



GitLab源码管理工具使用培训

2021/12/6

*****有限公司

面向对象：项目负责/研发/测试/运维支持

01. 初识gitlab——始于源码、不止于源码（3页）
02. gitlab与svn的区别（4页）
03. gitlab中的团队、仓库、用户、角色（4页）
04. gitlab中的分支（2页）
05. 如何在gitlab中提交代码？（3页）
06. 如何在gitlab中提交bug？（4页）
07. 如何在gitlab中管理文档/共享资料？（2页）
08. 如何在gitlab中分配任务（to-do list）（4页）？
09. 如何在gitlab中发布版本？（3页）
10. 如何在gitlab中互动交流？（2页）
- 11.gitlab使用价值：数据互通/过程可复盘/责任明确（1页）
- 12.gitlab操作终端：Web界面/命令行/客户端/IDE插件（1页）

01. 初识gitlab——始于源码、不止于源码 (1/3)



源码管理工具应该必备哪些功能？



Commit

Revert

Branch

Diff

Merge

log

...

CheckOut

源码层面



01. 初识gitlab——始于源码、不止于源码 (2/3)



那么gitlab有哪些功能？



Milestone

CI/CD

Code
Review

Commit

Revert

Branch

tag

To-do
list

Forum

Diff

Merge

log

...

Analyse

Bug

Release

Wiki

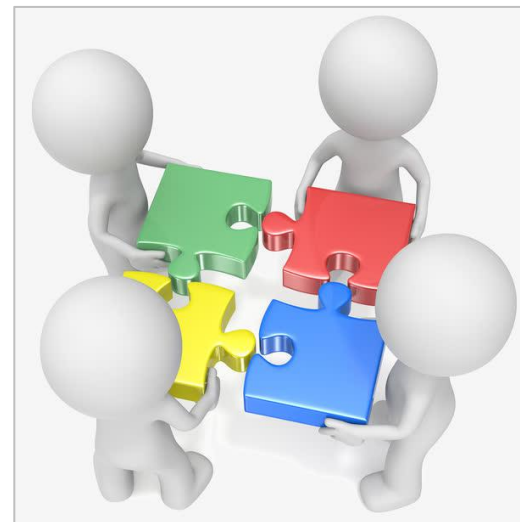
CheckOut



01. 初识gitlab——始于源码、不止于源码 (3/3)



```
265  /**
266   * 初始化
267   * @param callback 数据分发回调
268   * @return
269   */
270  bool FFmpegStreamReceive::init
271  {
272      bRunning = false;
273      this->callback = cb;
274      av_log_set_level(AV_LOG_FA
275      return true;
276  }
277
```



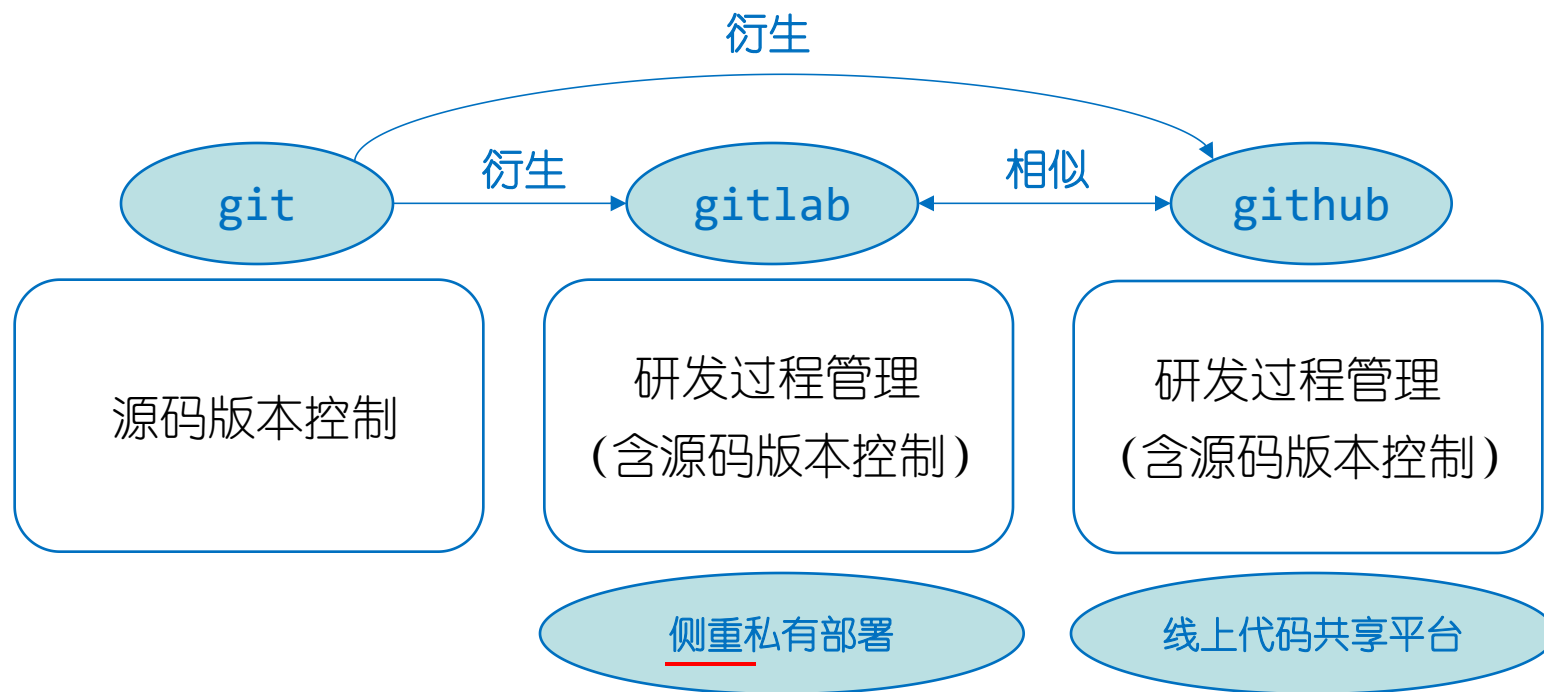
源码版本控制不等于软件工程

02. gitlab与svn的区别 (1/4)

容易懵!



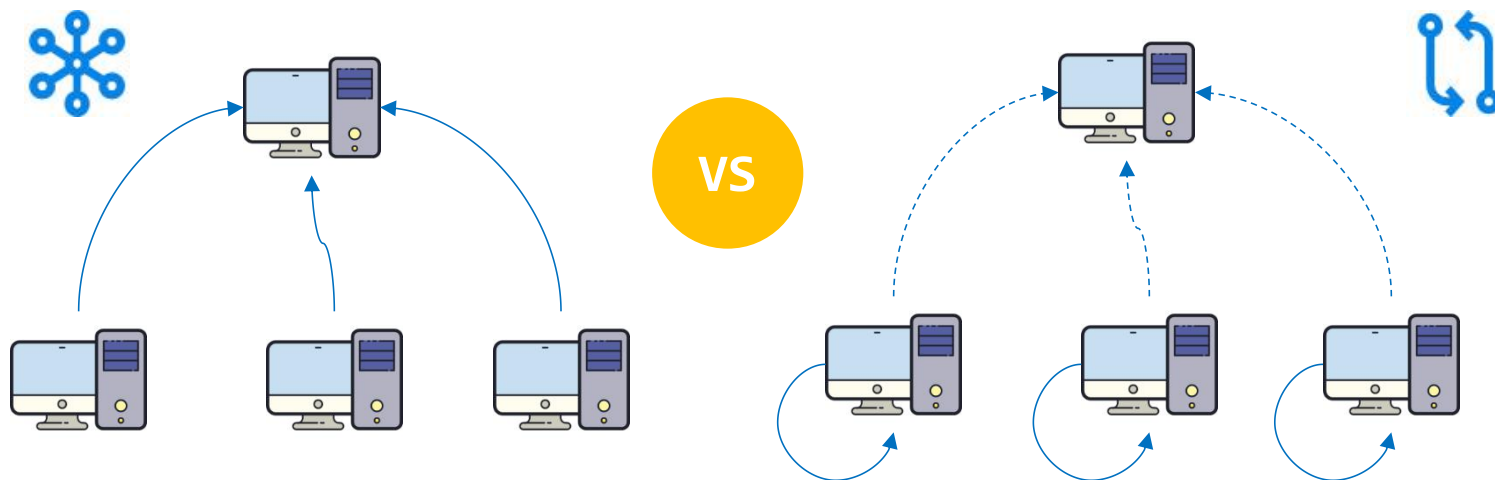
git/gitlab/github的区别?



