

# 规范化运营开发

夏永锋

# 大纲

- 规范化的必要性
- 版本控制
- 测试环境
- 项目文档
- 编码规范
- 人
- 问题与讨论

# 规范化的必要性

- 避免低级错误
- 应对突发情况
- 多人协作的前提
- 规范化与工作效率

# 版本控制(1)

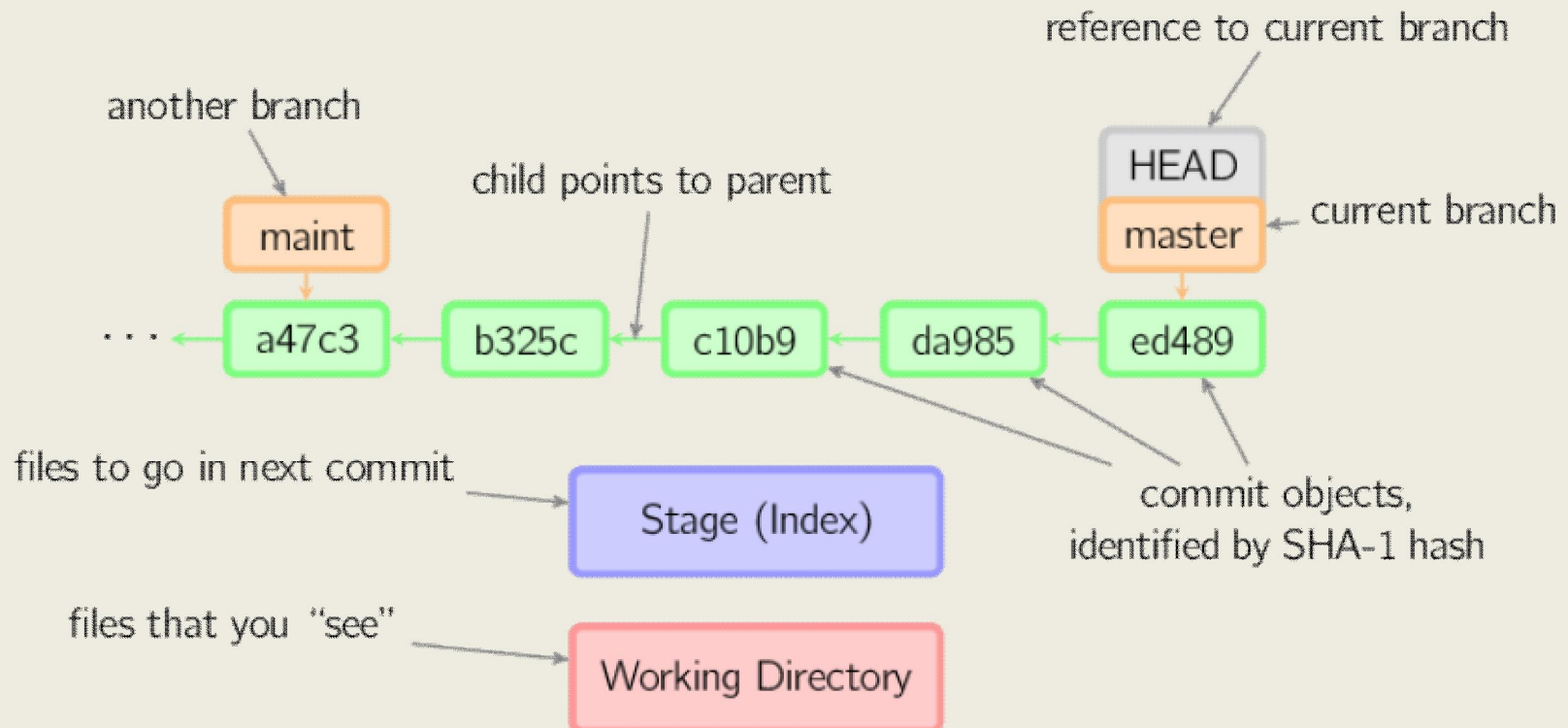
- 如何看待版本控制工具？
  - 代码备份存储？
  - 变更记录
  - 助力多人协作，提高效率
  - 自动化基础工具

