

## 2017 年工作总结

最美好的生活方式，不是躺在床上睡到自然醒，也不是坐在家中无所事事，更不是走在街上随意购物，而是，和一群志同道合的人，一起奔跑在理想的路上，回头有一路的故事，低头有坚定的脚步，抬头有清晰的远方。2017，“翻手为苍凉，覆手为繁华”。

在这一年里，我们发布版本共 16 个，5 个大版本，11 个小版本，我全程参与。跟随公司的脚步，在庆哥的带领下，稳步推进，步步为营，最后实现质的跨越。不仅保持了平台系统的持续迭代更新，更重要的是完成了对原系统基础架构的重构。我非常幸运的参与了整个新系统架构的开发过程，受益匪浅，得到了前所未有的成长。下面我就对 2017 年几个大版本的开发做个详细的总结。

首先，是版本 V2.2.0（首页改版）的开发，这也是我们技术团队的首战，关乎门面，甚是重要。共涉及三个功能模块：活动版块功能、门店推荐功能、评论和点赞功能，我主要负责活动板块的开发。在这次版本开发中，前端界面采用了全新的 vue 框架，一切从零开始，再加上前端开发人手不够，于是开启了“全民 JS 模式”，经过大家夜以继日的艰苦鏖战，一个多月后，终于不辱使命，交出了一份相对来说比较满意的答卷，使得界面的交互体验得到了很大的提升。也是在这个版本中，我们引入了七牛云的 cdn 加速技术来进行“对象存储”，明显的效果就是图片的上传、下载以及展示速度得到了质的提升。没有一项成就的取得是轻松的，初试牛刀，我们就遍体鳞伤，连续高强度的加班，导致大家的体质急速下降，精神力无法集中，身心俱疲，很长一段时间大家才慢慢恢复过来。充分吸取经验教训，这次版本之后，我们和产品部讨论制定了一套适合我们淘璞自己的项目开发规范，包括定期召开项目产品会议，周期性版本开发，大小版本交替进行，规范开发技术文档，细分工作量并合理安排工作时间，杜绝连续通宵加班恶习等。从而做到了有的放矢，为日后持久高效的开发奠定了基础。在此过程中，虽然开发工作非常艰辛，但成长也是巨大的，不仅学到了 vue、七牛云“对象存储”等新技术，还深刻体会到了项目管理的重要性，更是学到了如何在实践中学习的思维方式。但同时，也暴露了自己很多不足之处，比如，代码编写不够规范，学习新技术时，思路不明确，主次把握不当等。

接下来，是版本 V2.3.0，也就是营销活动（大礼包）的开发，我主要负责 MOS 营销管理服务、OMS 活动赠品图片管理、OMS 优惠券规则管理和 OMS 活动赠品统计管理四块功能的开发。作为 2017 年上半年的核心内容，大家都高度重视，做了非常充分的准备，我们技术团队也反复对开发需求进行了几轮详细的分析和探讨，并和产品部一起过了好几遍。在实际开发过程中，充分吸取了 v2.2.0 版本开发的经验教训，将开发工作进行了详细的划分，并分阶段分工开展工作。后端开发人员，先是分工设计数据库文档，接着分工设计接口 API 文档，然后开发接口代码并自主测试，最后分开与前端开发人员进行代码集成；前端开发人

员，先是分工开发静态页面，接着参照接口 API 文档对接接口并进行数据模拟，最后将静态页面集成到项目中去。整个项目开发过程都非常的紧凑有序，这说明项目管理有了很大的提升。对于 vue 和七牛云“对象存储”的技术也运用得更加熟练，代码编写规范也有了一定的提高。但是，也存在两个非常大的问题：其一，就是产品需求太过庞大，开发内容臃肿，存在很多无关紧要的需求，从而影响了主线任务的开发进度，差点失去控制。所幸，在每周的产品项目进展会议上，对原有内容进行了适当调整，删减了几处无关紧要的功能，延缓了报名活动的开发，使得主线功能的开发能够顺利进行，没有过多超出预期，这再次体现了项目管理的重要性。其二，就是低估了原平台系统前后端代码集成开发的复杂性，没有预留足够的开发时间，导致项目后期开发非常的赶，很多地方都无法测试到位，发布测试后出现一堆的 bug，甚至连主流程都没走通。