02. gitlab与svn的区别 (2/4)



svn的集中式和gitlab的分布式



1. 仓库数据在中心
2. 大部分操作需要联网

1. 仓库数据本地也有
2. 操作可以本地进行

像区块链？



02. gitlab与svn的区别 (3/4)



集中式工作模式的烦恼？

Checkout(diff/log)操作



Commit操作

没网/Vpn断了



VS

Checkout(diff/log)操作



Commit操作

没网/Vpn断了



02. gitlab与svn的区别 (4/4)



分布式工作模式的特点？

个体相对平等

个体自由度更高

应用场景

- ✓ 适合参与人数众多的项目
- ✓ 适合异地远程开发
- ✓ 适合多团队合作
- ✓ 适合开源项目

分布式
特性

功能
丰富性



那么这个gitlab适合我们公司吗

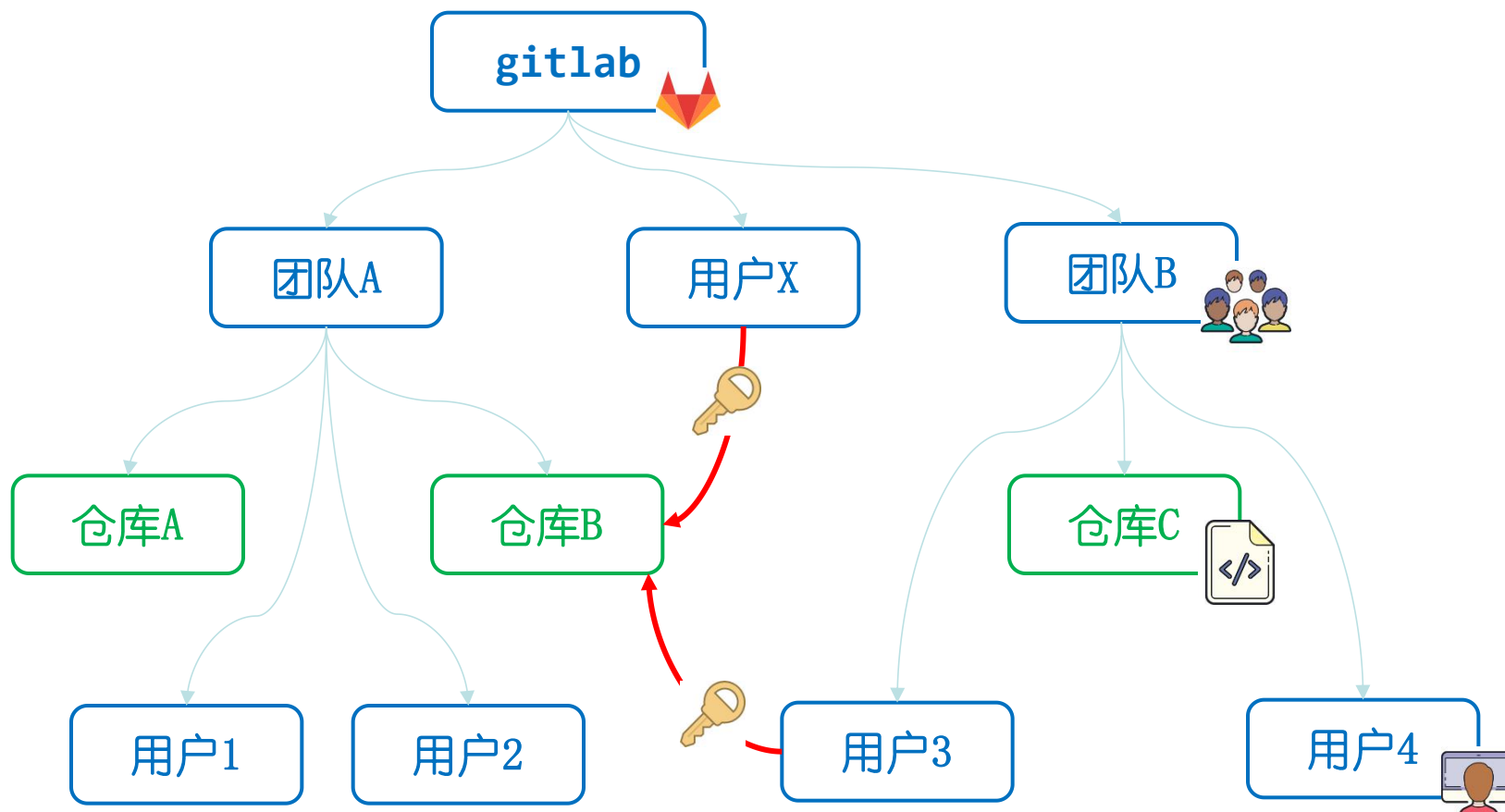
终极拷问！



03. gitlab中的团队、仓库、用户、角色 (1/4)



团队、仓库、用户的关系？




03. gitlab中的团队、仓库、用户、角色 (2/4)







03. gitlab中的团队、仓库、用户、角色 (3/4)








AI-Team > hw_ai_transcoding > Details






hw_ai_transcoding 
Project ID: 16

  Star 0  Fork 0

 574 Commits  10 Branches  0 Tags  137.6 MB Files  137.6 MB Storage

华为平台 转码+AI视频分析 项目



master-3 


hw_ai_transcoding /  


History


Find file

Web IDE


 


Clone 





Merge branch 'yangping' into 'master-3' 


wenglizhu authored 2 days ago


ddc08e7e 


 README


 Add LICENSE

 Add CHANGELOG

 Add CONTRIBUTING

 Enable Auto DevOps

 Add Kubernetes cluster

 Set up CI/CD



项目仓库主页概览信息

03. gitlab中的团队、仓库、用户、角色 (4/4)



Name	Last commit	Last update
doc	doc更新	1 year ago
egw	增加图片地址配置文件	1 month ago
fvm	上云视频每个尺寸只保留一个链接,	4 days ago
iva	添加日志跟踪天津出流慢问题	1 month ago
keygen	优化初次安装系统盘数据库不正确时fvm不能正确读取数据的问题。28181增...	10 months ago
pre-research	增加调研目录	1 year ago
scripts/huawei	增加IVA环境部署脚本	10 months ago
vbb/src/vbb	优化通道请求流的处理, 改用元子变量	2 months ago
web_src	1.tj:僵尸经常关闭	2 months ago
wtoe	将wtoe和vbb移到fvm同级目录下, 方便egww调用	1 year ago
wtoeSocket	上云请求云台控制, step解析不正确	1 month ago
.gitignore	add gitignore	1 year ago
README.md	Update README.md	11 months ago

README.md

hw_ai_transcoding

华为平台 转码+AI视频分析 项目
基于atlas 300 (型号3000) ubuntu 18.04 arm-64, graph + engine api



项目仓库代码目录/[readme.md](#)

04. gitlab中的分支 (1/2)



