**2017**

北京梦知网科技有限公司

部门：研发部 汇报人：张洪图

人生譬如朝露,岁月不容蹉跎

**张洪图年终总结**

年度工作概述

年度完成情况

02

明年工作计划

04

年度存在不足

03

01

**目 录**

**年度工作概括**

从2016年5月末到现在,我已经入职权大师1年半的时间.总结用一句话概括就是收获颇丰,收获等于我的薪资+我的知识成长+经验成长,知识成长更多是靠自己的学习,在经验成长上,更多是在工作中无形的磨砺,我对自己非常满意,同时非常感谢公司和穆哥对我的支持和信任.

以往的年度述职主要是做出了什么,有什么成果,有什么不足,明年的计划是什么的内容.但是目前我不想这么写我的年度述职内容,查看下自己一年都做了些什么,我记得穆哥在三楼时要求的每个人写出来自己的工作职责.而我述职的内容就是按照我当初自己思考的,我负责java组,那么我都需要做什么,来总结我的述职内容,以及我的总结和反思,当初自己些的内容如下:

  1.保证系统的代码质量,必要时做代码审查.  
  2.进行知识分享,促进团队形成学习和分享氛围.  
  3.提高team技能,从而对需求快速响应,完成更多公司需求.  
  4.保证系统安全平稳上线.  
  5.学习应用新工具来减少团队重复工作来节省非必要的时间.  
  6.进行业务评审,来判断业务的可行性,而不仅仅是需求翻译机.  
  7.尽可能了解全部的业务,保证每部分业务都可控,人员可替换.  
  8.在代码开发过程出现的问题提供必要帮助  
  9.协调团队个人成长需求和公司发展的需求,尽可能兼顾,同时保证个人成长.  
  10.对组员分配任务.  
  11.摆脱某个系统对某个人的依赖,实现人员可替换性,尽可能保证代码可读性.  
  12.对现有系统不足做必要优化.  
  ALL:保证代码质量,增加代码可读性,摆脱对某个人依赖,人员可替换,保证系统平稳上线,进行知识分享提高组员技能,协调组员成长和业务发展关系,分配任务,进行业务评审,保证业务可行性.通过组员技术业务成长,来体现我价值.

借着此次记录,我进行逐条对比,反思下自己的实践程度,保证代码质量,做必要的代码审查.现在接口项目有些积重难返,并且基本上没有重构的价值,同时代码质量的保证比较难,需要保证的就是尽可能在上线的时候问题少一些,旧项目由于深度优化的出现,没有十分迫切的需求进行重构.

在2017年下半年开始做代码审查,就是还是减少不必要的问题出现,在推进代码审核的过程中,java组其他成员还是很不适应的,有种凭什么让你看啊,让你审核的感觉.代码审核推进比较难,记得刚开始负责java组的时候,推进了三次,结果离差强人意还很远的,做过审核后,代码问题依旧很多,现在比较好的事情是查出来的问题,吩咐修改的,都能修改掉.本想搭建gerrit,进行审核,审核不通过的代码不允许提交上去的,由于我的linux知识相对比较差,导致搭建失败.最后亚辉搭建的纯代码查看的工具.深度优化完成后,代码审查需要加强,或许采用穆哥说的方式,周五下午进行代码审核,忙完这一阵认真思考下,并且这个一定在2018年使用,成为和上下班打卡一样的制度,来保证代码的质量.我们的成长在于有计划的提高自己,同时其他的成长也许是通过职场中出现的问题,然后我们对问题的处理,我们不断成长的,千万不要规避不要躲避的就是职场上出现的问题,因为在a公司出现的问题,在b公司依旧可能出现的,所以遇到问题第一个想法就是机会来了,提高自己的机遇来了,机遇以问题的身份出现,大多数人可能却害怕问题.

进行知识分享,促进团队形成学习和分享氛围.主要是我在进行知识分享,但是不能怪其他人不分享.可能是我对其他人要求过高,导致其他人分享被我打击了,这不行那不行,没有促进团队分享的氛围,这个责任有我负责,进行知识分享,遇到其他人可能会遇到的问题,我都主动邮件说明总结分享,知识分享来说,我个人对我自己的分享相对满意,因为每次的知识分享,不管文档还是代码,我都思考再三,如何做到傻瓜式分享,其他人以后看文档就可以自己明白,学会使用.这一点我对自己非常满意,并且发现现问题有价值的事情,我都自己维护我的分享内容.促进团队的学习氛围,这个没做到,分析原因,毕竟学习是自己的事情,如果业余时间自己不主动学习,本身就是it行业人员,不能主动学习自主学习,这个问题实在是我无法控制的.2018年的分享应该是这样的,我思考过,每周一三五下午五点-八点找学习视频让大家学习,从给其他人的资料可以看出,要学习除了自己肯学习,其他不肯学习的,只能逼迫学习.

