

## 关于代码托管平台的调查

**Question 1.** 作为一个计算机大类的学生，你通常使用的代码托管平台是什么？

答：我常用的代码托管平台是微软公司的 github.

**Question 2.** 比较代码托管平台，列出优缺点。

答：为了得出较好的平台比较效果，我们选择市面上主流的代码托管平台 github（最主流且本人一直使用）、gitee（较为主流但本人没用过）、gitlab（不是很主流但本人曾经使用过）进行比较。

- 关于网络环境
  - **Github** 网络环境不是很稳定，国内用户经常需要利用科学上网的方式才能链接登陆或上传、拉取仓库。
  - **Gitlab** 网络环境在可以使用是较好，但有时服务器会宕机维护，这段时间内通过任何方法都无法登陆或上传拉取自己的仓库。
  - **Gitee** 目前国内由深圳市奥思网络科技有限公司版权所有提供服务，网络问题相对较少。
- 关于联合生态
  - **Github** 与许多集成开发环境有着紧密联系，比如 vs code、overleaf 等众多集成终端，与此同时世界上许多的主流程程序员社区也都支持用 github 登陆，其生态环境是另外两个代码托管平台不能匹敌的。
  - **Gitlab** GitLab 被 IBM, Sony, JülichResearchCenter, NASA, Alibaba, Invincea, O'ReillyMedia, Leibniz-Rechenzentrum(LRZ), CERN, SpaceX 等组织使用。但面向大众的合作的终端或社区相对较少，生态环境相对较为闭塞。
  - **Gitee** 与国内一些主流产品进行合作，如提供微信小程序等，同时也提供主流浏览器的插件（但优化一般，更新不及时）以及一些集成编译器的插件，总体来说生态环境在国内市场较好，但放眼世界相对 github。
- 关于功能

	GitHub	Gitee	Gitlab
代码托管，支持 Git/SVN	仅支持 git	均支持	仅支持 git
开源项目、代码片段	支持	支持	支持
Issue	支持	支持	支持
Wiki	支持	支持	支持
Fork + Pull Request	支持	支持	支持
组织	支持	支持	支持
在线 IDE	不支持	支持	支持

TABLE 1. 功能表

	GitHub	Gitee	Gitlab
仓库自动备份	不支持	支持	不支持
禁止 Git 强推	不支持	支持	不支持
支持仓库访问 IP 限制	不支持	支持	不支持
企业级研发协作	收费	五人免费	不支持
敏捷开发管理	不支持	支持	不支持
任务看板（可灵活定义）	不支持	支持	不支持
支持多级任务、关联任务	不支持	支持	不支持
自动代码质量分析	不支持	支持	不支持
快捷生成工作周报	不支持	支持	不支持
代码克隆检测	不支持	支持	不支持
自动生成 JavaDoc/PHPDoc	不支持	支持	不支持
多语言 README 自动渲染	不支持	支持	不支持
支持微信/钉钉通知	不支持	支持	不支持
完善的管理界面	不支持	不支持	支持
完善的权限控制	局限	局限	完善
订阅	支持	支持	支持
文本渲染	支持	支持	支持
讨论组	支持	支持	不支持
协作图谱	支持	不支持	不支持
Gist	支持	不支持	不支持

TABLE 2. 功能表 (续)

可以看出，gitee 推出了一些新功能。这些功能大多数针对公司的项目开发环境，并且与中国市场上的主流软件和工作环境紧密结合，可以在一定程度上提高工作效率。但这三个代码托管平台在基本功能上并无本质区别，其提供的服务基本可以保证日常使用需求。

- 关于语言
  - Github 仅有英文版本，对中国用户不够友好
  - Gitlab 支持中文
  - Gitee 支持中文
- 使用场景
  - Github 世界上最大的开源代码托管平台，汇聚了世界上大多数的开源顶尖项目，可以帮助程序员之间互相交流和学习，也可以参与各式各样的开源项目以及托管自己的代码仓库。
  - Gitlab 基于 Git 实现的在线代码仓库软件，可以用 GitLab 搭建一个类似于 GitHub 一样的仓库，但是 GitLab 有完善的管理界面和权限控制，具有较强的可塑性和自定义性，一般用于在企业、学校等内部网络搭建 Git 私服。
  - Gitee 全中文环境，而且大部分用户都是国人，不会出现不稳定情况，但优秀的库相对于 GitHub 也少。提供免费的 Git 仓库，还集成了代码质量检测、项目演示等功能。对于团队协

作开发，Gitee 还提供了项目管理、代码托管、文档管理的服务，5 人以下小团队免费等许多面向中国特色的功能，适合中国公司团队做开发项目使用。

### Question 3. 开发一个代码托管平台应如何改进？

答：开发一个代码托管平台，我认为第一步要做的事情是去调研了解用户需求。在了解我身边同学以及我个人的需求后，我得出以下可以改进的地方。

首先网站的稳定性是一个极为重要的问题，客户肯定不希望链接时多次链接失败甚至需要挂梯子才可以使用，更不希望网站常常宕机一两天甚至更长时间。所以好的网络状态是必不可少的，时时刻刻保证客户可以连接到平台进行仓库的拉取和推送（这一点 gitee 做的较好，另外两个调研的网站需要对中国市场进行优化改进）。

第二点便是语言的问题。想要开发一款占据主流市场的软件，我觉得支持多语言是一项非常加分的环节（尤其是当有众多竞争对手的时候）。首先如果作为一款全球性的软件，英语版本一定是必不可少的，其次作为中国的开发者应该加入汉化版本，对于中国的消费者而言，即使很多人英语水平尚可，但还是更愿意使用中文版的软件，如果两个功能类似的软件在中国抢占市场份额，具有中文版本的一定会相较于只有英文版本的更具竞争优势。

其次便是，作为一个依托于 git 的代码托管平台，其实对初学者有一定的使用门槛，很多人并不会在终端中输入命令去操纵仓库，且 git 命令相对较多且复杂，对初学者不算友好，更不用说计算机行业以外的人。所以我觉得，可以做一些图形化界面，让更多的 0 基础用户更好的入门，让用户有一种可以通过点击进行操作的舒适感；并可以在适当的时候，推出一些教程或指令表，让用户更好的掌握平台的使用，带给用户更好的体验。

然后平台的生态环境也是一个极为重要的方便，如果一个平台可以和用户平常所使用的各种软件相关联，比如利用该平台的账号登陆或者一键推送仓库等等，那必定会大大提升用户黏性，增加消费者的使用。

最后，根据我实际的体验来说，我个人认为使用代码托管平台最为复杂繁琐、且对用户来说最容易放弃使用产品的步骤，便是将本地的私钥绑定到平台的环节。这一环节不但较为复杂，且网上的教程说法千奇百怪，总会出现各种奇奇怪怪的问题导致绑定失败，许多用户在这一步便放弃了产品的使用。我觉得，这一部分可以由代码托管平台开发相应的脚本文件，实现这一功能的一键点击完成，帮助用户跳过繁琐的配置环节，直接跳到使用环节。如果这样做不能保证安全性，便可以提供详细的官方教程，让用户的配置环节尽可能的变得简单易操作。