

wmnothing的专栏

目录视图

摘要视图

RSS 订阅

个人资料



wmnothing

访问： 121341次
积分： 2571
等级： BLOG > 5
排名： 第8475名

原创： 101篇 转载： 0篇
译文： 0篇 评论： 313条

文章搜索

🔍

文章分类

- 开发管理 (61)
- 技术研究 (18)
- 美好思故乡 (10)
- 读书破万卷 (13)

文章存档

- 2015年09月 (1)
- 2015年07月 (1)
- 2015年05月 (3)
- 2015年03月 (1)
- 2014年08月 (1)

展开

阅读排行

- 用批处理实现Windows下 (7437)
- 企业文化与“酱油党” (7035)
- 代码重构需要亲力亲为 (6542)
- 怎样简单检测socket的健 (5837)
- 解决C盘剩余空间不足的 (3190)

2016软考项目经理实战班 python编程常用模板总结 【博客专家】有奖试读—Windows PowerShell实战指南

代码重构需要亲力亲为

标签： [delete](#) [工作](#) [优化](#) [网络](#)

2012-09-12 00:55 6546人阅读 评论(4)

分类： [开发管理 \(60\)](#)

版权声明：本文为博主原创文章，未经博主允许不得转载。

代码重构需要亲力亲为

吴旻

泰岩网络工作室

这段时间我处理了大量代码和业务逻辑的细节问题。

应该说，我已经快有1年没有关心这些细节了。没关心是觉得弟兄们希望自己能处理好这些问题，我应该适当相信他们；现在出来关心则是发现弟兄们对这些问题的把握有偏差。

第一个让我觉得吓了一跳的是代码的行数。功能没加多少，代码行数则快翻番了。我看到的最大的一个文件有将近2000行，我凭直觉认为至少可以砍掉一半行数。这些代码的某些函数相似度极高，完全可以合并成一个。无论是业务骨干，还是初级程序员，大部分人都是在不停的添加代码，哪怕根本不需要添加代码；极少人能静下心来把这些代码进行整合、优化。

第二个让我觉得惊心的是代码架构变成了“高度化中央集权制”。其实只有少数参量是需要对外公开的，但事实上却是全部参量都放到了一个公开的类中，这样做的目的很明确，就是想什么时候用什么时候用，想在哪儿就在哪儿。但“中央政府”不可能替“地方政府”做好一切，“地方政府”也不可能为了屁大点小事，还得去“中央政府”请示一下。面向对象的一个原则其实就是“属地”原则，自己能解决好的事情，就别闹那么大动静了。

第三个让我觉得不能理解的是运行效率问题。几个重要的工程，运行起来都很消耗CPU，这让我极度不解。实际上这些程序大多是在进行数据的简单运算，应该不需要什么CPU，仅仅需要有限的内存而已。两点典型的错误是，一个程序在不停的new和delete内存，其实完全可以一次申请，多次复用，最后退出前释放；另一个程序则是在不停的写文件，虽然这些数据需要保存到文件中，但不是必须立刻写入到文件中的，可能半分钟写一次就好。

第四个则是我认为有些代码取得的成绩没有引入的问题多。一个就不能正常运行”。我不得不一次又一次的重申解决问题的方法有很多，而现在问题的关键是，没有人告诉我他需要帮助。

第五个应该是大家都不太愿意理解并简化业务逻辑。代码最终还代码的意义就不大了。我不停的在问弟兄们，你觉得怎样才是合适的？默。也有人向我解释了半天，也无非就是说，这个逻辑不是他定的，代因的。我想这也是第一个问题所说的代码膨胀的根本原因。



情侣酒店



- 程序员心中的“高山”与“大” (2156)
- 我们怎么问问题 (2128)
- 程序员心中的“高山”与“大” (1807)
- 项目管理就是整合业务链 (1764)
- i++与++i的汇编级实现 (1747)

- 评论排行
- 企业文化与“酱油党” (59)
- 代码重构需要亲力亲为 (47)
- 我是一个“太极高手”吗？ (13)
- 项目管理就是整合业务链 (12)
- 敏捷----功夫在工具之外 (10)
- 技术大牛？ 业务大牛？ (8)
- 青春就是用来浪费的 (8)
- 水与火的选择：对付路边 (8)
- 软件工程的“政治”与“经济 (7)
- 面向变化编程 (7)

