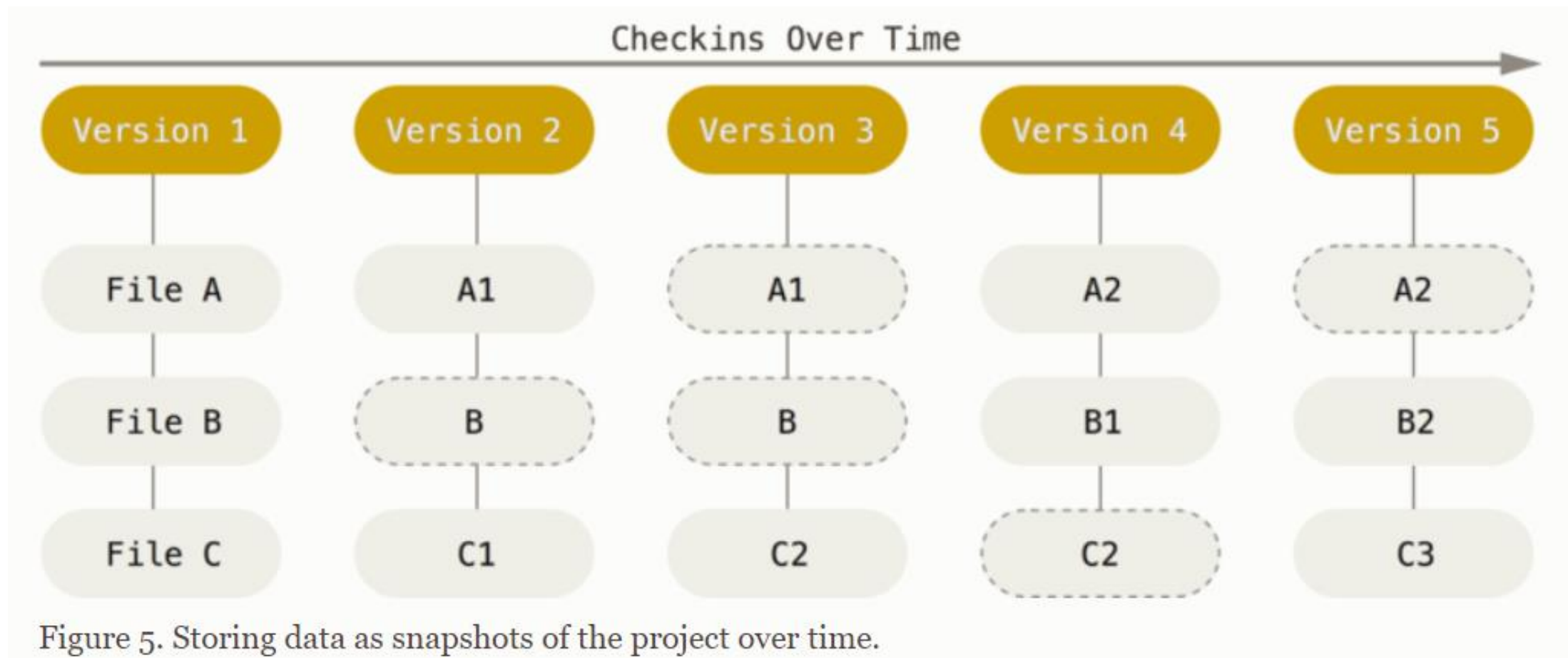




**01**

**How to use git opportunely.**

## 1.3 Snapshots, Not Differences



### 1.3 Three