# 版本控制(2)

- Git VS. Subversion
- 分布式版本控制的优势？
  - 没有中心，处处是版本库
  - 尽可能避免污染共享库

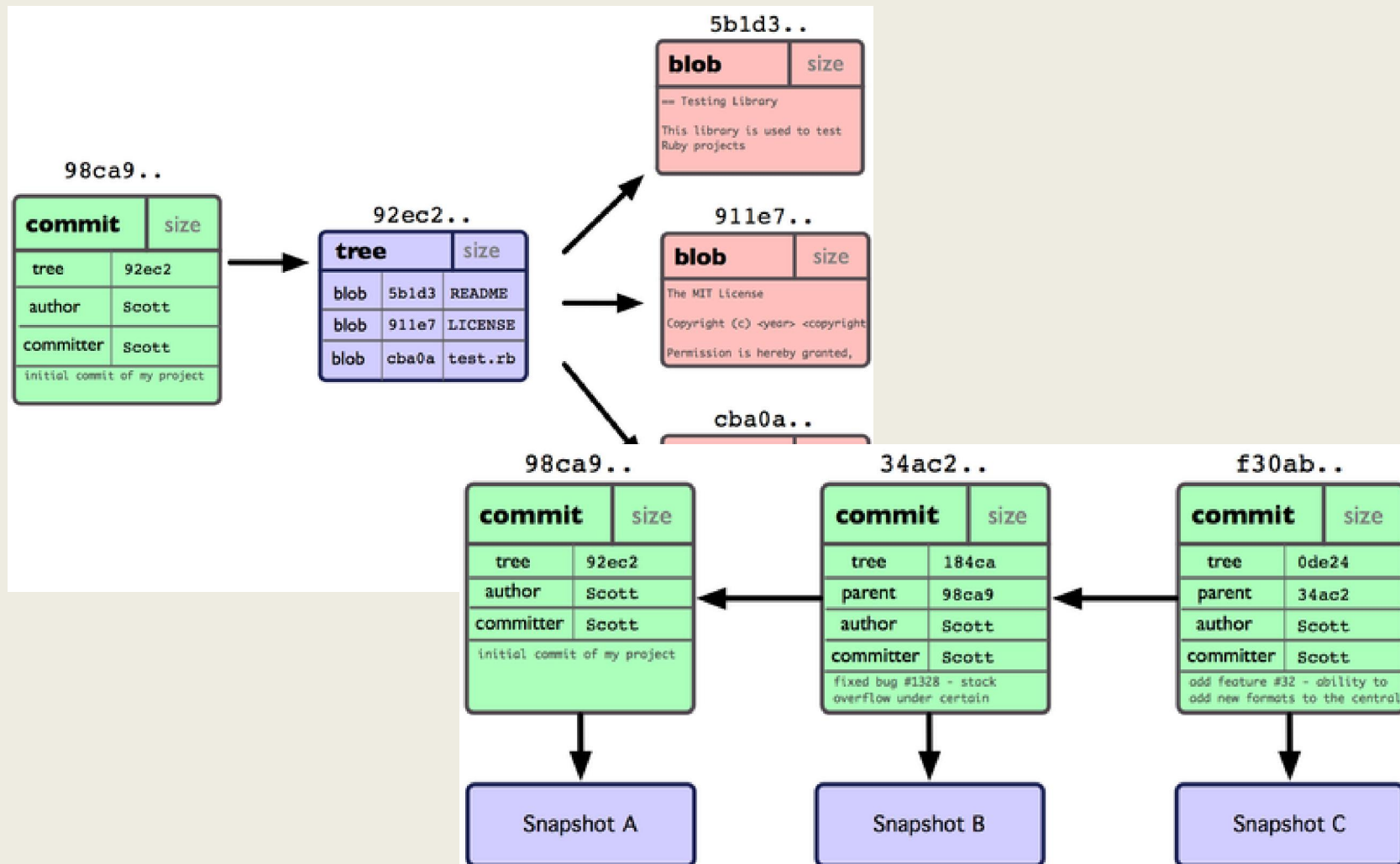
# 版本控制(3)