从新开始认识gitlab中的branch

目的都是为了
源码版本控制

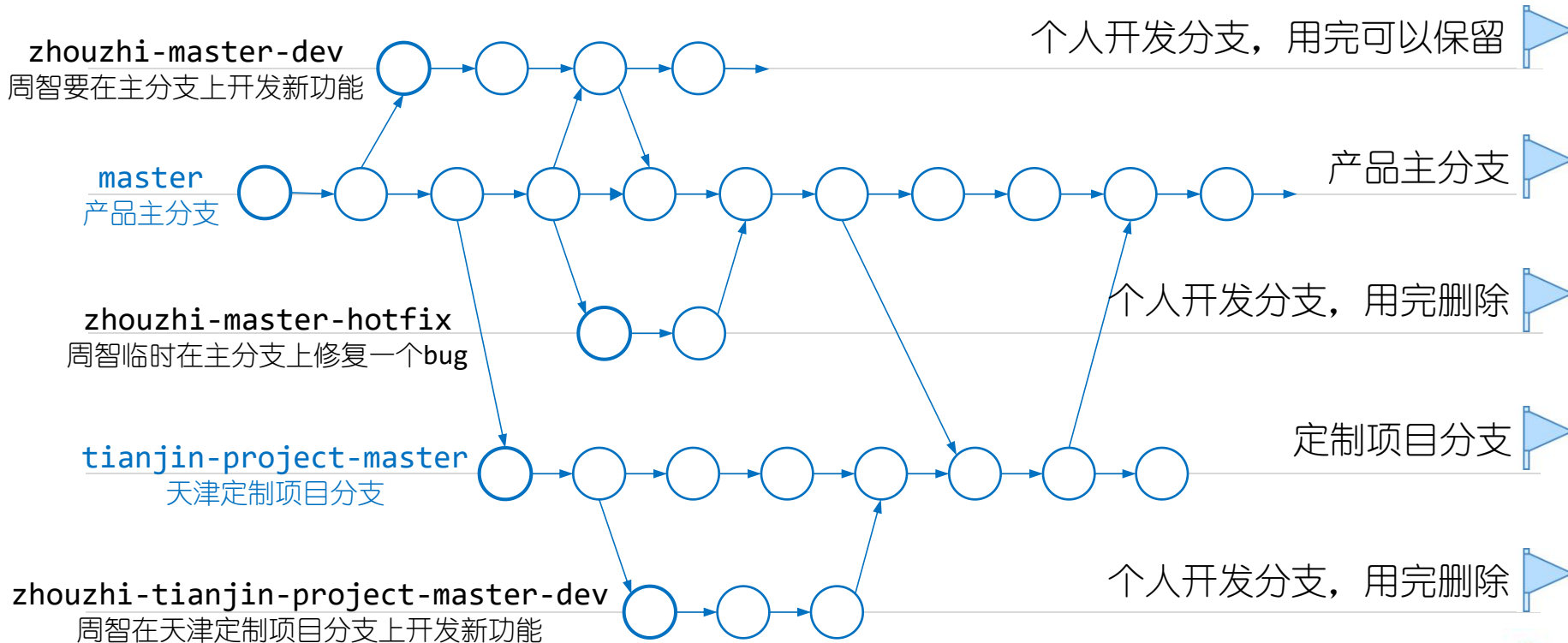
1. 底层实现机制不同（轻量级）
2. 使用场景不同（频繁使用）

04. gitlab中的分支 (2/2)



gitlab中branch的典型用法？

举个例子！



分支权限！

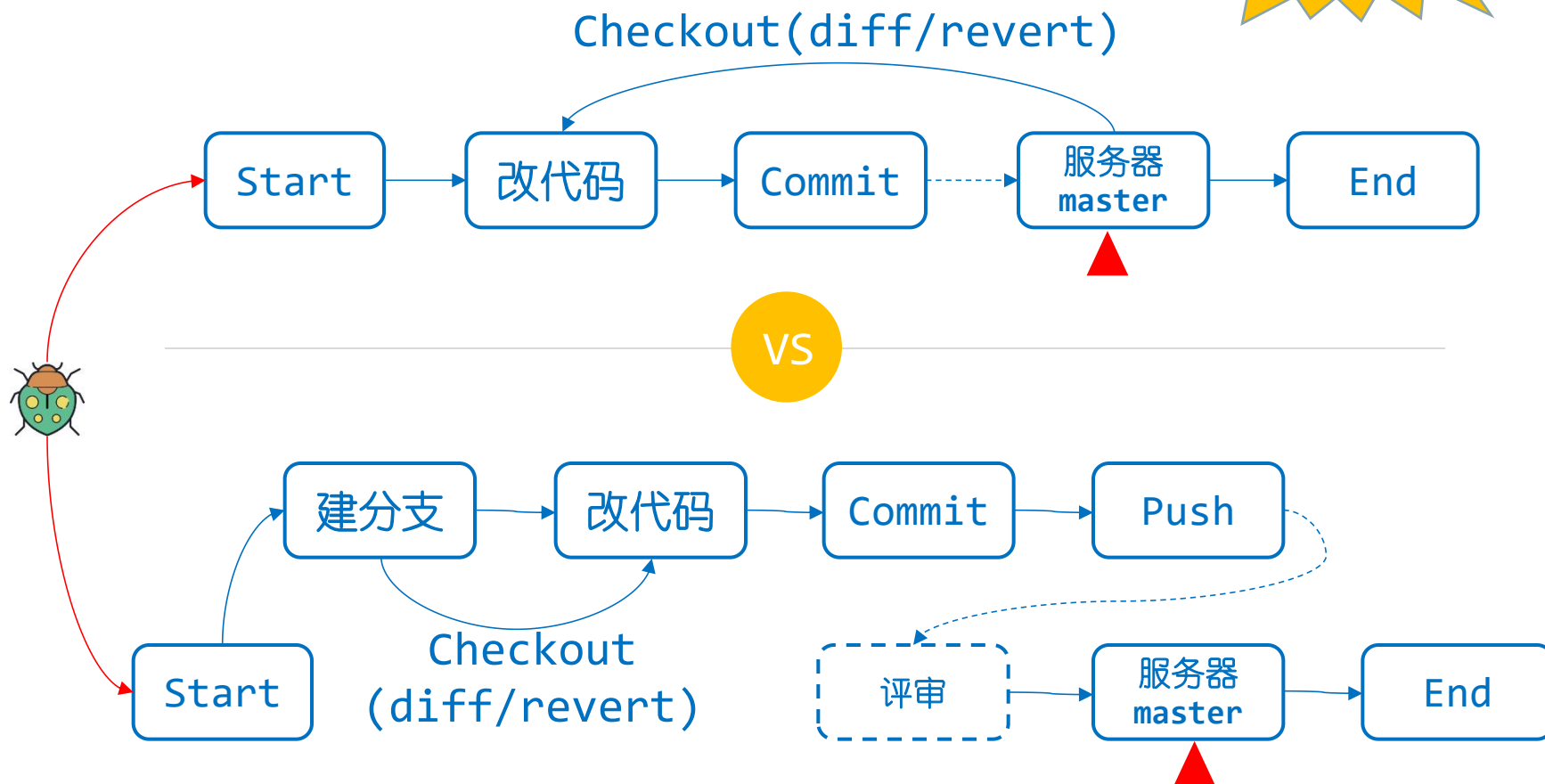


05. 如何在gitlab中提交代码 (1/3)



gitlab中修改bug如何提交代码？

再举个例子！



05. 如何在gitlab中提交代码 (2/3)