2017年都分享了什么内容呢?微服务内容六讲(11月和12月),根据我在架构师课程中的内容,我拿出来,然后让大家快速入门,入门口有问题积极归纳总结大家相互提高的,学习可能出现的问题,各种我趟过的坑都写的很明白,对自己的文档编写能力,我本人目前很满意的.同时总结出来应该不应该和值得不值得的思考,负责java组,那么就是应该不应该,如果应该,那么就不要考虑太多值得不值得的问题;如果从个人角度来说,可能更多考虑值得不值得的问题,应该不应该反而是次要的考虑角度.分享内容某种程度上是自我提升,分享不容易,需要组织整理文字和案例,但是这正式体现了菲恩曼博士的,如果你真的会了,那么请你讲出来的学习方法,只要你真的讲明白了,那你一定会了啊.

分享jenkins的使用,搭建jenkins,维护jenkins文档上面,搭建jenkins是运维的事情,但是要学会各种参数的配置的.其中维护jenkins文档内容,不断增加新内容,这个感觉比当初编写文档的时间差不多,比如增加权限内容,增加各种出现问题的处理方式等等,只是得不到必要的回应.同时我也不知道是否有人看过,但是这个问题依旧回归到应该不应该去做,如果是值得不值得上,可能我早就放弃了维护.这个是我做的非常有价值的事情,减少了人工部署出现问题的概率和减少处理类似问题的时间.

Idea的远程调试,之前的测试环境,pre环境,灰度出现问题本多时间是增加log,不断增加log去查看,自从有了这个小功能的分享,直接对测试和pre环境进行断点跟踪,节省了java开发大量的处理问题的时间,以及idea的其他插件的分享,但是都不如这个实用,好用.

若干次git分享,分享了基础原理,还有若干高阶实用操作,但是大家一直都在实用基础操作部分,高阶部分虽然培训了,但是大家都没有很好的使用.对我的内心是个小小的打击啊.从svn到git的推进,手动部署到jenkins的推进,这两个算是我在工具上对研发工具上的更新迭代,同时推进的同时尽可能做出有价值的培训.

得到的经验就是想推进什么东西,那么一定要有你自己的配套的分享和讲解的内容,你知道的不算什么,让大家尽可能都知道都熟悉,才算是你的功劳.推进使用工具,那么必须做必要的分享和培训,不然是没有任何价值的.如果我只是口头上说git好啊好啊,大家依旧还是使用svn的.我说jenkins好啊,有什么用呢,推进与分享,才是比较好的方式.让大家感受到真的好用,那么大家愿意去使用,那么就成功了.

提高team技能,从而对需求快速响应,完成更多公司需求.这个基本上属于自己主动分享的范围,从分享的次数和分享的文档维护上,我个人对我的分享还是比较满意.2017年的分享没有系统性和条理性,最后12月的spring cloud文档的分享,编写文档和代码编写还是相比比较系统的.做spring cloud的入门和使用没有什么问题.快速相应,完成更多需求.从分享中收获的价值,感觉比分享要多的多.有的时候会有这样的错觉,还是正确的感觉呢,不用怕任何事情,因为你分享的东西其他人未必看,看了未必认真学,学了未必去用,剩下的那个可能真的会学一些东西.对需求快速相应,主要在于所选择的技术,以及架构的使用,好比一栋大楼,如果建筑师规划的时候就弄的很差,那么茅草屋是无法搭建出来高楼大厦,如果大厦就是乱七八糟的设计,怎么也建设不出来出名的建筑的.所以这块整体就是架构的能力,2017年参加架构师课程,从思维角度提升很多.所以2017年技术角度的推进就是spring cloud,虽然会在使用的过程中会出现各种类似的问题,但是我还是我有信心不断趟过去的.