## •Git基本概念图



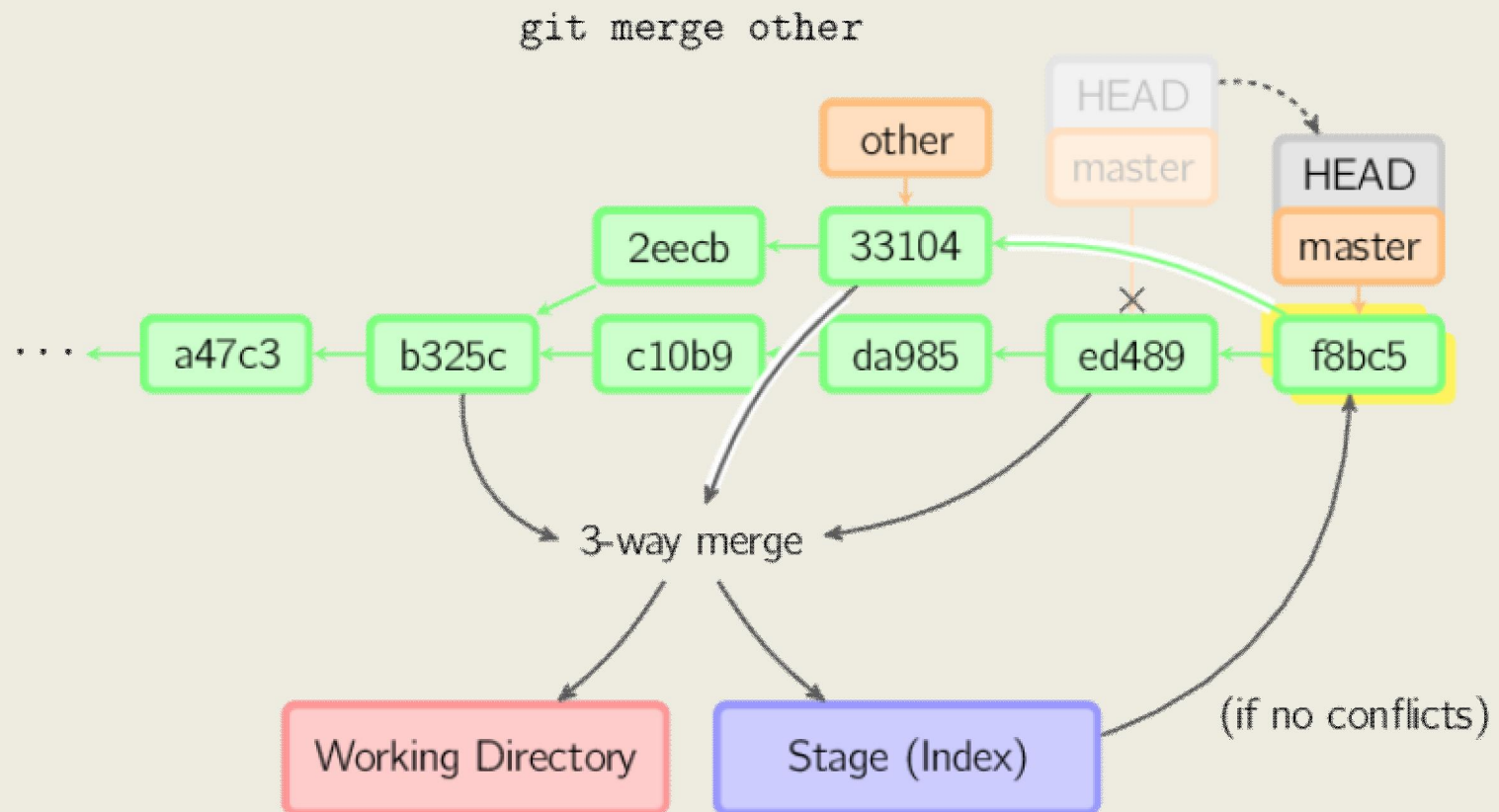
# 版本控制(4)