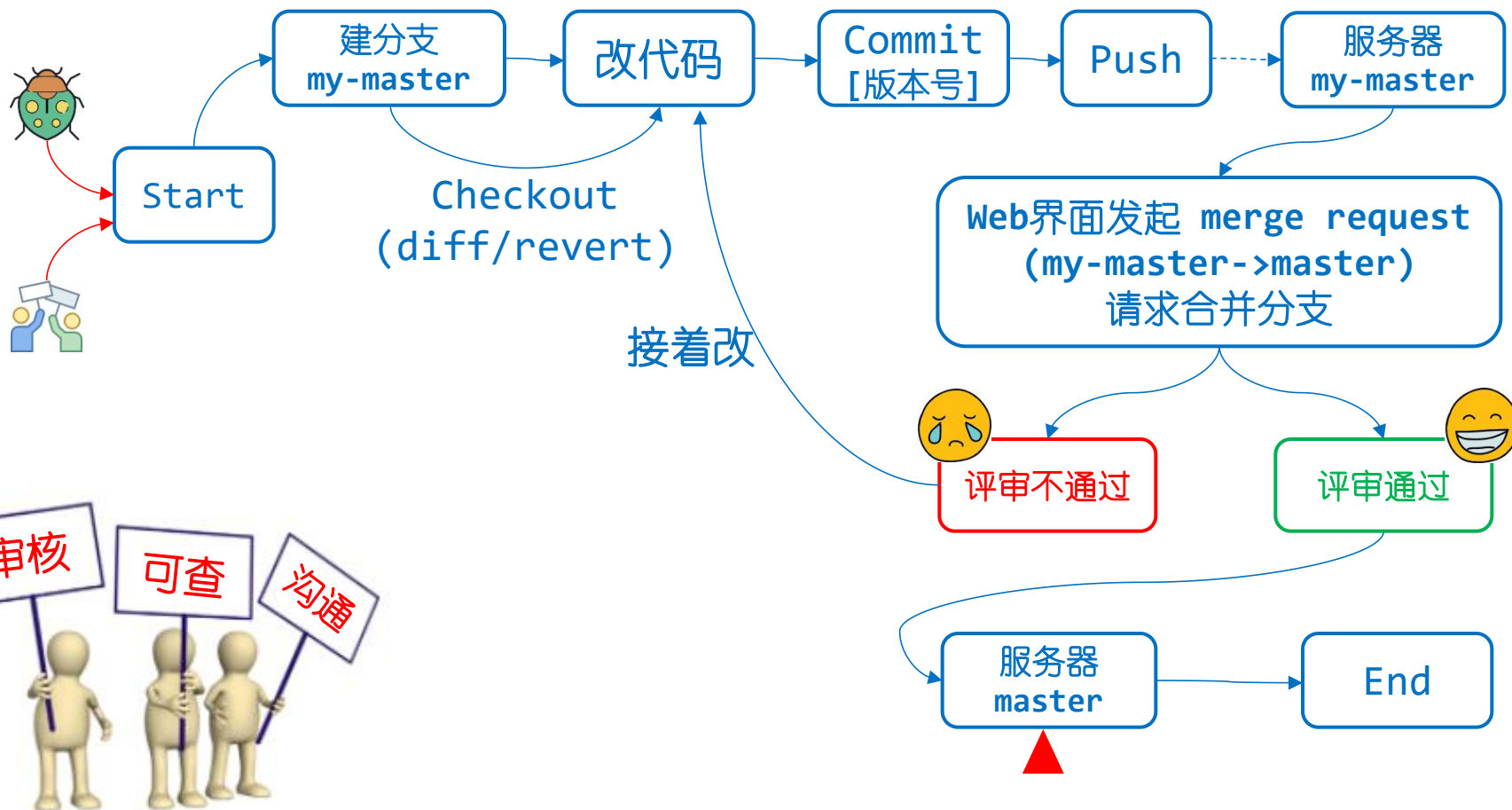
gitlab中如何做代码评审？



假装这是测试找出来的虫子



这是产品经理提的正当需求



05. 如何在gitlab中提交代码 (3/3)



gitlab中提交代码需要注意的细节

考虑评审人能读懂代码

1

单一职责原则

1. 一个Commit只负责一个工作，一个Commit包含的代码不能太杂
2. 一个MR也只负责解决一个事情，多个事情开多个MR

保障代码质量最后一关



应审尽审

1. 除特殊用户外，其余人提交代码必须走评审流程
2. 原则来讲，任何人都不允许直接往重要分支上push代码

有据可查/责任明确



日志/交流做到有据可查

1. 代码提交日志尽量精准、简洁
2. 除口头交流外，也尽量线上交流 (*做到责任有据可查*)

减少合并冲突



本地代码及时同步

1. 定期同步代码，尤其是待合并的远程目标分支
2. 分支尽量用完删除，再用时再基于最新的远程分支去创建

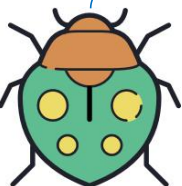
06. 如何在gitlab中提交bug (1/4)



gitlab中bug和code集中管理的优势



疯狂提bug



#131
(bug编号)



一键跳转



commit 680bf7ba
(commit版本号)

熬夜改bug



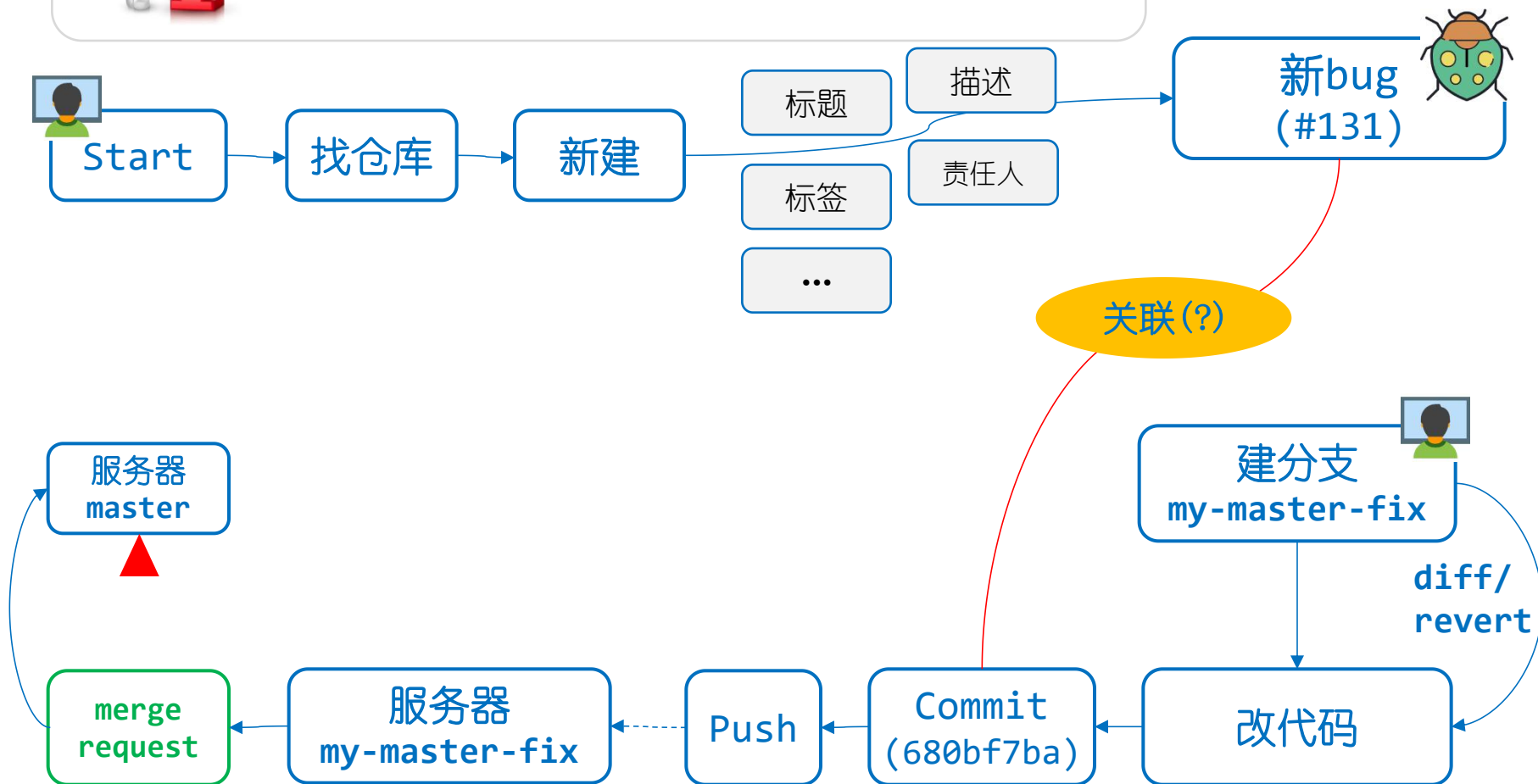
复盘方便
责任明确



06. 如何在gitlab中提交bug (2/4)



