版本控制之道

几乎所有项目[61]，都要使用版本控制，它究竟有什么优势呢？

时间机器

假设你使用的编辑器，不支持删除，那你就得特别的谨小慎微，甚至是如履薄冰：因为你打错了字没法删除

放松下来，目前我所接触的所有编译器中，还没有变态到这种程度的。

如果编译器提供了删除功能，却没有 undo，那可能会更可怕：如果你不小心选中了全部文字，手一抖……因为不能 undo，你知道，如果此时不小心按下 delete，你就得从头来过……你会为可能产生的后果而发抖

然而，命运总在你不想被打扰的时候来敲你可爱的门，在你手抖的刹那，你真的鬼使神差的按了下那个可怕的键……你的选择只能是重新来过或者放弃……

你会比任何时候都希望时间倒流一秒钟

还好，几乎所有的编译器，都会帮你回到一秒种、一分钟、一小时……之前

假设你想回到一天之前？一觉醒来，你突然想起，有一部分内容其实应该保留下来……但是编辑器在重启之后，就不能够再帮你回到以前的任何时间……

这种情况下，版本控制才是你的救命稻草

分支控制

如果项目只能朝一个方向发展，你会时常在确定方向的问题上犹豫不决。而使用版本控制，创建一个分支各自发展，在适当的时候合并分支，是最好的解决办法

协同工作

很多项目需要协同工作，版本控制能够提供协同工作需要的环境，解决协同工作可能产生的问题

冲突合并

项目成员可能会对一处内容进行不同的修改，版本控制能够反馈这些冲突，以便解决它

常见版本控制系统

传统的版本控制系统，需要在服务器上搭建一个中央仓库，所有成员都是客户端，具有不同的权限。这种方式适应大教堂开发模式，但比较依赖网络，且不够灵活，使用比较广泛的有cvs、svn等

而分布式版本控制系统，则不需要中央仓库，所有的成员都可以完全掌控己方的版本控制系统，通过多种方式灵活的协作。这种方式适应集市开发模式，典型的代表是git

---

[61] 项目并不总是意味着开发一个大型软件；你帮领导打一份文件，其实也是项目