- 推荐文章
- *基于直方图的图像增强算法 (HE、CLAHE、Retinex) 之 (二)
- * python绘制非常漂亮的图表
- * 数据库性能优化之SQL语句优化
- * 拉开大变革序幕（下）：分布式计算框架与大数据
- * Chromium网页URL加载过程分析
- * Hadoop中止下线操作后大量剩余复制块解决方案

最新评论

fix/fast协议中PMAP（存在图）和jinghuayanyu: group也是这样，有没有PMAP是根据其所包含的filed是否需要PMAP中占位，但是我不明白的...

fix/fast协议中PMAP（存在图）和wellway: 具体要怎么解析操作尼？大哥

父辈的教育理念

wmnothing: 今天再总结一下：我愿意为你当牛做马，但你得听我的。

VS2008宏不能运行的解决办法

ahdung: 有用，非常感谢！我的是VS2010，需改vsmsvr10.exe.config才行。

无论你说什么，都有人不能或不TYTSN: 带团队，遇到这样的人多一点的话，光擦屁股就够了。遇到这样的必须要踢出去才行。

用批处理实现Windows下子目录: JoeBlackzqq: 谢谢，非常有用。昨天电脑跑了一天，今天回来想创建一个工程，竟然提示空间不足，我看了下，只剩下几百KB...

有想像力的程序员才有未来

xiaoshuq: 服务器网卡能随便换？这管理也太混乱了吧。评论朋友说的真不错，作者真是有一种本位人上的优越感。

面试与程序员

huxinguo1983: 好

我们怎么问问题

nschoy: 还没有修炼到家,因为我很多时候觉得那两句话不是用同

事实上，我所在的团队近年来取得了很大的成绩，也得到了大家一致的认可。不过我确实对这些代码非常不满意。代码重构这件事，我也叮嘱过几个骨干人员要做好。但显然我们对什么是“做好”产生了分歧，他们理解的“做好”就是运行起来没问题，而我理解的“做好”是傻瓜也能看得懂。

在相当长的一段时间里，我都感觉到程序员不太希望我出现。我的出现对他们似乎有太大的压力，于是我尝试着淡出他们的视线。他们偶尔也会来找我，但基本上是在无论如何也进行不下去之时。我经常笑谈，弟兄们只留给了我半天时间。我明显能感觉到，程序员们浪费了大量时间去解决一个什么问题，最后只剩下半天时间了，已经到了完工的截止日期了，突然出现在我面前，告诉我进行不下去了。要么希望我在半天内帮他解决这些困难，要么希望我宣布工作延期。

我偶尔会感叹，为什么不能留给我一天半的时间呢？

身为团队负责人，事无巨细都要管是不合适的；但代码重构这件事，看来还真的是需要亲力亲为。

顶

踩

上一篇

程序员心中的“高山”与“大河”（二）

下一篇

软件设计需要有一点历史观

我的同类文章

开发管理（60）			
招人遇到的奇葩事件	2015-05-12	阅读 379	无论你说什么，都有人不能... 2015-05-08 阅读 385
遇到没“人性”的管理：你真...	2014-06-03	阅读 668	对的是什么？ 2014-01-21 阅读 637
监控维护人员职责	2013-09-23	阅读 1257	敏捷----功夫在工具之外 2012-12-13 阅读 816
我们怎么问问题	2012-11-13	阅读 2128	程序员心中的“高山”与“大河”... 2012-08-21 阅读 2157
程序员心中的“高山”与“大河”...	2012-08-17	阅读 1807	
更多文章			

主题推荐

重构

工作

网络

猜你在找

大数据时代的<集装箱式>架构设计与Docker潮流	软件架构师应该知道的 97 件事笔记
开源大数据技术架构设计	软件架构师应该知道的97件事
高并发集群架构超细精讲	软件架构师应该知道的 97 件事笔记
大型门户架构高级可扩展方案讲解	《软件架构师应该知道的97件事》
PHP面向对象设计模式	软件架构师应该知道的97件事笔记
关闭	