States : modified, staged, and committed:

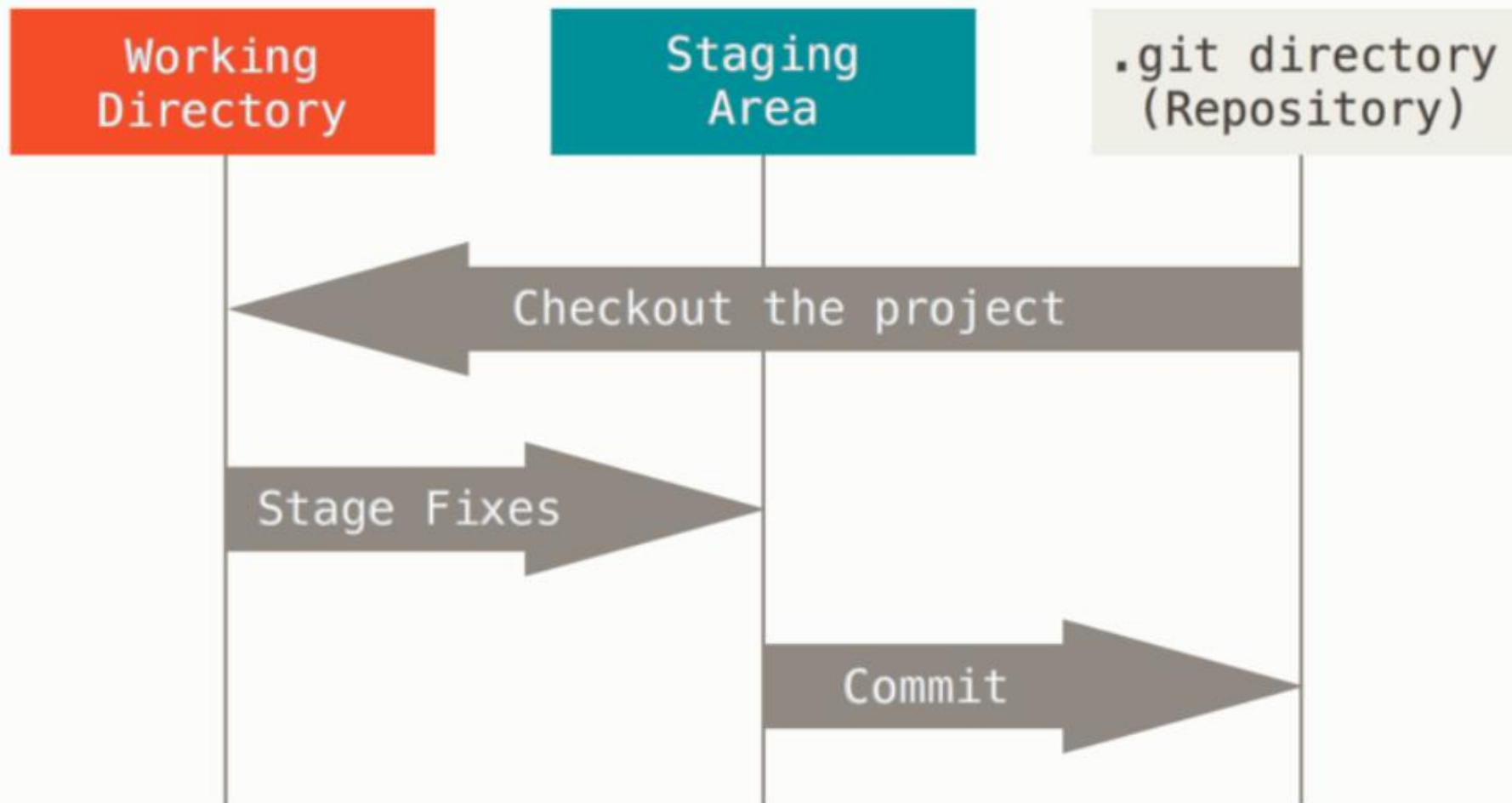


Figure 6. Working tree, staging area, and Git directory.