再接着，是版本 V2.4.0（报名活动）的开发，我主要负责 OMS 报名活动统计管理和 MOS 生态圈赠品统计管理两块功能的开发。因为报名活动只是营销活动的一种，得益于 V2.3.0 的开发，这块节省了很多开发成本。同时在吸取了上次开发的经验教训后，做了如下改进：一方面，精简主线开发功能，去掉花哨操作，大大明确了开发任务；另一方面，增加了一台专门用于集成工作的电脑，在代码集成的时候就能空出一个人手，在一定程度上提高了代码的集成效率；最后，严控测试版本发布基准，必须保证本地主流程测试都通过，才能发布测试版本。但是，由于没有充分考虑到报名子活动的特殊性，将其硬套进营销子活动的设计模式中，导致报名子活动无法匹配原有营销子活动的设计理念，最后导致报名子活动需要重新设计，为此还设计了一套新的数据表结构。再者，受限于原平台系统架构设计，无法真正意义上实现前后端开发分离，从而也无法根除代码集成时所带来的痛苦。

不知不觉，就到了年中。除了上面说的三个大版本开发外，这半年还陆续发布了 v2.2.1、v2.2.2、v2.3.1、v2.3.2 和 v2.3.3 五个小版本，紧接着，又陆续发布了 v2.4.1 和 v2.4.2 两个小版本，作为大版本的补充，小版本的发布基本上就是解决大版本发布所带来的 bug 和暂缓的一些小功能模块，比如统计功能等。至此，我们团队的开发节奏基本步入正轨。

紧接着，是版本 V2.5.0（商户优惠券）、V2.5.1（扫一扫和支付成功页改版）、V2.5.2（大礼包改版）、v2.5.3 和 v2.5.4 的开发，其中，我主要负责 OMS 商户券使用和统计管理、支付成功页改版、COS 会员礼包领取机会等功能的开发。充分发挥上半年所积累下来的技术开发经验，接下来的这几次大小版本的开发和发布都很顺利，基本实现了项目的持续迭代更新。但与此同时，原平台系统架构所暴露出来的问题也越来越严重：一者，难于扩展，认证授权无法控制，很多工作无法开展；再者，运维艰难，发布部署操作非常繁琐，哪怕一个很小的界面改动，都需要重新发布部署；又者，前后端开发无法分离，代码耦合度太高，导致开发效率非常底。时不我待，为此，在庆哥毅然决绝，“英明神武”的带领下，开发团队分成三组：基础架构组、业务开发组和前端开发组，齐头并进，遍地开花。大家自愿开启了”99

工作模式“，朝九晚九，白天开发版本功能和解决 bug，晚上加班研讨学习新技术和 Spring Cloud 微服务系统架构相关知识。

最后，就是版本 V2.6.0（淘璞帮重构）的开发，我主要负责订单服务的开发。经过前段时间的技术准备，在这个版本中，后端我们采用了全新的系统架构——Spring Cloud 微服务架构，它由四大基础服务（配置中心、注册发现、网关服务、认证服务）和若干业务服务（订单服务、商品服务、促销服务、账户服务）组成，每个微服务都采用 Spring Boot 独立开发、测试和部署，并且实现了 Docker 一键发布部署功能。前端也采用全新的 Ionic 框架，独立开发部署，并通过网关服务统一调后端 Rest 接口。至此，前后端开发工作才实现了真正意义上的分离，认证授权功能完全自主掌控，版本发布部署高效灵活，从根本上解决了原平台系统的三大诟病，而且使得界面的交互体验有了极大的提升。但是，问题也出现不少，尤其是前端开发工作存在很多不足之处：一、前端代码管理杂乱无章，导致开发过程中，合并代码时经常出问题；二、没有环境变量配置管理，开发、测试和生产环境无法快速切换；三、无法实现热更新，每次前端界面的修改，都需要重新发布版本。四、前端开发人员对新系统的认证体系一知半解，不知道 token 获取和刷新的意义，导致发布生产后，问题频出。再说后端，由于新系统架构的匆匆上线，很多必要的功能都没有实现，比如：授权、熔断、负载均衡、监控、分布式日志管理、异常处理、单元测试等等。能参与到新系统架构的开发中来，我感到非常荣幸，因为从中我学到了很多有用的新知识、新技术和新技能，再苦再累都是值得的。虽然我对新系统基础架构的学习方法和理解程度还不够到位，但是，我相信，皇天不负有心人，只要继续坚持和努力，虚心请教，总有一天我能把它彻底的啃下来。

回首 2017，我们有坚定的脚步，展望 2018，我们有清晰的远方。以淘璞帮为切入口，我们在 2018 年将陆续完成对 COS、MOS、OMS 和 FRONT 系统的重构，完善 Spring Cloud 微服务系统架构体系。路漫漫其修远兮，吾将上下而求索，万里长征，我们才刚刚起程，要学习和实践的东西还很多，任重道远。我的目标也非常明确，紧随公司的脚步，深入学习和实践 Spring Cloud 微服务架构体系相关知识和必备的一些技能，确保在新的征程中，不落队后，激流勇进，再创佳绩。年轻不应安逸生，但求奋斗镀此身。在奔跑的路上有淘璞的兄弟姐妹相伴，实乃人生幸事也。

2018 年 1 月 3 日

贺汕森