

前端项目02

上次回顾

项目，对自己的要求，做成什么样

1. 数据量想大
2. 网络，电脑的性能想的差一些
3. 用户想的傻一些

前端架构师的身份，如何看待项目

从工程师=》前端管理的过程，别的知识体系

1. 项目人员的规划
2. 需求问题（项目管理）
3. 项目的可维护性（未来）
 1. 代码的实现是开心的现在
4. 小项目的leader之后，如何进一步

1. 每一条路，对知识和能力的要求i仕不一样的

就从这个项目开始展开

1. 登录注册
2. 用户中心
3. 文章管理
4. 关注点赞（多对多的关系设计）
5. 评论

大部分的项目，都可以这个几个需求展开扩展

做项目之前

做项目的代码

做项目之后

1. 制定规范(规矩)
2. 开源（开发效率，影响力）
3. code review
4. 工具（开发工具）

代码之后

1. 人效（每个维度综合考虑）

1. 代码量（吐槽），可以用gitlab的api，来统计
2. 每个人都要触碰到自己的极限
 1. 比如轮岗（每个人尽可能能维护两个模块+）
 2. 比如内部系统奖励机制，发挥内部主管能动性
 1. 日志监控，主动领任务
3. 。。。。

2. 基础建设

1. 项目长期维护的必备设置，美好的未来
2. 命令行工具 cli
3. 组件（代码规范，单元测试）
4. 埋点（数据采集）
 1. 性能监控 浏览器或者node，统计的性能参数，发给我们服务器
 1. perfomance
 2. lighthouse (宏观的)
 2. 错误监控
 1. sentry
 2. fundebug也挺有意思的，复现功能
 3. 原理也不难，window.onerror, try catch

主动上报

4. promise报错

3. 用户行为日志监控

1. ga

2. 百度统计

3. growingio

5. 构建发布

1. gitlab

2. github(action)

3. 自动化发布，钉钉推送消息

3. 计划（向上管理）

1. 团队的资源

2. 团队的成长

3. 目的是通过某个项目，优化自己的知识体系

同一个项目，同一个需求，不同的前端，做出来就是不一样

对团队和自己的要求

技术级别，有点像学历，

1. P5

1. 独立完成 经验丰富，给我需求，奥利给

2. P6

1. 要求担当
2. 前沿研究，踩坑
3. P7

1. 领域专家
2. 体系化知识

技术路线，P几只是其中一条路，winter老师，玉伯，张云龙，尤大，张xinxu。。。有很多

1. 玉伯，体验科技 前端技术和管理深挖
2. winter P8之后，再搞教育
3. 张云龙 fis的作者 走的是全栈CTO的路线
4. 鑫旭，css 死磕
5. 尤大 独立开发者

前端的路线又很多，每条路要求的知识都不一样

1. 大公司前端架构师
 1. 技术原理，vue nuxt, egg等等
 2. 计算机基础
 1. 算法，网络，操作系统，数据库....
 2. 从算法开始，算法训练营欢迎大家

3. 网络协议跟上

3. 前端的前沿技术

4. 有没有兴趣读完vue源码

2. 大公司技术管理

1. 多关注人

2. 管理，听起来高大上，干起来很辛苦

3. 人员的招聘规划，进度的管理，资源的协调。。。懂技术，懂人

4. 同时，你的技术还不能落下，否则地下的人对你没有技术尊重，队伍不好带啊

5. 工作的内容会发生变化

1. 把我们的小开社区对外发布维护，你需要几个人

2. 这几个人，怎么招 跟hr聊

3. 招到之后，怎么建设团队

1. 提升积极性

2. 提升成员技术能力

4. 流程有效执行

1. 绩效 KPI，还是okr

5. 绩效差的哥们，和好的哥们，怎么去跟他沟通

6. 怎么留人和开人。。。。有很多类似这种任务

3. 小公司技术总监

1. 稍微弱化前端，强化后端或者全栈
2. 强化产品能力
3. 强化管理能力
4. 接触的更广，更泛泛，更需要综合实力

4. 独立开发者，独立做项目

1. 公司就你一个人，做一个产品来养活自己
2. 全栈！！
3. 营销
4. 产品
5. 用户增长....
6. 什么都干
7. 极可能多用开源和成型产品

5. 等等

摸鱼：上班偷懒

钓鱼：真的钓鱼

做项目之后的反思

加班

很多公司都有加班，长期996的结果，并不能提升你的效率。只会让你爱上摸鱼

1. 学习的重要性大于加班
2. 短期996可以接受，常年996 还是算了

我不是科班出身的，我有半年时间，学习时间只学算法
以后学不动了会不会被淘汰啊

1. 只学皮毛，一定会被淘汰
2. 学核心的知识，核心知识4，永不过时
 1. 算法
 2. 编译原理
 1. ast codegen核心理念
 3. 武林高手要学习内力，而不是十八般兵器
 4. 任何一个知识点，都有一本经典的书，死磕下来，你就超过了大部分同行
 5. 任何一个之前的技能，都不可能简单的获得

前端的路很宽广的，是技术圈最懂用户体验的，最有可能独立做出项目的

年龄限制，只针对增删改查程序员

读书

1. 无论是你想赚钱，提升编程，职位晋升，都需要看书
 1. 就算你想找对象，我也推荐一本书《魔鬼约会学》，如何聊天
 2. 技术层面的书，豆瓣8分以上，都值得看
 3. 我早上看书，无论几点睡觉，6点~7点起床，打两把游戏，看一个小时书
 4. 开课吧的装修，就是图书馆，公司也可以看书
2. 读书其实是有快感的
 1. 薪资的提升《算法》
 2. 逼格比较高，适合装逼

项目的思考，职业生涯的思考