## •Git的三种对象



# 版本控制(5)

## •Git分支合并





# 版本控制(6)

- Git贴士

- Git帮助文档: `git/git help`, `git help [cmd]`
- 设置用户名和邮箱: `git config --global user.name xxx`, `git config --global user.email xxx`
- 彩色输出: `git config --global color.ui true`
- 图形化Git: `gitk`
- 注意错误提示

# 版本控制(7)

- 最佳实践:  
<http://www.oschina.net/translate/version-control-best-practices>
- 图文教程:  
<http://marklodato.github.io/visual-git-guide/index-zh-cn.html>
- Pro Git

# 版本控制(8)

- 开发流程:

git branch (当前应在master中)

git checkout -b yongfengxia (创建yongfengxia分支)

modify && git add xxx && git commit -m "xxx"

git checkout develop

git merge yongfengxia

git commit -m "xxx"

git push origin develop

测试后需上线

git branch (当前应在master分支中)

git merge yongfengxia

git commit -m "xxx"

git push origin master

代码已合进master稳定分支

生产服务器从master分支拉取代码, 上线

git branch -D yongfengxia

# 版本控制(9)

- 开发流程解决的问题？
- 一定要严格遵守开发流程么？

# 版本控制(10)

- `.gitignore`
- 已进入版本控制的文件无法被`ignore`掉。
- 配置文件的问题(`ignore`掉或切记不要提交本地机器的配置文件)

# 版本控制(11)

- 服务器代码迁入Git版本控制的过程

# 版本控制(12)

- Gitlab代码托管平台
  - 友好安全便利地管理代码与版本信息
  - 代码 + wiki文档
  - Code review(正在实现基于该功能的code review任务自动打包与分发)

# 测试环境

- 源码编译：Nginx、PHP、MySQL、Memcached、Redis、Python、Ruby、Git、...
  - 总结见[文档](#)，个人认为该文档含金量较高
- Git Hook实现自动部署测试代码
- 问题：数据源？如何解决？
- 配置文件版本冲突
- 进一步的工作：Git Hook异步处理



# 项目文档(1)

- 工作的整理
- 方便别人快速了解你的工作
- 保证一致性（如工程结构）
- 项目移交
- 文档

# 项目文档(2)

- 编程建议
  - 代码上线之前，务必先删除调试语句。
  - 清楚地理解你写下的每行代码。
  - 一份代码“写三遍”。
  - 恰当处理cron脚本/API文件。
  - 前端页面代码尽可能分离HTML、CSS、JS。
  - DRY（Don't Repeat Yourself）。
  - 尽可能避免使用依赖特定操作系统、文件系统的操作。

# 编码规范

- 前端规范（JS、CSS、HTML）
  - [Google HTML/CSS Style Guide](#)
  - [Google JavaScript 编码规范指南](#)
  - [jQuery JavaScript Style Guide](#)
  - [Kissy CSS编码规范](#)
  - [Kissy HTML编码规范](#)
- 后端规范（PHP）：语言层面、框架层面
  - [PHP之道](#)
  - [安居客PHP源代码规范](#)
  - [PHP最佳实践](#)
- 关键是**统一**！

# 人

- 人(开发人员)是项目的行为主体。流程、规范的实施全依赖于人!
- 开发人员的追求：更快、更好、更强!

# 问题与讨论

谢谢！