



快速打造移动应用

@许式伟

- 七牛云存储 CEO
- Go 语言大中华区首席布道师（自封）
 - 《Go 语言编程》作者（之一）
 - 《Programming in Go》译者（之一）
- 盛大祥云计划发起人
- 金山实验室（云存储）创始人
- 前金山技术总监
- WPS Office 2005 首席架构师

- 故事假设
 - 我是一个从业的技术人员
 - 移动互联网时代
 - 我想投身创业大潮，实现梦想
- 快速的含义是什么？
 - 快速发布产品
 - 快速产品迭代
 - 快速形成口碑

我面临哪些问题？

- 第一：团队
 - 要哪些角色？从哪里来？
- 第二：客户
 - 谁是用户？如何让他们知道我们？
- 第三：产品
 - 要做什么产品？客户核心诉求是什么？
- 第四：融资
 - 要多少钱？从哪里来？
- 第五：项目
 - 什么样的开发计划？多久发布？

- 创业就是不断消除不确定性的过程
- 善用第三方资源
 - 做好第三方资源的成熟度评估
 - 把精力放在刀刃上，能够不自己做的事情不做
- 小步快跑，及早发布，快速迭代

- 比业务更重要的，是人
- 不需要等所有人都齐全
 - 达到第一阶段目标所需的人准备好就行
- 第一批人非常关键
 - 要找熟人
 - 尽量共事过，彼此知根知底
 - Why ?
 - 消除不确定性，降低风险
 - 有些事情，你错不起
- 及时补充人才
 - 在特定的阶段需要特定的人才
 - 及时补充团队的短板

- 目标群体是谁？
 - 选择自己非常熟悉的领域
 - 选择自己非常熟悉的群体
 - 你非常了解他们的习性，了解他们的述求
 - 如果你在他們心中是解決他們問題的專家就更好
 - 如果還不是，那麼從現在開始，準備好怎麼逐步去成為專家
- 目标群体越窄越好
 - 越窄就意味着越垂直，意味着面临巨头的直面竞争的可能性越小
 - 但不必过分惧怕与巨头产生竞争关系

- 产品不需要一上来就很完善
 - 满足客户的核心诉求就可以发布
 - 尽早发布比什么都重要
- 产品不是规划出来的，是运营出来的
 - 获得用户反馈
 - 快速迭代
- 什么是互联网精神？
 - 互联网产品的本质
 - 发布 => 获得用户反馈 => 改进并发布 => 用户反馈 => ...
 - 互联网营销的本质
 - 口碑营销

- 向谁要钱？
 - 财务投资 or 战略投资
 - 我个人更倾向于尽量找财务投资
- 什么时机要钱？
 - 总体原则是：越早越好
 - 但要在找好团队后拿
 - 可以写个ppt拿，可以写个产品原型拿，也可以做一定用户拿
- 要多少？
 - 建议能够让团队无营收下生存 1.5 年 - 2 年为宜
 - 考虑到团队有试错风险，第一次建议是 2 年
- 出让多少股份？
 - 不要超过 20%

- 支持的终端类型
 - iOS / Android / Web / PC(Windows/Mac/Linux)
 - 可以集中精力做好一个端
 - 根据目标用户群体，锁定一个端
 - 除非产品形态需要，不要一上来求大而全
- 第一次发布
 - 合理周期：3 ~ 6个月
- 产品迭代
 - 合理周期：1周 ~ 3个月

- 时间成本才是最大的成本
 - 如果业务方向还在探索
 - 用最快的速度最低的成本试错
 - 如果业务方向已经明晰
 - 用最快的速度把对手甩得更远
- 人力成本是第二大成本
 - 能够不自己干的事情不要干
 - 哪怕做这事也许真不难
 - 用最低的成本做事情（不要忽略隐性成本）
 - 提升团队的工作效率

- 邮件/邮件列表
 - Gmail、Google Groups (免费)
- 文档
 - Wikidot (免费* , 面向内部)
 - 七牛云存储 (面向客户, 改天写个 “七牛如何基于七牛打造七牛” ^_^)
- DNS
 - DNSPod (付费)
- 集群监控
 - 监控宝 (付费) + 自有监控平台
- 代码管理
 - Github (付费)
- 日构建系统
 - Jenkins (有机器和系统管理成本 *)
 - Drone.io / Travis-CI (付费)
- 缺陷管理
 - Redmine (开源软件, 有机器和系统管理成本 *)

优选云服务，次选
开源软件方案，极
端情况自己做点小
工具

- 服务外包取代技术外包
 - 技术外包不对结果负责
 - 使用开源软件本质上也是技术外包，只是是众包模式。
 - 运维难度越来越大
 - 就算有相应的开源软件，看管软件的运行过程，以保证服务的健康运行，也已经是巨大的负担
 - 竞争越来越激烈
 - 巨头横行
 - 大量的同质化产品
 - 如何让自己跑得比别人快？创业者需要善假于物。

- 有时候，慢才是快
- 研发过程中的“慢才是快”
- 口碑营销中的“慢才是快”
- 公司运营中的“慢才是快”

- 过渡设计 vs. 架构设计要保留对未来需求变化的预见性
 - 有人会问：话都是你说的，这两者不是矛盾的么？
- 需求深度分析的必要性
 - 不要慌慌忙忙去赶工，理清脉络会有助于你后面的快速发展

- 提升团队的工作效率
 - 做正确的事情
 - 提升单位工作时间的产量
- 磨刀不误砍柴工
 - 认真梳理团队做的事，和做事的方法
 - 避免瞎忙
- 不要让大家习惯性加班
 - 加班必须有明确的述求，并且给予必要的补偿

- 模块开发的“慢才是快”
 - 七牛模块开发方式
 - 需求分析 => 架构设计讨论 => 开发 => 单元测试 => PeerReview => FormalReview => 预发布环境 => 灰度发布 => 上线
 - 很慢么
 - 为何要这么慢？
- 我之于“代码的生命周期”的观点
 - 代码生命周期可大致分：研发期、维护期
 - 研发期多投入1天，维护期可能节省1个月甚至更长

- 口碑建立是个慢热的过程
 - 量变到一个级别后才能形成质变
 - 然后才是口碑的爆发期
- 口碑建立前期需要慢火细熬
 - 不要追求用户量
 - 要追求质，确保每一个吸引过来的都是真正的客户
 - 然后，让他们爽
 - 努力超越他们的预期

- 学会阶段性反思
 - 不能光顾着跑
 - 方向跑错了，跑得越远死得越快
- 从获得的各种反馈信息中，思考
 - 我们的方向有没有问题
 - 是要微调、还是要彻底掉头？
 - 战略方针有没有问题

欢迎微博交流：

@许式伟

@七牛云存储