

2.人件读后感

2.1.错误在所难免

营造一个不容许任何失误的氛围会让大家持有戒心。他们不愿去尝试那些有可能变坏的事情。当你试图体系化流程时，当你倾向于墨守成规时，你就在强化这种戒心，于是大家就会人为地被禁止做出关键的战略决策，因为他们害怕犯错。在不允许犯错的规定下，或许平均的技术水平会稳步提高，但团队的社会氛围）却会遭受可悲的伤害。

与营造一个不容许任何失误的氛围相关的事情往往是：将犯错与绩效挂钩或将错误公示(犯错人员或团队受到心理或经济上的惩罚)。

之前在公司实习，我们也有这样的举措：如果某产品线或团队要紧急发布(非周二或周四发布日发布)，则计分板上会在这个团队后边贴一张猪头图片，这个计分板放在人人都会看到的公司前台。虽然看起来这像是一个开玩笑似得防止团队紧急发布的小措施，它也确实起到了作用，上海研发中心的紧急发布次数从每周几十次骤降到十几次，但是它仍然起到了不好的影响：

1. 开发人员不会再因为某些小故障去重新发布以优化用户体验，明明程序很容易改动，也一定要等到下一次发布日再发布。比如，有一个较慢但又对性能不会有严重影响的SQL。
2. 开发人员对新技术的引入变得更加谨慎，更倾向于用大家都在用的已经存在的代码，而非自己去写，即使明知这些代码质量和性能不高。比如，不再使用新的反应式编程框架自己封装而是继续沿用老员工封装好的工具类，即使明知道这些工具类性能不如自己独自实现。

就像书中所说的那样

他们会把问题藏到脚垫下面，或者直接扔给产品的最终用户。

后来公司技术大佬也发现了这个问题，开会投票后取消了这个限制，采取了积极措施的去减少错误(紧急发布/故障等)，比如：

1. CodeReview人数增加，DBA对开发人员的SQL进行Review后才允许发布
2. 部门或团队之间的沟通交给TeamLeader或对应的PM，减少普通人员的沟通成本，提高生产时的专注度
3. 适当砍产品需求，特别是不合理的需求!!!
4. 对于新产品线不限制发布时间与次数；对于稳定产品线实行发布积分制度(如果没有紧急发布积分会一直增加，如果有紧急发布则会消耗积分，积分为0后只能借其他团队或产品线的积分，积分最高的团队团建经费由 300/人月->600/人月)

2.2.质量是免费的

质量是免费的, 但只有愿为质量倾其所有的人, 质量才是免费的. 价格和质量的对立在日本并不存在; 相反, 高质量带来成本的降低却是被广泛接受的想法. 让制造者来设定他们自己满意的质量标准, 会带来生产效率的提高, 从而抵消为提高质量而产生的额外成本。

要保证软件的质量我们比如投入足够多的人力及物力, 在这种情况下我们的质量得到保证, 高质量的软件在可读性/可扩展性/性能等方面带来的好处, 远远高于提高质量所投入的成本。

在从公司来到学校后我深有感触: 之前在公司, 我们基础组的RD:QA \approx 1:1.5, 业务部门的FE:RD:QA \approx 3:5:5, 而且有完善的CodeReview流程, 代码的可读性非常的高, 当你接手一个项目只需要读代码就可以很容易的了解项目的范围与脉络, 你去调用老的代码基本不会出现bug。

来到学校后, 我也接手了一个同学的项目, 项目属于demo级别, 但里面的坑并不是demo级别的, 比如: 使用非GA版本的框架, 曾调试一个BUG两天, 最后发现是框架本身有BUG, 在切换到GA版本就可以了; 随处可见的魔数, 经常需要输出数据调试代码, 才能搞懂变量具体的含义; 硬编码一些变量, 比如, 服务调用直接硬编码localhost进行调用, 当部署到服务器上便无法使用. 在这里并不是吐槽这位同学, 因为我甚至每个人, 在特定环境下(赶时间完成某项作业/第二天就要拿出demo), 进行编程的时候, 都会比较随意。

其实上边提到的一些问题, 只要我们遵守一些约定/规范, 很容易改正, 我们甚至不用CodeReview也不会花费很多的时间成本, 但是当开发时, 如果一味的追求速度, 对代码质量毫不关注, 在我们去调试并尝试修正这些问题时, 我们会花费数倍的时间。

综上, 建议在开发时花费一些时间去确保代码质量, 这在你不知道的未来会节省很多时间。

2.3.创造环境

我认为她简直就是启发式管理的活榜样。一个下雪天, 我拖着病体搭建着我们那不够完善的系统, 准备用户演示。莎伦进来, 发现我在控制台前强撑着精神做事。几分钟后, 她端着一碗热汤出现了。

了让大家能够顺利工作, 有时候你不得不对那些为阻碍工作顺利进行的因素采取强硬态度。造成工作时间流失的原因很多, 但其实每一项都大同小异。这样那样的失败通常是一可能大部分都是一一由于组织提供给你的工作环境造成的。电话铃不停地响, 打印机维修人员顺道过来聊聊天, 复印机不工作了, 捐血组织者打电话说时间改了, 人事部不停催促更新的能力调查表, 下午3点之前要提交时间表, 更多电话打进来。然后一天就这么过去了。这样的一天让你没能有效工作一分钟。

确实是这样的, 之前实习的公司是开放的办公环境, 大概像这样



这种办公环境虽然看起来很开放很不错, 但实际你在写代码的时候: 你的同事可能在和另一个部门的人打电话沟通; 花草维护工推着一辆哗哗响的车子在你旁边经过; 你同事在旁边吃着香喷喷的外卖, 你都能分辨出他今天吃的徽菜还是江浙菜. 到处都是声音与味道, 会让你的工作效率大大降低. 如果能选择, 我一定会选择有隔断工位或空间较大的公司, 像这样:



图片来源: 视觉中国 www.vcg.com

可能管理者很难决定这种环境(空间小,工位多), 但是完全可以采取一些措施来优化这样的环境, 比如:

1. 吃外卖必须去茶水间
2. 打电话必须去茶水间或会议室
3. 闲杂事情(如打扫卫生/浇花)在上班前(一般11点开始上班,时间完全够用)搞定
4. 为每个员工提供一个优质的降噪耳机