文只是解释说明，具体操作请按[《Windows 下 git 配置与使用指南》Wiki](http://wangcongming.info/2010/07/31/git-%E7%B3%BB%E5%88%97%E4%B9%8B%E4%B8%89%EF%BC%9Awindows-%E4%B8%8B-git-%E9%85%8D%E7%BD%AE%E4%B8%8E%E4%BD%BF%E7%94%A8%E6%8C%87%E5%8D%97/) 执行。