保证系统安全平稳上线,穆哥也不想被这些琐事所困扰了,2017年的上线基本上都是我负责处理的java端的事情,整体还算满意,出现问题的次数不算太多,唯一害怕的就是大版本迭代,每次都如履薄冰啊,战战兢兢,还好12月末和穆哥,盛哥,沈哥处理完新版上线的事情.之前的公司上线的流程非常不错,GitLab流程,然后又在现有流程做的修改,整体改动不多,再次体现个事情,有机会一定去正规的公司磨炼一番,这样整体素质的提升确实会与众不同,减少探索的时间.这个也不能让穆哥处理的,本身就是应该我处理的事情.目前对自己还算满意.同时订单深度优化处理的时候,给亚辉去处理上线的事情,测试人员来主导上线,只要有git技能比较好的,遵守上线流程,其实上线出现问题的可能性不会太多.看了<凤凰项目>一书,工作内容可以分为以下几类,业务内容,部门内部项目,变更,计划外的内容,其实我们做了比较多的计划外的内容,那么就是不断填坑,消耗我们的人力的.所以2018年更多是消灭计划外的内容,不断进行正向流程做业务内容和部门内部项目,这一点我的认识很深刻.同时,其实我是个约束点啊,从17年就认识到了,所以我不断找人来承担订单部分的工作,来分担我的任务,减少问题都汇聚到我这里,但是目前来说,去年从邓金磊那里来看,选的人没有很好的分担我的内容,那么今年呢,我一定要拜托问题向聚集靠拢的情况,我需要有人对我进行分担.不然就会有太多问题在我这里积压,其他人处理我还不怎么放心,事实上分派了几次,我确实没有放心的理由.这算是我管理上的进步啊.

学习应用新工具来减少团队重复工作来节省非必要的时间,新工具git,jenkins,流程设置,idea,idea插件分享,工作中我遇到的问题都做归纳总结然后分享给研发群,不管是态度还是方式方法,我对我做的算满意.2017年找了比较多的工具来提高生产力的,好的工具确实可以让人感觉事半功倍的.每每自己发现好的工具都会分享给大家,基本上是一种自动自发的行为,发现好的,一封邮件发送给研发群,想到的比较多的是,自己用了如何,其他人用了会如何的思考,不仅仅从自身思考,思考其他人的时间和思考自己的时间比例差不多.我和我们虽然有一字之差,但是作为做事情的动机来说,导致的差异非常的大.如果是我们,考虑就是如果这么做了,对我们大家有什么帮助和启发.如果是我,考虑的就是这东西对我有什么好处呢.2017年思考团队的内容和时间更多,思考我的主要是自我的技术提升,这个主要靠参加架构师课程来提升自己的技术和经验.

进行业务评审,来判断业务的可行性,而不仅仅是需求翻译机.之前可能做的最多的就是需求翻译机,黑客与画家中也说过,如果编程却不思考,和市政工程中的那些施工人员有什么不同呢,挖水渠挖坑道,16年做的不错,因为对业务不熟悉,17年上边年被各种业务折磨着,17年下半年是思考的最多的,很多技术人员必须要有些产品思考,这样结合技术的使用,感觉事半功倍,可以减少不必要的坑.有时候想想,厉害的技术人员,一定也是比较不错的产品人员,否则技术人员一定是走某个领域的专家路线.本身不是一蹴而就的事情,必须是不断对业务领域的熟悉程度去推进,技术去推动业务,任务技术强大,业务熟悉,才可能做得到的.这一点我很满意,从微服务第一讲就说过,技术和业务的关系,相互促进,只有当对业务熟悉后,技术的威力才会得到增强,同时掌握两把刀,技术和业务一致对外;如果不熟悉,那么只有用技术去死磕业务.如果在一个公司要有向上的发展,那么一定是技术和业务两把刀,缺一不可.2017年很多时候是讨论业务,不是被动地参与业务讨论,当然这过程中还是有很多问题的,比如烦躁地时候好似死猪不怕开水烫的感觉,这就是被动的接受.更多的需要我去思考,思考为什么要这样,是否有必要,以及做的时候是否可以有灵活的处理方式.

尽可能了解全部的业务,保证每部分业务都可控,人员可替换.开始我是这么思考的,但是发现执行的过程中有很多人员时间的浪费,这算是对我的历练.尽可能了解全部的业务,订单流程业务,下单部分到递交审核过程中,我可以说我最熟悉.但是发现递交后台那块其实不怎么熟悉的,所以只能说掌握70%,审核流程直到递交到商标局那块不怎么熟悉的,深度优化那处理发现比较深刻,这块弥补很多.2018年将审核流程部分和递交到商标局这块彻底掌握.保证每部分业务可控和人员可替换,这部分浪费比较多的时间,比如用户部分,让志江,亚辉都掌握;消息中心,亚辉,畅通,金磊掌握,其实不需要这么多人掌握的,两个人掌握就好,再多人掌握是浪费,因为人员改动多,其他人都不熟悉,那么最后还是需要重新熟悉的,这块浪费比较多的时间.并且在有些产品会议的时候,很多人其实不需要参与的,但是为了达到业务可控,人员可替换,浪费人力.这是2017年下半年想到的,目前及时纠正这之前的错误做法.