## 1.5 Git Installing

```
MINGW64:/c:/Users/zhoujianwen/Desktop

zhoujianwen@sz18725_02 MINGW64 ~/Desktop
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

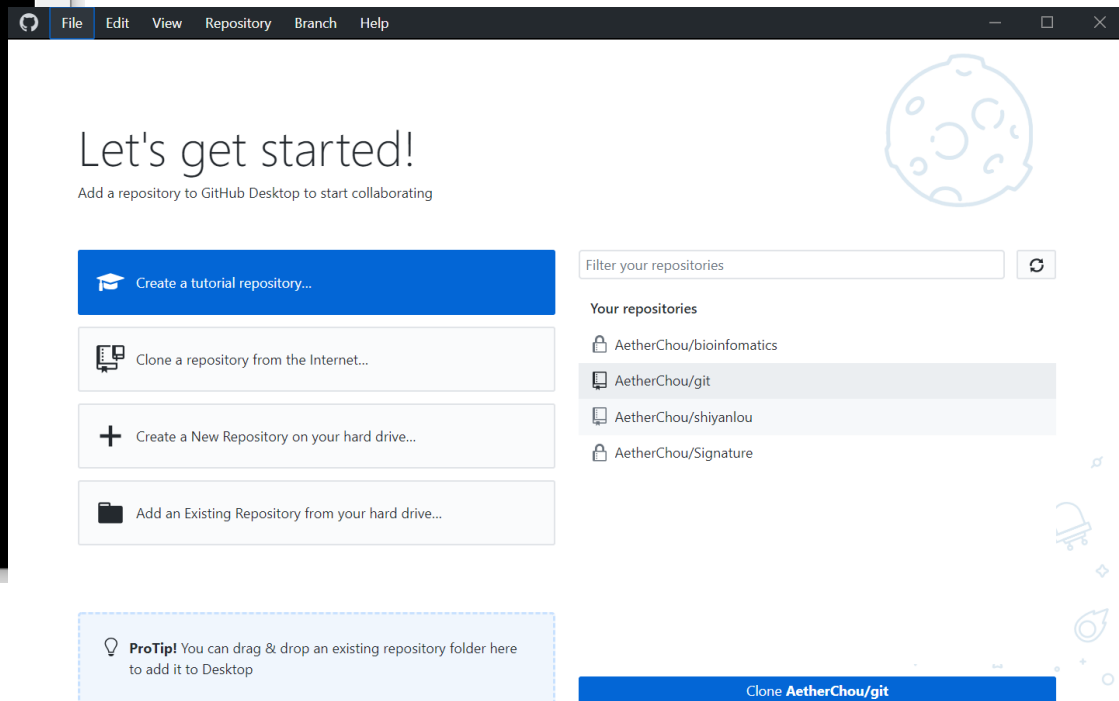