运势揭秘

属鼠2016运势 属牛2016运势 属虎2016运势 属兔2016运势 属龙2016运势 属蛇2016运势 属马2016运势 属羊2016运势 属猴2016运势 属鸡2016运势 属狗2016运势 属猪2016运势


查看评论

33楼 zhuzhaoze 2012-09-24 15:25发表


在这个问题上有很大的共性，如果团队中每个人都用心一点会省去很多不必要的麻烦，有时候并不是不能做，而是不用心。

32楼 itwangpan 2012-09-24 15:22发表


一个意思.特别是领导的变换比我要快很多的时候.
代码重构需要亲力亲为
nscboy: @jAmEs_:客户关心的是疗效.如果真有效.我想大部分客户还是不难说话的.客户也是人.你投入了精力...

 打酱油的比较多，当你真正热爱编程的时候，就会注重每一个细节了


31楼 [BadPattern](#) 2012-09-24 13:42发表

 表示我对重构很重视，有些代码烂的生出很多问题，缝缝补补，成指数级腐烂，花几个星期重构了一下，顿时问题少很多。


30楼 [YANGYW_112299](#) 2012-09-24 10:22发表

 其实大多数情况下，垃圾代码的产生全都归结于项目负责人把开发周期压的太紧。。。下面的人不是不会做或做不好，而是做这些事情需要时间，而且是大量时间。。说到这，吐槽一下，那些整天不看代码只想出版本的项目负责人真该死！活该项目全是些垃圾代码！


Re: [wmnothing](#) 2012-09-24 11:44发表

 回复YANGYW_112299：两方面说吧，我见过因为工期短而差的代码，也见过工期足够长而差的代一个原因，但这里面的水分其实很大，有些人用一天就能做完，有些人一周也做不完。所以，需要！


Re: [nscboy](#) 2014-01-21 20:19发表

 回复wmnothing：有些代码,投入一天就能工作,投入一周会工作的更好.这样的代码很少有,天的时间去优化.虽然优化后会对后期维护节约好几个月的时间.现在的问题是,要不是被需求逼迫着.没有人原意去动那些已经完成的烂代码.


Re: [sanguorewrite](#) 2012-09-24 11:23发表

 回复YANGYW_112299：我觉得很多重构后脱裤子放屁的代码更难重构或者说更难维护。简单才是硬道理


29楼 [zhouqm27](#) 2012-09-23 17:51发表

 我在工作中也不停的重构，因为工作中任务比较多、工作量很饱满，所以必须是抽时间去搞，例如周末加班，不这样搞就感觉天天面对看不下去的代码难过。结果大概一年来才抽出不多的时间大体重构了一番：两三万行的模块应该有一半的代码被我增加、修改、删除过，还在继续整理中。因为我的整理，也搞出重大工程事故来，但这样的情况都是原来的代码写的不清楚、小组内其它成员也不知道啥意思.....等处理好故障反过头来搞清楚原因后，发现往往都是原有代码可读性差造成的....成果很显著，有的文件约一半的代码被我砍掉，可读性极大提高。估计新成员要看清楚、上手就很容易了.....


Re: [wmnothing](#) 2012-09-24 09:40发表

 回复zhouqm27：实践出真知。感谢你和我有相同的体会和经验。相信我们还有更多共同的语言。


28楼 [SelfMedicated](#) 2012-09-23 14:31发表

 我还是个在校生，但我有实习经历。当我看到"为什么不能留给我一天半的时间呢？"我的心突然沉了一下，我想这应该是一个沟通问题吧？我有一个疑问，就是作为一个程序员，哪些问题就应该自己解决，哪些问题应该及时向大家说出来？我之所以有这样的疑问是因为，你老是在问别人的话，就显得太无能，你要有问题却老不问别人，那就会带来很多负面影响。而当我在一门心思的专研一个问题却没有结果的时候，到底是"问问别人"还是"自己解决"，我就把握不住了，或者说我把握不好沟通的时机，请问谁能点化一下我吗？或者我换个角度问，楼主希望手下的程序员什么时候问你问题，而什么样的问题，你希望程序员能自己解决呢？


Re: [wmnothing](#) 2012-09-24 09:41发表

 回复SelfMedicated：容我想一下，这件事不是一句半句话能说明白。如果你能知道自己想做什么，也知道别人想做什么，原则上就没有问题了。但事实上肯定是不知道的。