协调团队个人成长需求和公司发展的需求,尽可能兼顾,同时保证个人成长.每个人都需要成长的,比如有的人有技术成长的诉求,有的人有管理成长的诉求,下半年主要推动java组内业务问题负责人,分配给谁了,那么就有谁积极主动去推进,去联系其他组的人员,这个我还是认为做的比较好,给每个人多一些主动负责去推进事情的机会,但是如何掌握机会,则是每个人的事情.这涉及到个问题,那就是,你怎么看待在一个公司内的机遇问题.事实上把握住了机遇其实也是有代价的.那就是如果你赶上了机遇,却没有很好的利用机遇,没有踏着机遇向上前进,那么机遇就会下次远远地躲避着你的.所以尽可能用心做好事情,而不是做完事情,本质上真的有很大的差距,同时给每个人的提升和更多的机会,是截然不同的.世界上最富有的人中有一段话很好,那就是工作是你最好的朋友,要花时间和金钱来投资自己.思考自己,思考自己的工作,否则去什么公司都是一样的.

**年度完成情况**

12月,完成微服务六讲文档编写,同时进行微服务第一二讲内容,第三至六讲以文档形式下发给java组.完成深度优化项目框架搭建,以及深度优化任务排期,为技术人员编写深度优化的理解文档.

11月,组织整理商品优化设计的文档和数据库设计,组织和整理深度优化的数据库设计和文档编写.优化深度优化接口参数传递,减少参数和对象的转化,使用新版参数校验规则.参与商品优化的推进.开发和推进海外智能注册,处理安全项目的上线推进.拆分出来商标搜索项目和专利搜索项目并且上线.

10月,完成spring boot为基础的深度优化框架搭建,单元测试,工具类,深度优化业务掌握.整理完成4-5份深度优化问题归纳.完成安全项目的开发和测试.和先潮讨论深度优化的业务,将第三方订单,智能注册订单,交管订单,附件内容,全部进行优化,使用tk.mybatis来开发处理,这算是对整个订单部分的除了已经有的查询无法修改,和下单等内容,其他内容关于订单部分已经全部修改完成.主要在海外注册订单,以及安全处理,两个大模块的开发过程中,同时进行优化开发.

9月完成spring boot版本的订单优化开发,旧版参数传递接口的处理,增加单元测试,但是这个并不是我所想要的,这个架构内容.还是在11月12月推进使用spring cloud技术架构开发,方便未来的可拓展性和根据使用spring cloud来强迫大家来提高自己的技术能力和水平.

8月,处理批量支付和再次安全控制id,ip,method等内容,给java组讲解spring boot的框架使用,有8月份的spring boot的基础,那么理解使用spring cloud将会有大的帮助和裨益.

7月,和先潮了解和掌握深度优化的业务内容,同时整理spring boot的架构内容,使用tk.mybatis,yml,代码反向生成工具,全局异常处理,代码层面数据库读写分离.

6月.处理钉钉产品开发,协助处理担保商标注册,研究多线程知识分享,mybatis内容分享,交官文件开发处理.处理商标接口被刷数据的第一次安全处理开发.

5月,对git原理部分进行分析分享,促进和推进java组的沟通交流站立会等内容的实现,分享idea的远程调用的使用.

1月-4月,主要是两个中心的开发和调试,以及新版网站的开发处理,这段时间属于疲于奔命,业务的实现,而不是思考,更多的是下水工的工作方式,少思考,疲于应付开发的事情.这段时间应该推进使用jenkins.还有大部分时间和任务是处理代码分支合并的事情.

从2017年12月内容,反向一步一步总结每个月的工作内容.

工具改善,从开始就是以产品为主导,我们以下水工的行为去开发,少思考,完成实现,导致问题很多很痛苦的,并且从开始使用粗放式地开发,手动部署到开发环境,使用单独的svn分支去开发,到使用git,jenkins,idea的插件开发,逐渐使用新工具来减少人为出现的问题.工具上逐渐发生改变.