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
```

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



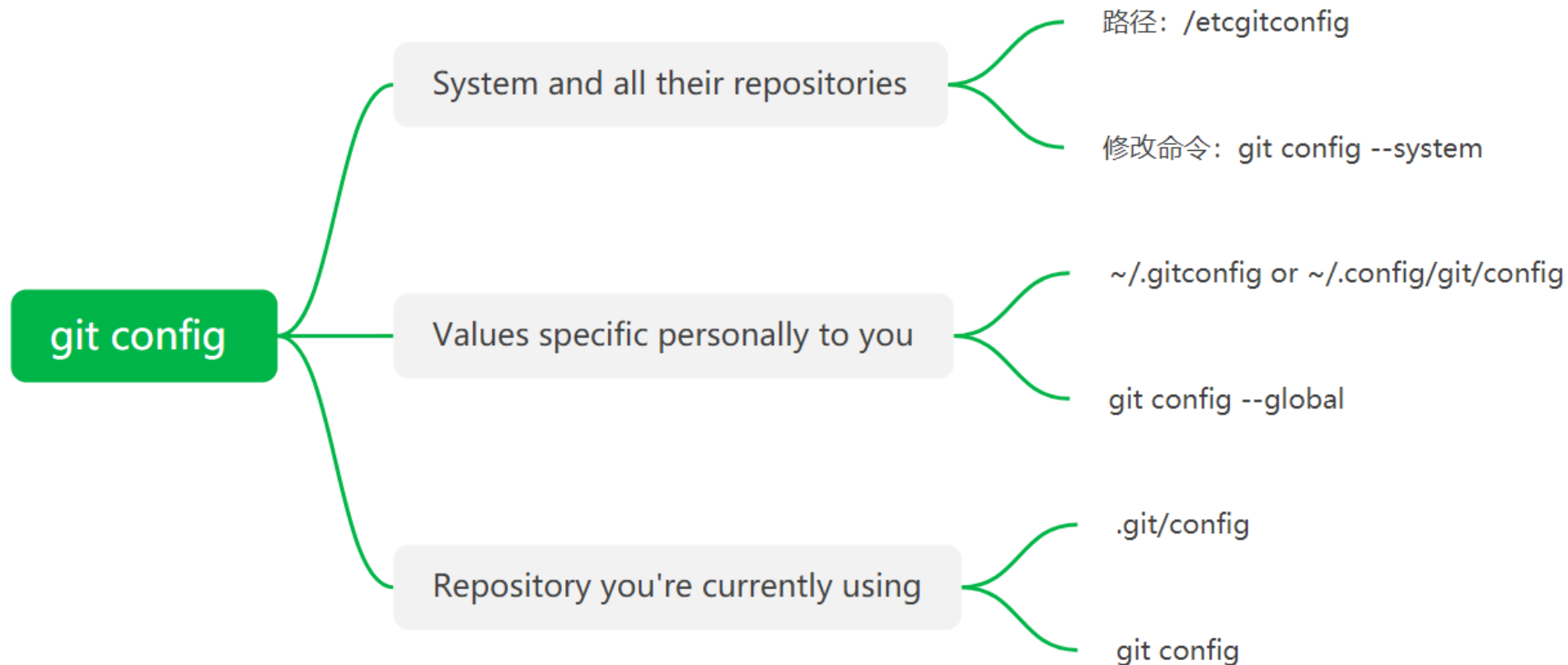
<https://desktop.github.com/>

## 2.1 Getting a Git Repository