gitlab中提交/修复bug的流程

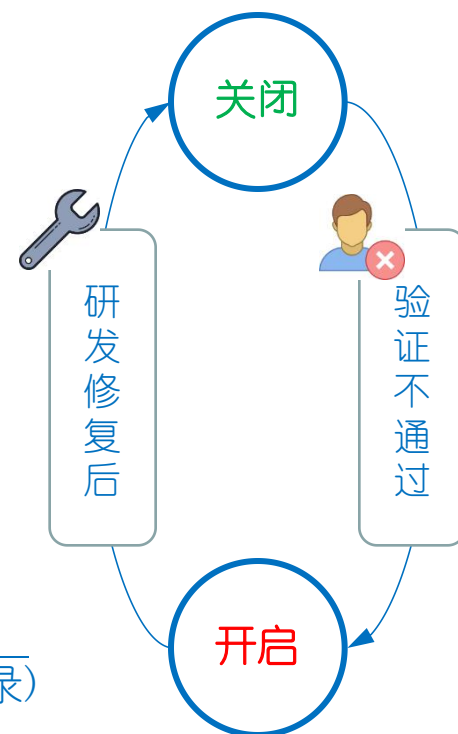
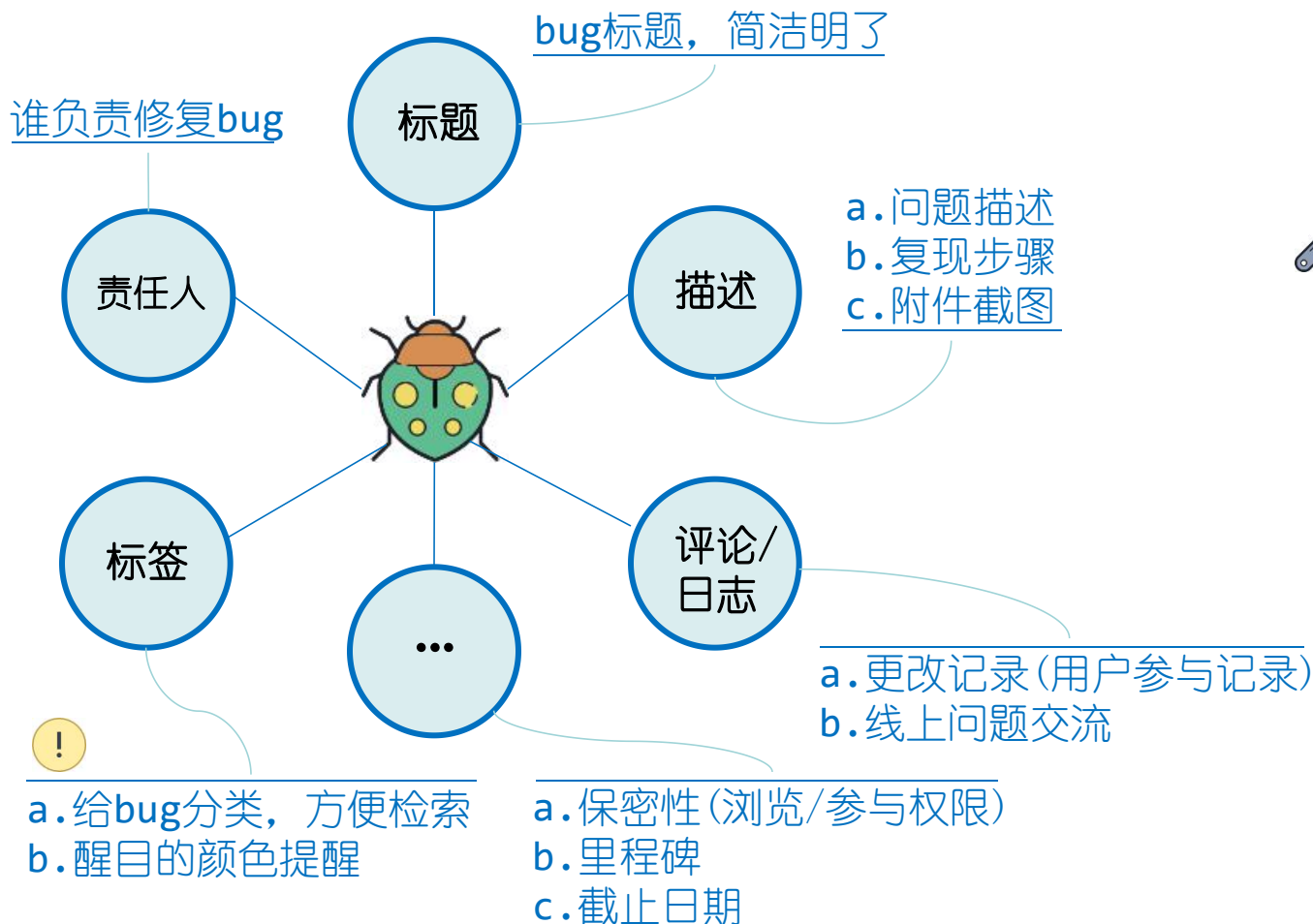


06. 如何在gitlab中提交bug (3/4)



gitlab中bug的基本信息、状态变换？

跟bugzilla比？



就俩状态？
原因后面说

06. 如何在gitlab中提交bug (4/4)



gitlab中如何Close一个bug ?

关联



找到bug
(#131)

添加
评论

插入Commit版本号
(680bf7ba)

点击
Comment & Close



先提交代码，后将commit关联到bug

VS

提交代码时，将bug关联到commit



Commit代码
(680bf7ba)

添加messgae

插入
'fix #131'

代码合并后
自动Close

关联

07.如何在gitlab中管理文档/共享资料 (1/2)



软件开发过程中的文档形式有哪些？

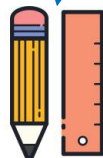


✓ 适合gitlab



过程管理类

立项相关文件
结项相关文件
项目研发计划文件
项目资源投入说明
成员绩效明细文件
...



需求设计类

需求类文档
设计类文档
技术调研性文档
系统设计原型文档
测试类文档
...



ReadMe类

项目仓库概述
代码运行环境说明
代码编译步骤说明
代码目录结构说明
跟代码相关的说明
...



协议接口类

接口交互文档
SDK/库使用文档
过程协议文档
...



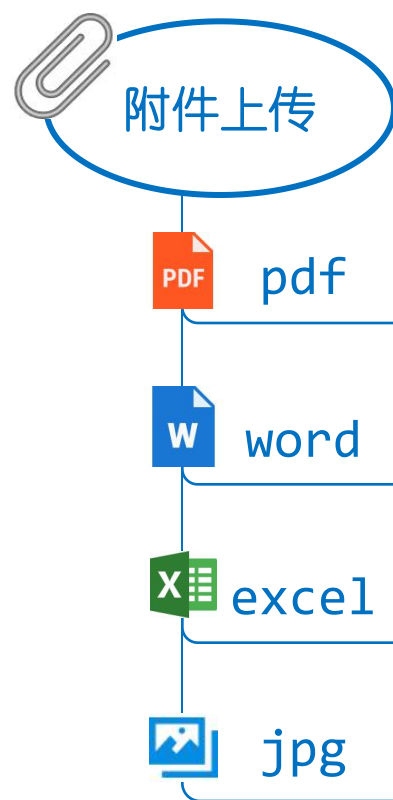
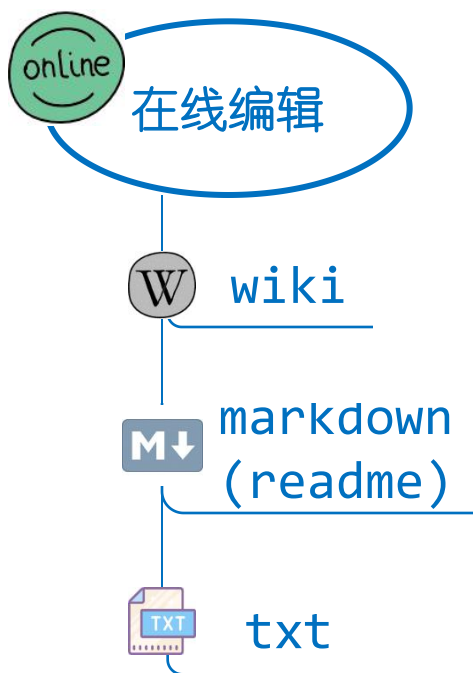
使用手册类

系统使用说明书
系统部署手册
...