参与业务,从开始产品主导,慢慢我们熟悉业务,可以不断提出来自己的主张,而不是产品说一不二的内容,因为不断熟悉业务内容,所以才不断有发言权,这个过程中,经过不断地思考业务不断思考开发,不断思考和总结发生的事情,有的地方,不足的地方,然后才会不断进步,事实上,我从4月开始随着思考的内容不断增多,处理比较多的问题,个人认为减少了和弥补了业务思考的不足.

技术推进,从整理redis的入门思维导图,spring boot中代码层面读写分离,yml文件,全局异常处理,返回值统一处理,传递参数格式调整,微服务分享等内容,我个人进步很大,推动的速度可能比较慢.明年我会推进rabbitMQ的分享和使用,mysql的分享和使用,源码阅读的分享和使用.

管理提升,从原来的简单直接,到现在动脑多一些,虽然我尽可能换位思考,尽可能从每个人的角度去考虑问题,但是我不可能兼顾每个人,需要对公司负责,和对部门内部人员负责,管理从来不可能是面面俱到,兼顾到每个人.同时,我也无法让每个人都满意,做自己应该做的事情,分派任务,和每个人的沟通,需要提升沟通的方法和策略,管理事情,领导人,这个是我18年需要学习和改善的,以及重点突破的内容.个人认为管理上有进步,但是目前来说我个人不满意.

**年度存在不足**

管理上的不足,虽然尽可能兼顾公司和每个成员,但是发现没有让每个组员有明显的进步,这个需要我思考的,如何通过我的分享让每个人有明显进步.转换下问题是,如果我培训大家,怎么能让我的培训有效果,让每个人都能明白听懂,这个需要思考.

如何做到尽可能做到程序的可拓展性,互联网行业的情况就是需求多变,现在很多时候做不到及时响应需求,这固然有产品的部分问题,但是既然是这个行业,那么就要多思考这个问题,尤其业务复杂的时候,这个应该作为互联网行业优先考虑的问题.那么涉及到个问题,就是代码可读性,个人认为虽不能打90分,个人给自己打80分,还是可以的,代码可读性,基本上没有问题的.

个人管理或领导的不足,看不到问题才是最大的问题,事实上很多时候虽然我思考和反思,但是我依旧看不到我的问题,那么就需要不断学习新方法,拥有新的思维,换个作者的角度去思考我的做法的高低和好坏.计划阅读个20本书来提高自己的管理和领导能力,认真去实践,同时减少其他人在我这里忍受的痛苦.

技术上的不足,今年参加个架构师课程,拓展了技术面和思考层次,但是感觉还是不足,没有足够的技术深度去思考问题,只是从运用和浅层次原理上处理,需要加深部分技术的深度.

**2018年计划**

技术计划:研究源码,分享研究源码过程中探索出来的方式方法;研究mysql技术,达到比较深层次的使用,熟悉知晓大部分的运行原理,同时加以分享;学习运维技术,现在深刻感受到运维技术对我的限制,这个是对我来说是个瓶颈.源码研究和mysql同时是要进行分享的内容.

突破计划:一直有想下一本关于rabbitMq的java版书籍,争取把这个实现了.

读书计划:2018年阅读50本社科经管书籍,阅读20本技术书籍加以实践.今年完成64本社科书籍,5本技术书籍,1次为期4个月的架构师课程.2018年成为1个有优秀学习方法的学习者,在学习法上有所突破.

部门计划:把ELK做分享,推进ELK的使用.然后加深对spring cloud的方方面面的使用,以后项目的推进和开展,以围绕spring为核心进行.

公司计划:努力做好迎接好公司200万以上订单量的java技术能力支撑.

**最后**

最后,感谢穆哥和公司的信任,让我管理java组的事情,通过推进java组技术和业务的事情,我个人对我的进步分享的满意.那么关键是要有这样的机会,所以非常感谢公司和穆哥能给我这个机会.

在2017年,我对自己的进步,自己的思考和反思,以及不断提升自己做事情的动机和能力方面,都得到上很多的进步.同时争取2018年,随着我的进步能推动研发部门的技术进步和整体进步;同时也希望随着公司的进步,我也能随之有更大的进步.

积极主动去处理事情,积极主动去思考,多反思,多总结,一定会有巨大的进步,不要害怕犯错,但是不要犯不必要的错,尽可能少犯重复的错.