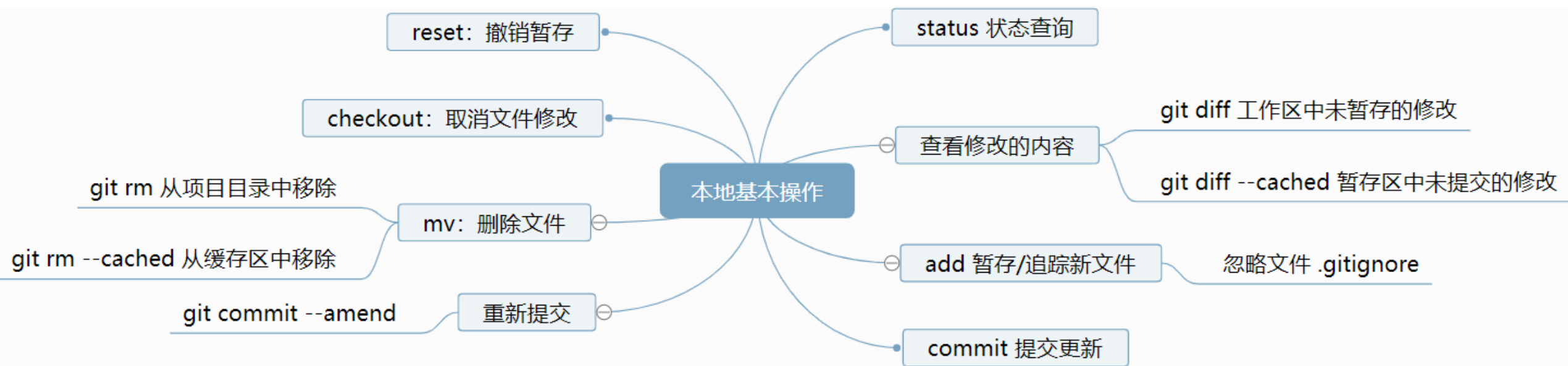
```
# Initializing a Repository in an Existing Directory  
$ git init  
  
# Cloning an Existing Repository  
$ git clone https://github.com/libgit2/libgit2
```

