加班到凌晨 1 点之后的一些思考

作者：xiaodaojava
原文链接：https://ld246.com/article/1590497721246
来源网站：链滴

先来一张皮卡丘镇文!

鬼知道我经历了什么

看了小刀朋友圈的应该都知道,昨天小刀加班到了1点才下班,好久没加班这么晚了... 可能对各位程序员说, 加班已是家常便饭了. 但真得是太痛苦了, 也不得不感叹老了. 想大学那会, 有好友来学校找我玩, 然就带着去网吧通宵... 然后第二天接着上课啥的,不耽误, 然而今天感觉飘了一天, 干活也集中不了精力. 以今天一直在思考,为什么会加班? 为什么会加班! 为什么会加班!!!

像催命符一样的"上线"

上线这个词对所有的程序员来说, 都像是催命符一样, 同样也是加班的代名词.那么上线到底可怕在哪? 觉得可怕在于未知. 就是不知道这个代码发到生产环境是否能正常运行, 是否还有隐藏未知的bug等等, 一旦出现了bug, 影响到了现在的业务, 那势必会带来不少的影响.

有一句笑话,说每次上线都要"杀一个程序员祭天", 联想到古时,祭天是祈祷平安顺利. 为何要祈祷平安利,因为未来的不可知.

如上两段,我都提到了一个词,未知. 对. 就是这样, 因为我们对上线之后的运行情况未知,因为我们对未来知,所以才会对上线恐惧.那只要我们解决了未知, 是否就可以让上线成为胜利的号角. 为了解决未知, 除像奇异博士一样,能看到古今. 我们虽然看不到古今,但我们可以多写单元测试,正规输入的,不正规输入的其他乱七八糟输入的等等, 多写多测(这里很想提一下 测试驱动开发), 这样一定可以暴露很多问题,把上的风险降到最低.

回想起我们在开发中, 其实单元测试写的很少, 忙是一点,但开发人员自己的疏忽也是一点,以至于很多题都是在联调的时候暴露的.但联调的时候,用例只有几个,能跑通这个流程,基本上联调就过了,并没有带真正的业务场景数据,所以在上线之后,线上回归之时,又会暴露很多的问题. 然后就不停的改,不停的发. 里的压力也会越来越大.

本段关键词: 未知, 测试用例

开发人员的分工

这话要从古代打仗时说起了, 打仗时期的人员配比都是经过鲜血的洗礼 这里我想说的是"三三制战术", 个人分别负责: 进攻-掩护-支援. 那放到我们现在的工作中, 一个项目,是否也要配比三个人? 一个开发, 个测试, 一个产品.

在小公司中,不可能有这么多的人员, 那么就是一人多用. 这里的一人多用, 有两种解释:

1.某人又当开发,又当测试. 这个和现在说的DevOps,开发自测,开发交叉测等有些类似.

2.一个人可以同时兼顾多个项目的开发. 当然我们现在也确实如此,一个人要维护4-5个项目, 然后一个试也是负责多个项目的测试.

如果是采用1方案,就等于是划定了个地盘, 你自己的一亩三分地, 有人加入就有人能分担,没人加入,就己扛起来. 第二种方案,如果是一个人还好,如果是多个人,就难避免的出现人员职能交叉的地方.比如三开发,都要在这个项目上进行开发. 那这个项目到底是归谁负责? 由谁来主要维护? 但凡是出了问题, 要就是看谁倒霉,在谁开发的时候报错了,就让谁解决,要么就往上抛,让领导来决定由谁来解决.

我们现在就是处于方案2这样一个比较混乱的状态,这样就直接导致大家都不想接手某个功能,某块代码.

人少活多不是借口,关键是要理清关系,责任人,但凡团队人数到3人以上的时候,就要好好规划一下人员工,职能分工. 不要三个和尚反而没水喝了

本段关键词: 三三制战术, 人员分工

本文总结

写代码是一种艺术,艺术来源于生活. 我始终相信一句话, 计算机中的种种设计,一定都能在现实生活中到影子! 那么我们就要用心去感受生活,去体会生活, 去思考生活,然后再反过来,去思考代码,去思考设计等. 祝大家早日脱离加班的苦海!