链滴

2016 一个想做架构师的码农

作者：88250
原文链接：https://ld246.com/article/1483027307708
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

工作

今年在房星主要负责了几个项目的技术设计和实现：

- Protyle：敏感数据存取
- CC：呼叫中心
- Shotgun：房源群发
- Spider：房源爬取
- Auto：项目自动部署平台
- StarChart：户型图测绘，以链家作为对标
- SymX：企业内网论坛
- Petal：配置中心，才开的新坑

技术架构方面服务化的 Java 技术栈基本铺好路了，接力棒开始交给了其他伙伴。

房源群发和爬取这个看上去有内容可做，不过泼冷水的是 58 已经上线了集客家，也是做房客源推荐群发的，然而就在前不久我已经向老板请缨明年去专职做这一块了，怎么办 oy

三十而立

hi，你们想过当你40岁以后你能做什么吗?

能力与价值

今年开始奔四了，但我才刚意识到一个现实：我的能力不足以支撑我的生活。据说人最痛苦的事情之就是实力不足以支撑其野心。

在工作上，公司给了我一个较自由的定位，我也一直在问自己，到底能为公司创造什么价值？目前除上面那堆破代码，我没能创造出什么价值，而一年过去了。无论是对公司还是个人来说，时间都是非宝贵的，“天下武功，无坚不摧，唯快不破”。

我不希望过几年以后会变成依靠资历经验的 PPT 架构师，但我希望写代码能成为副业。怎么实现呢没想好，也许我就是想太多做太少了。

持续学习，持续解决问题吧。

Bootstrapping

Bootstrapping 即引导/启动，比如当我们按下计算机电源按钮后到操作系统启动这个过程，比如某言编译器自举实现。这里我想比喻的是创业初期都必须面对的首要问题：如何将事情启动起来。一旦转起来后，就是不断调整和加速了。启动过程其实是在解决核心矛盾：构想在实现时必要的输入，最输入下换回最大价值。

比如：

- B 端用户还是 C 端用户切入
- PGC 还是 UGC 为主

- 做发布源还是聚合分发
- 分布式还是中心化
- 是工具还是社交

等等这些其实是问题根源，如果没考虑清楚说明思路还不清晰到能够创业，还需要再思考再观察。

大 V

大概是 4 月份的时候通过 GitHub repo 邮件邀请了一些大 V 加入 B3log 开源），引起 justjavac 的烈不满，现在想想自己确实很傻，继续安静地开源，他们那个圈子不适合我。

今年掀起的直播热把很多人想做网红的欲望点燃了，知识变现的方式也逐渐清晰了，但问题是真的有么简单么，没持续干货输出的话对得起雪亮眼睛的群众么。我本来也燃了，搞了两次后灭了，下次直就直接打码吧，兴许在直播时别人还能帮发现 bug....

有不少人建议我出书，但关键问题是我写不出来啊，写代码已经是我目前的极限了，我但凡有那水平写代码就真成副业了 joy

开源

NetBeans

Oracle 甩锅给 Apache 后还需要一定时间才能稳定下来，不过新版的 NetBeans Logo 提案感觉很错：

B3log

B3log@GitHub 已经陆续加入了一些伙伴，也有伙伴开始 create/fork repo 到 org 中了。最令人高的是 Solo, Wide, Sym 这三个项目也加入了不少贡献者，在开发、设计、推广等方面都在推动着项目这样的开源是我一直所期望的，相信这也是很多有着 开源情结 的开发者所向往的。

今年抖机灵做了一个 QQ 聊天机器人小薇，这是一个典型的使用各种开源项目、在线服务强行粘合的序。腾讯的 QQ Web 接口虽然不稳定而且限制很多，但令我没想到的是居然还是有一些人通过小薇易用性进行各种商业的推广。我的视角太开发者，在别人看来程序只是一个工具，我却为之倾注了情。