## 1.6 Git Setup – Config

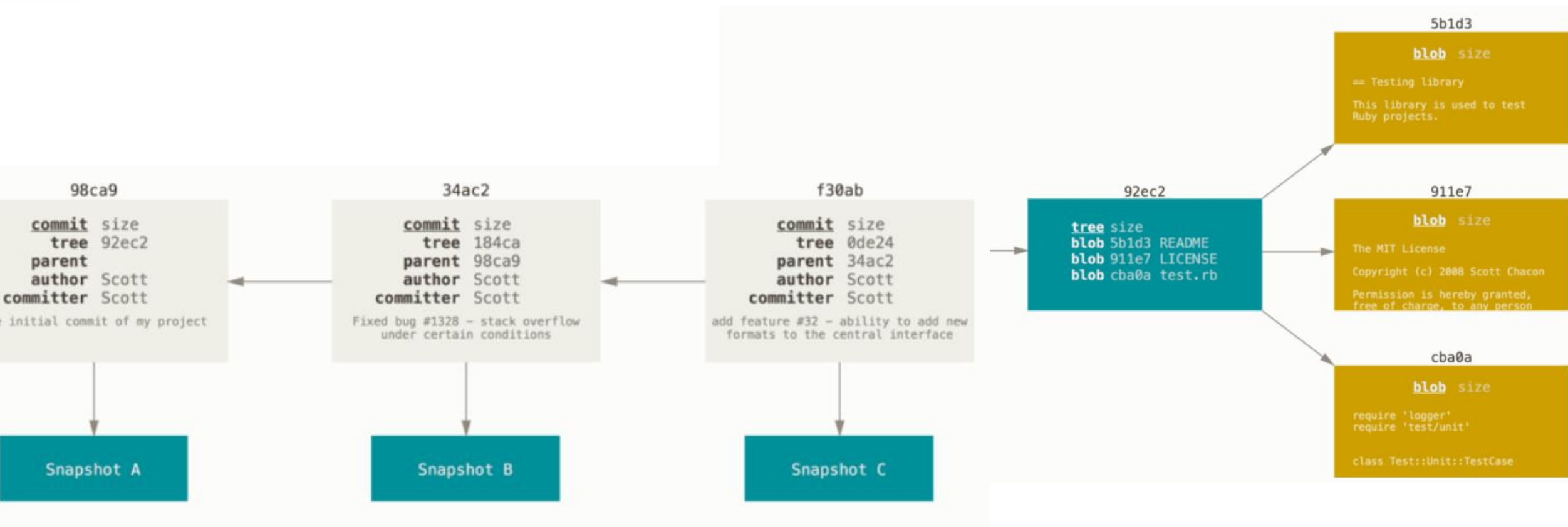
```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```



## 2.2 Recording Changes to the Repository

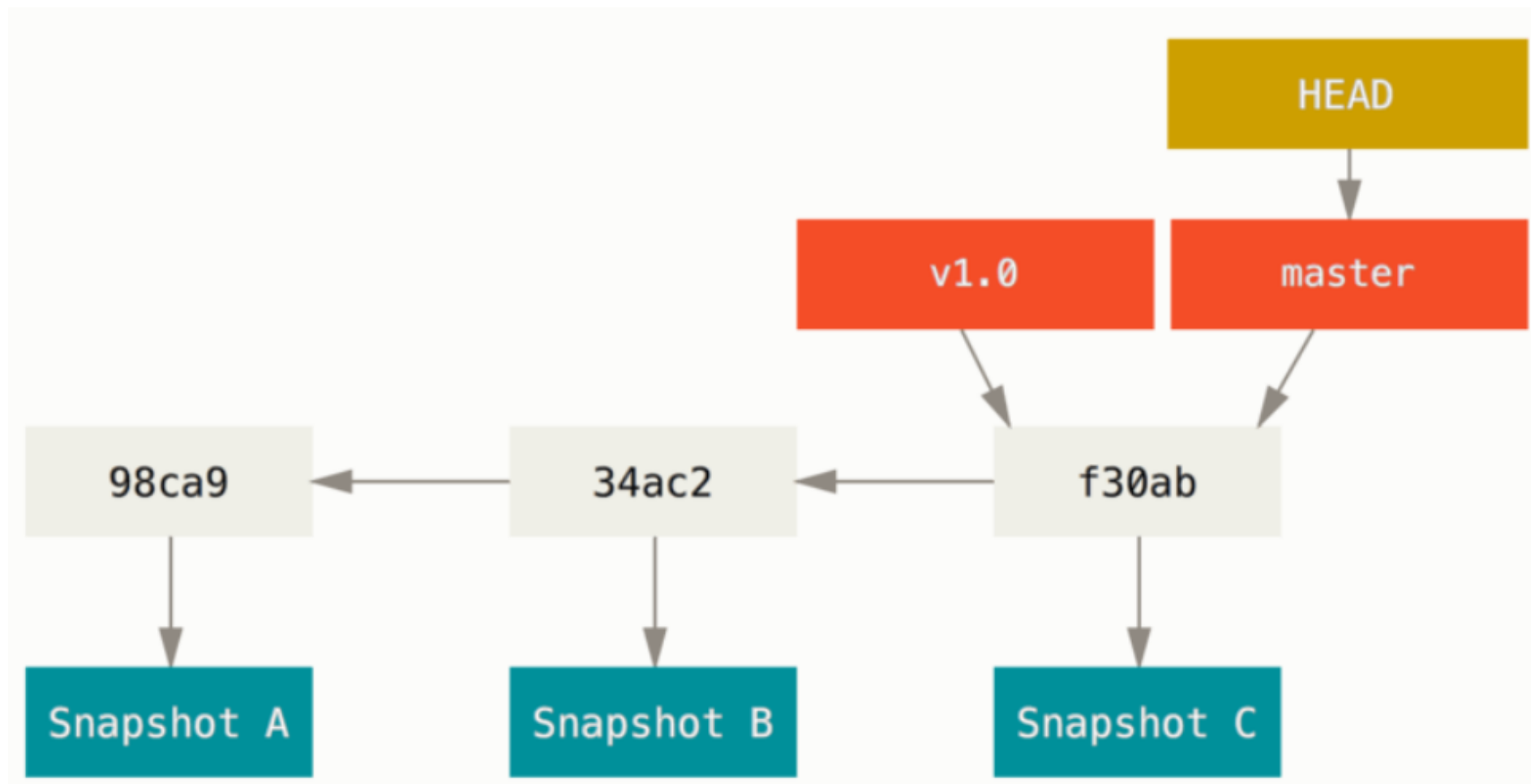