07.如何在gitlab中管理文档/共享资料 (2/2)



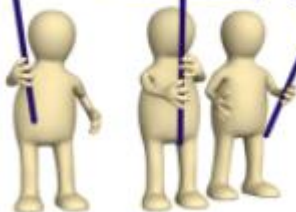
文档在gitlab中的呈现方式？



共享

版本

权限



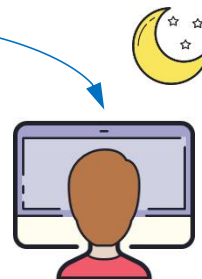
08.如何在gitlab中分配任务 (to-do list) (1/4)



先来梳理一下to-do list功能应该具备什么特点？



发起人



接收人

- 
- a. 能直观跟踪任务进度
 - b. 能根据条件快速检索、过滤
 - c. 能结合实际bug/code灵活安排任务
 - d. 能讨论交流，能插入链接资源
 - ...



- a. 能收到任务通知/过期提醒
- b. 能根据条件快速检索、过滤
- c. 能讨论交流
- d. 能与commit关联
- ...

08.如何在gitlab中分配任务 (to-do list) (2/4)



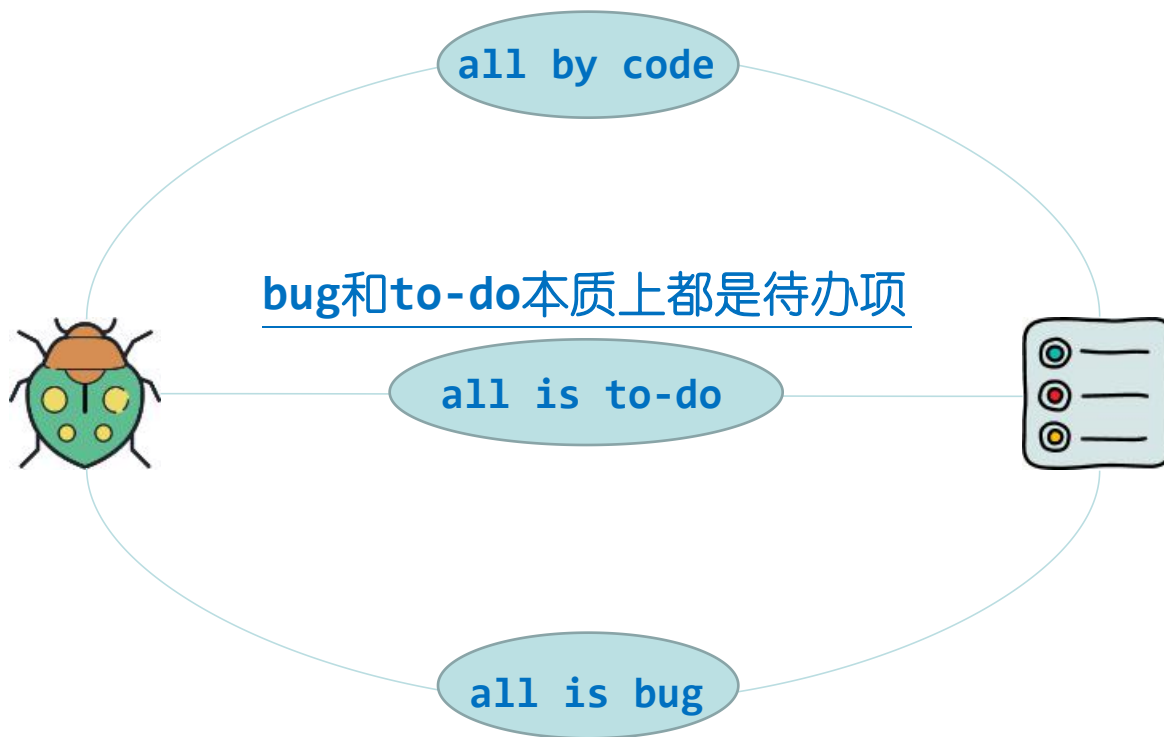
gitlab中的bug和to-do区别大吗？

不大！

差不多！

感觉一样！

基本都要通过commit代码去解决



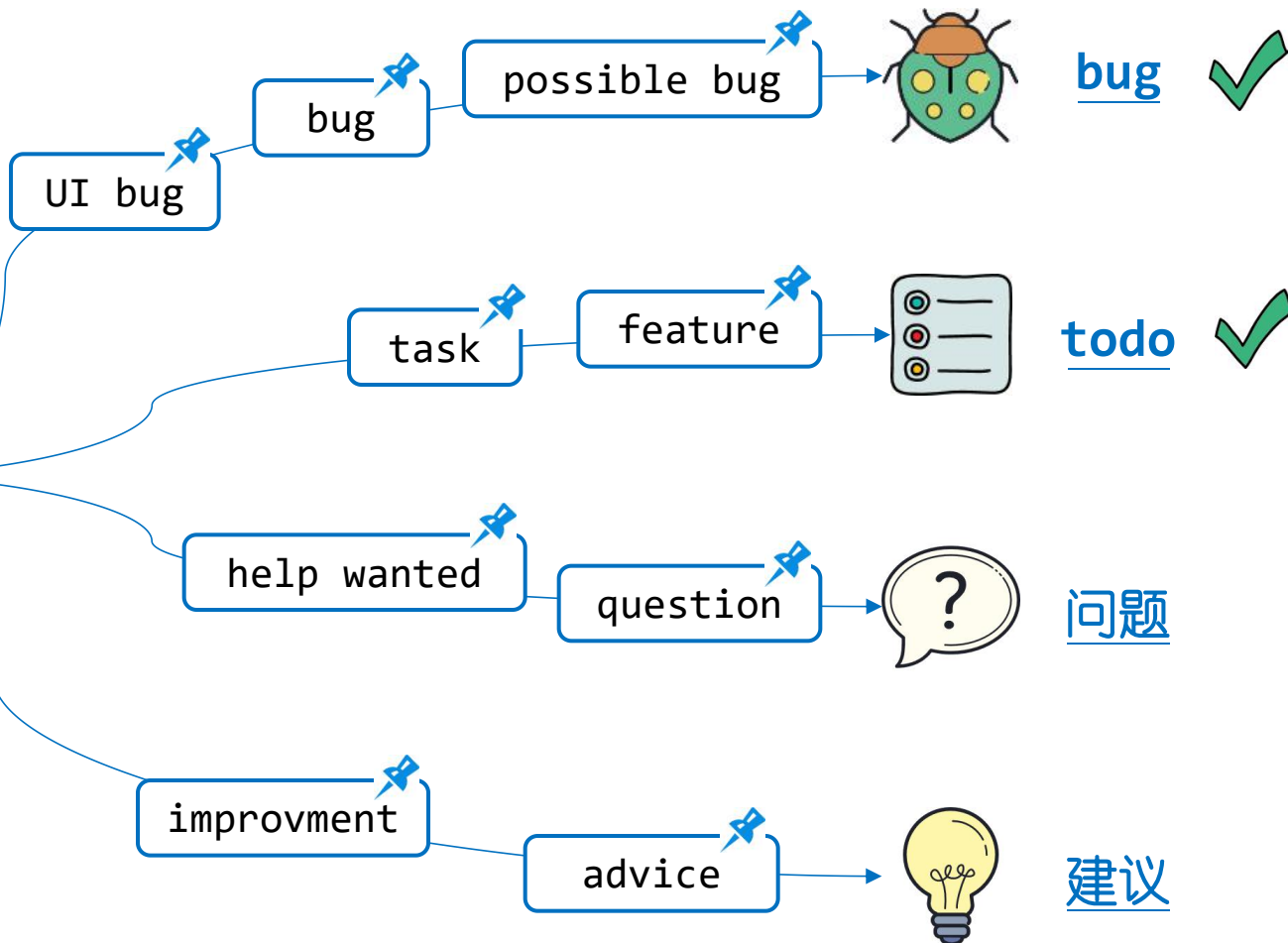
根据需求设计，feature就是‘还没开始实现’的bug

08.如何在gitlab中分配任务 (to-do list) (3/4)



gitlab中如何区分bug和to-do或者其他？

标签



08.如何在gitlab中分配任务 (to-do list) (4/4)



gitlab中如何定义标签 (Label) ?

举个例子!

bug

feature

售后反馈