功能困境

开源项目你做的功能少了，有人说你系统简陋，什么都不能完成；你做的功能多了，有人说你系统太杂，不够人性化。

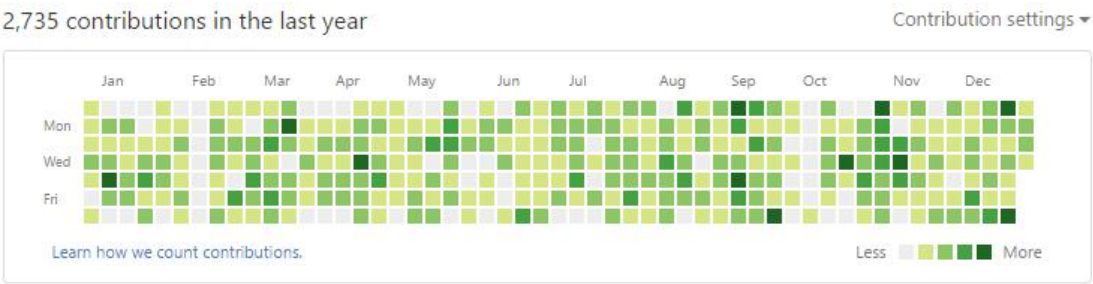
这就是业务型开源程序的困境，就算博客、社区这样“小”业务的开源程序也同样存在，因为程序最

服务的是有特定功能需求的端用户。从这个角度看，业务型开源项目的组织方式是以项目创始人的 **目标** 最为相关，后续参与进来的贡献者都是在帮助他完成目标。当然，这个目标有时很简单，就是希望自好用，也希望有其他用户来使用这个程序。

基础技术开源程序就不存在这个困境，因为可选替代很少甚至没有，而这部分开源项目的目标是大公之间博弈决定的，他们更偏向于技术规范/标准的制定。

GitHub

2016 年我的 GitHub 时间线：



每天都提交一点东西，文档也好，代码也好，就像浇花一样。

另外，请不要问我为什么有那么多时间搞这些、是不是工作不饱和、是不是单身狗等奇怪的问题，因你肯定没在半夜 push 过 GitHub smirk

坚持

不存在所谓的坚持，做任何事总是有动机并且需要有**正向结果驱动**的。做了这么多年的开源，我集结一个社群，认识了一些朋友，和他们一起撸过相同的代码，碰撞过差到不能再差的 idea……

从中我得到了一些建议，而把这些建议付诸于实践，最后得到可验证的结果就是我继续推进 B3 的理。所以根本不存在所谓的坚持，这是一件很自然的事情。说到底，自己做的架构，该自己写的代码终是需要自己来写，对自己负责，对用户负责。

黑客派

起名小达人自认为的巅峰之作被 12 岁网瘾少年认为是黑客基地。

黑客派取自《黑客与画家》，Hacker 就是黑客，Painter 就是画家，分别取前三个字母就构成了 https //hacpai.com ，派字含有画家风格流派之意。

黑客派 2016 年注册了 3.5K 用户，离 10W 的小目标又近了一步....

技术

Hybrid

入门了 Unity ，以 Java 的风格写了些 C#。

我发现最新、最火的一些技术（框架）越来越热衷于将一个语音编译成另一个语言、桥接，比如 Unit、React Native、Dart。以前我只觉得 JVM 系内各种语言已经很复杂了，这下好，直接是不同语系

。

APP 的混合开发也从 Webview 嵌入变成了整合另一个项目的导出，比如 Unity 和 React Native 都以导出 Android、iOS 项目，然后再将导出的项目通过代码方式整合进已有的原生项目中。目前看上没什么大碍，但我担心膨胀下去 IDE 会罢工的。还好我是服务端开发，不是全栈工程师，就凑下热闹。

Java

Java 9 明年就发布了，目前我个人还停留在 7 的规范下，明年是应该在开源中用下新特性了。另外，atke 也是时候翻新一下了，自己用的轮子总得是最圆的吧。

golang

作为前排吃瓜群众，我认为 golang 已经很好用，足够解决我面对的问题。事实上，今年在没怎么大新的前提下 Wide 依然保持了 B3 系开源项目受关注度第一的位置，这足以说明 golang 的魅力。

另外，“golang 在国内比国外火”的观点我不敢苟同，仅就 Wide 而言，大部分 PR、issue 都是歪仁创建的，问题描述和代码补丁都很 nice，很多时候我也疑惑：这帮人都不用搬砖的吗？也许他们确不用搬....

明年该干嘛还是干嘛，在固化的阶级中继续挣扎。