## 3.1 Branching : Commits and their parents

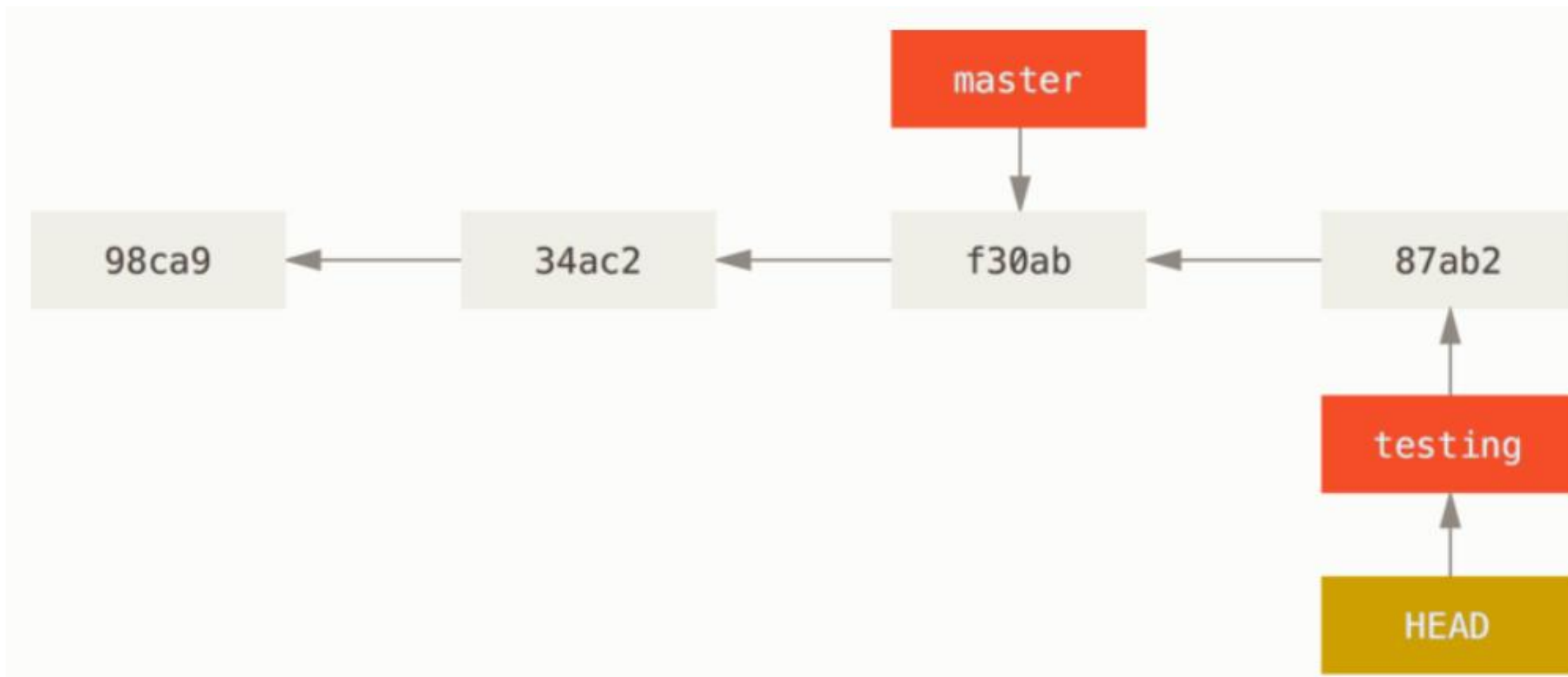




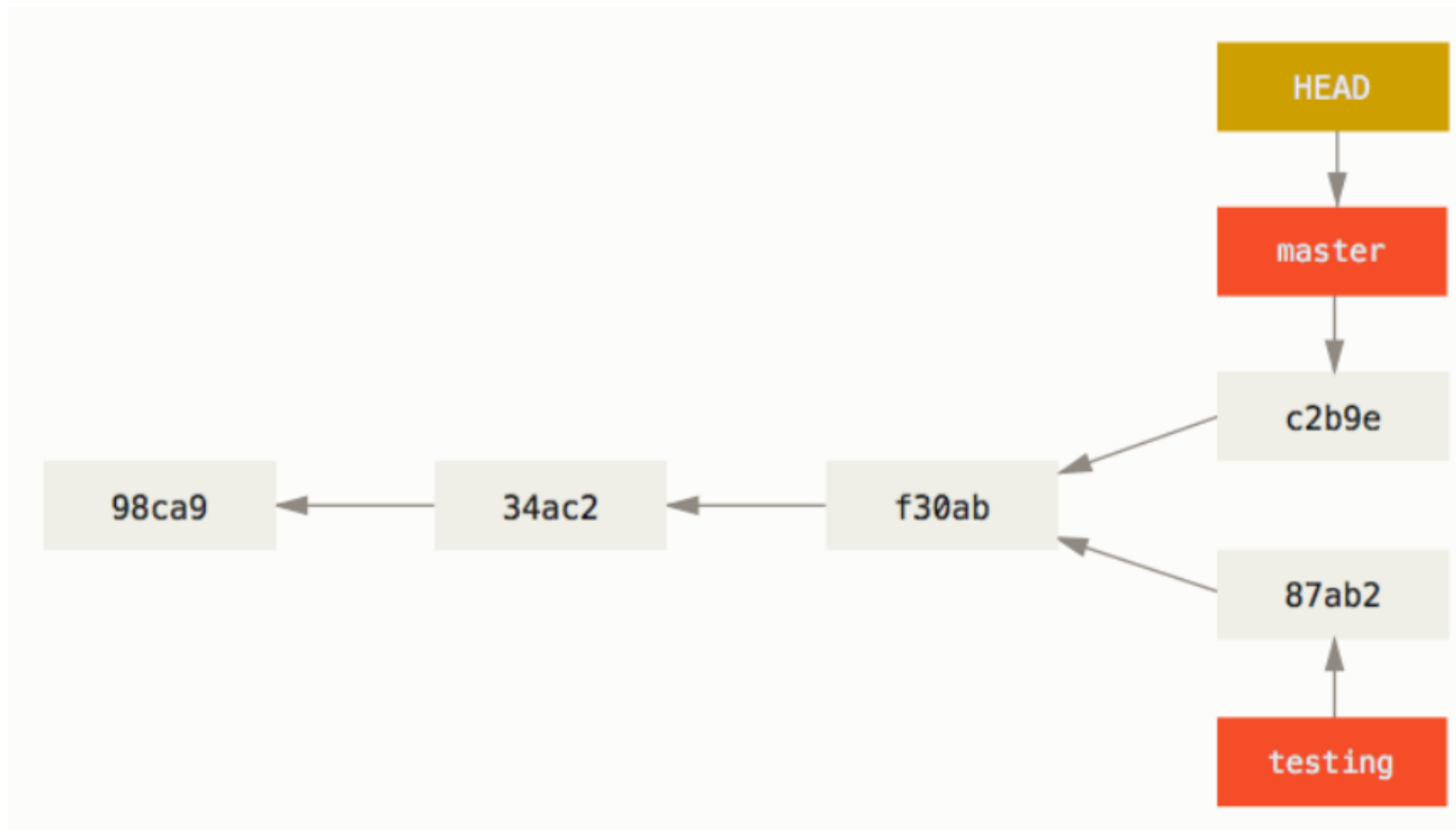
## 3.1 Branching



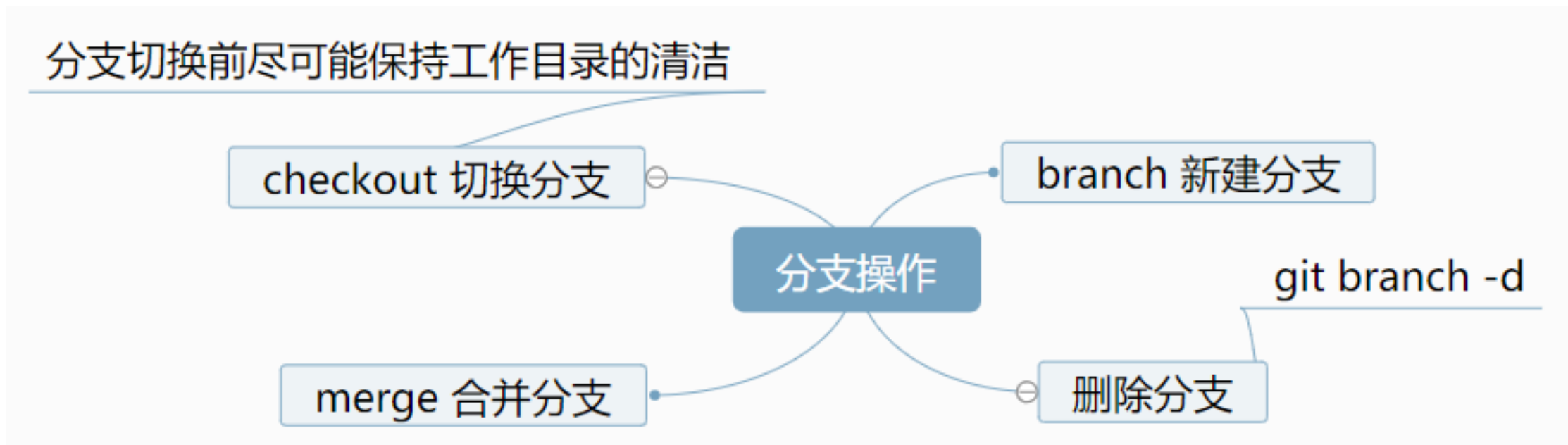
## 3.1 Branching



## 3.1 Branching



## 3.2 Basic Branching and Merge



利用Branching实现“临床流程”的合作修改

技术的日新月异，很难有一门具体的编程语法能通过熟练而积攒下来。而贯穿整个程序员职业生涯，能够沉淀下来，愈久弥香的东西，一定是超越具体语言，形而上的东西。编程水平的提高，伴随程序员整个职业生涯，都是一个延绵不绝，不进则退的过程。这既是程序员职业的痛苦所在，也是它的魅力所在。

——沈世钧