iteration1

doing

task

模型问题

iteration2

suggestion

刻不容缓

验证通过

暂不处理

非常重要

优先级低

验证不通过

偶现

09.如何在gitlab中发布版本？ (1/3)



gitlab中标签(label)和标记(tag)的区别？



 label

直观/分类/筛选

自定义名称创建

 tag

归档/快照/索引/检索

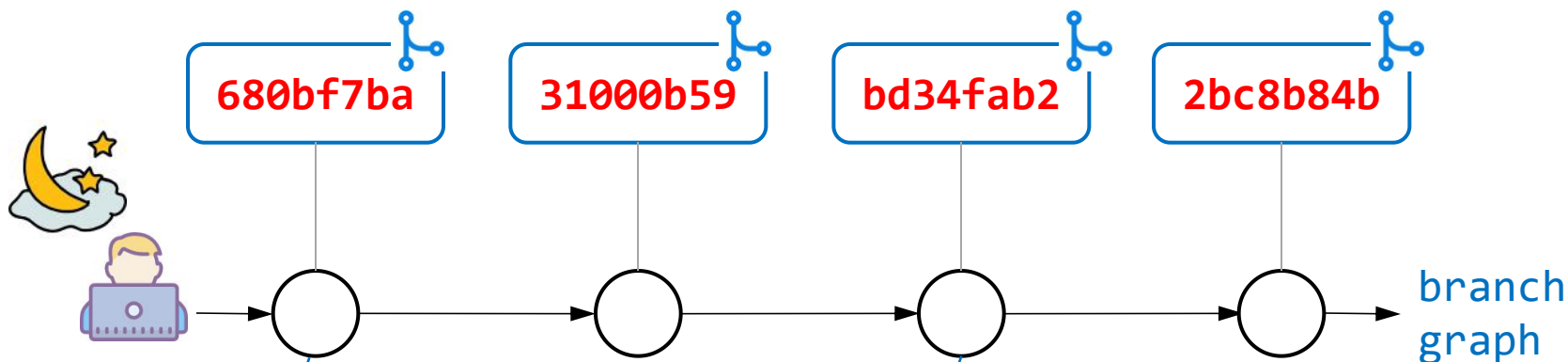
基于commit创建

09.如何在gitlab中发布版本？ (2/3)



gitlab中tag的作用是啥？

归档



tag message
balabala...

tag message
balabala...



won't lost with map

09.如何在gitlab中发布版本？ (3/3)



gitlab中发布版本的过程是？

先创建tag，基于tag创建release



v1.0-pre



- a. 发布说明
- b. 源码下载
- c. 附件下载
- d. 版本对应分支
- e. 版本对应commit号
- f. 版本对应tag
- g. 版本创建人/时间

VS

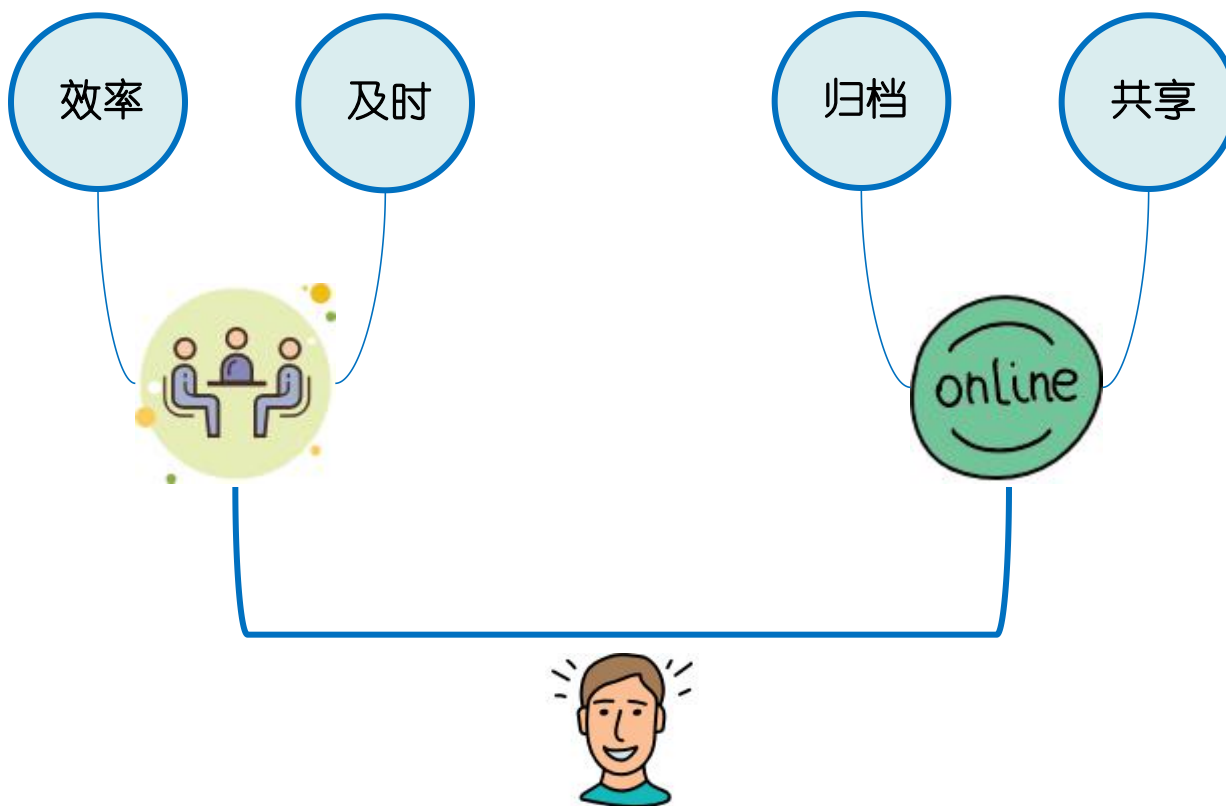


直接release，顺带生成对应的tag

10.如何在gitlab中互动交流？ (1/2)



线上交流的必要性？



都有必要



10.如何在gitlab中互动交流? (2/2)



gitlab中互动交流的场合有哪些?

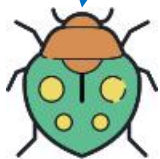
举最后一个例子!



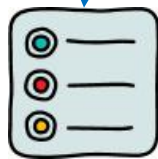
+



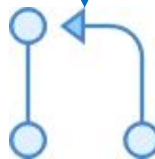
评论+@指定人



bug



to-do



merge
request

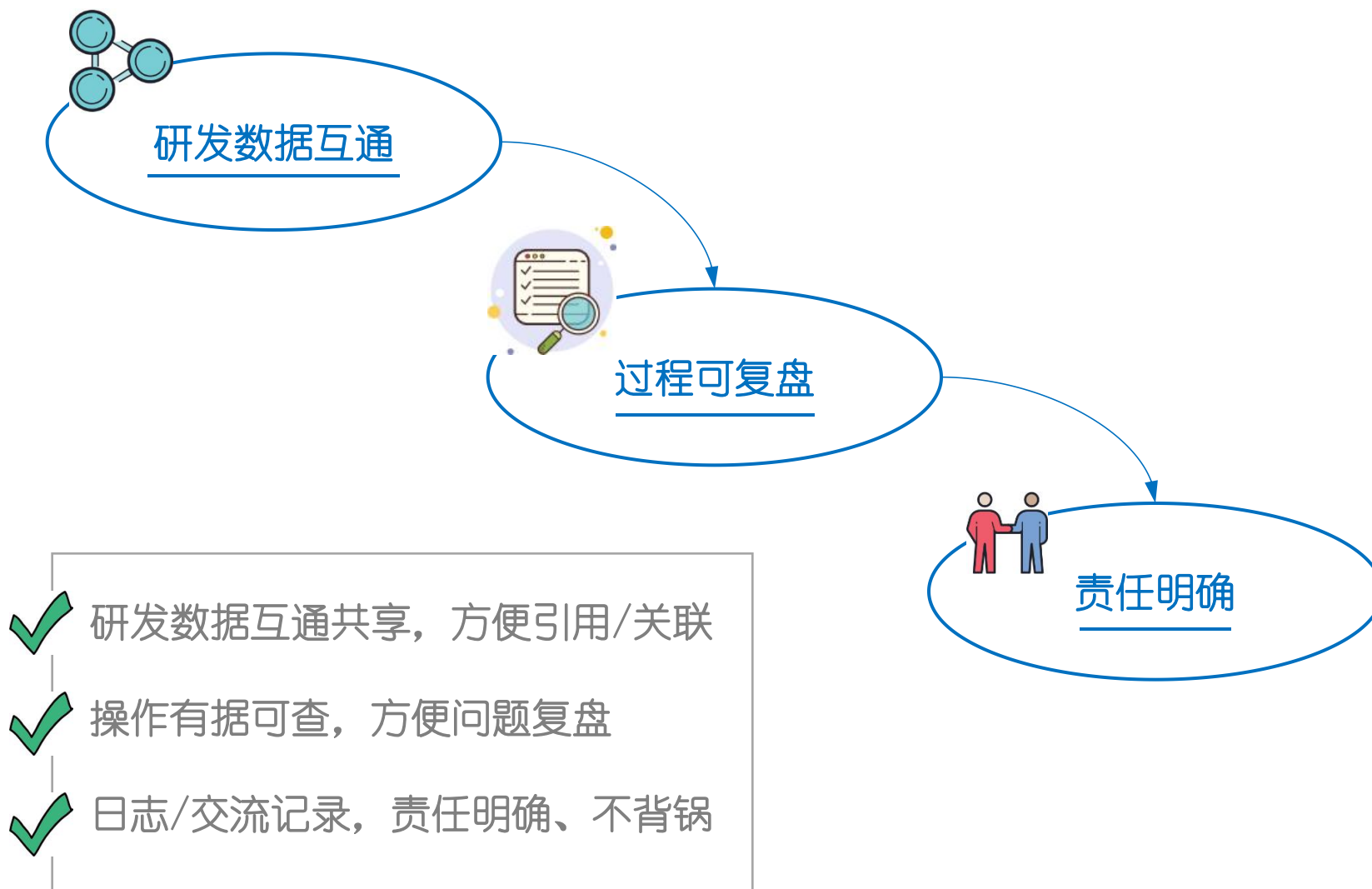


commit

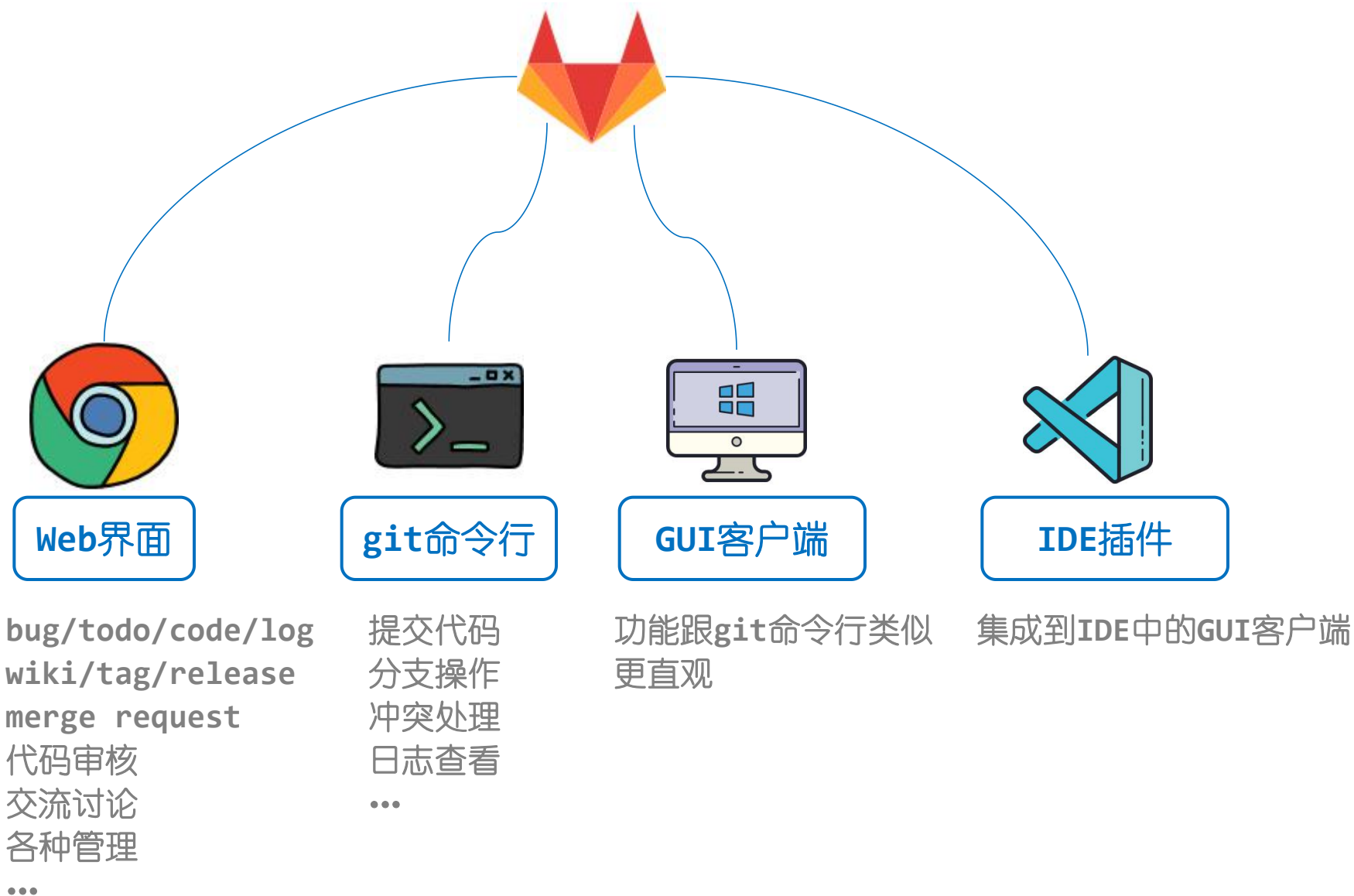


wiki

11.gitlab使用价值 (1/1)



12.gitlab操作终端 (1/1)



谢谢