Re: [zhouqm27](#) 2012-09-23 17:54发表


 回复SelfMedicated：多问多沟通，不会有人觉得你傻的。不耐烦解答的人，一般都是没有责任心的人。

27楼 [anhuixiaozi](#) 2012-09-23 10:39发表


 非常赞同楼主的说法
记得刚进项目组，看了前辈们代码，不是不好，觉得看的头都大了，一个类就一个方法，一个方法几千行，现在逐渐重构，需要一个过程。

关闭


26楼 [ruleless](#) 2012-09-22 12:10发表

 现实如此啊！我觉得很多吃程序员这碗饭的并不想在技术领域成就卓越，再看看CSDN的抱怨贴、迷茫贴，看得很多没多大定力的人也开始浮躁了。我觉得做一份工作还是要有热情吧，有工作热情的人才会自觉将工作做到完美！


Re: [wmnothing](#) 2012-09-24 09:38发表

 回复ruleless：如果可能，我们应该学会喜欢；如果学不会喜欢，那么就学会“认真”吧，认得清哪些是“真”，哪些不是真。有些事情确实复杂，但也正是因为这些复杂，才区分出了能力。


25楼 [zhousmq](#) 2012-09-21 21:10发表

 真希望能有你这么一个领导带着我！


Re: [wmnothing](#) 2012-09-24 09:45发表

 回复zhousmq：你以后就会是这样的领导，努力吧！


24楼 [ww63176491](#) 2012-09-20 20:57发表

 代码走查的任、权利可以下放到某个人或者几个人组成一个小团队，定期进行代码走查，这样避免你什么都需要亲力亲为，有些事情完全可以交给下面的成员帮助你做一些过滤工作。另外还有一个思路：与其关注代码重构，不如逆向而行通过其他方式逐步弱化代码质量对开发造成的影响。


23楼 [sanguorewrite](#) 2012-09-20 19:06发表

 “做好”是傻瓜也能看得懂 完全同意。
假如lz把参数东一个西一个的放。改一地方要动N个文件，这样的代码架构才叫丑陋。维护起来也和你说的“悖悖”。


22楼 [yy2004005](#) 2012-09-20 17:56发表

 很能理解你，新加入一个团队在对他们的代码就行阅读时就发现问题跟上述一样，很多事情需要我们来自己来。


21楼 [蓝斯](#) 2012-09-20 09:20发表

 主要还是时间问题
有时间啥事都好办
不然赶项目都来不及哪还考虑得了那么多
做好了没功劳
做不好还怪我们延误进度
最后看你没事做了又安排其他事情给你


Re: [nscboy](#) 2014-01-21 20:23发表

 回复蓝斯：要把项目代码交给你负责,这个问题就好解决一些了.你会为现在的优化和重构在以后获得丰厚的回报的.
要是开发不断换人,这事就没法做了.这样不断换人的项目还是早死造托生吧.


20楼 [he036002](#) 2012-09-19 17:12发表

 说的都是很基础的东西。事情难就难在基本功。一直也在坚持基本功。但如果领导催的紧的话，我说的领导是指你的骨干成员们，最底层的码农们就不得不急功近利了。
关键是能给码农们多少时间去写好任务代码。一直听说一天的工作量是8行代码，但从来没有过这样的待遇。


19楼 [baibuds](#) 2012-09-19 16:08发表

 学习了。


18楼 [baibuds](#) 2012-09-19 16:07发表

 学习了^^


17楼 [迷之雾](#) 2012-09-19 13:22发表

 现在在做项目，有时重构不是你个人的问题，还牵扯到时间，精力，测试等问题。
当然我是提倡重构的。


16楼 [romandion](#) 2012-09-18 13:23发表

 我要求每个团队的成员，每天都要提交他的代码。然后每天早上或者下班后，看下代码 变更情况。基本上都能很快发现问题。代码看多了，基本上，每个成员的缺陷，和当时的问题还是能较早的发现。同时还可以针对性加强技术和业务培训。


Re: [wmnothing](#) 2012-09-19 12:25发表

 回复romandion： 嗯，这和团队的整体风格有关系。一旦做到了这一点，事情就简单了。

15楼 [fire4cwj](#) 2012-09-18 11:21发表

 两天时间去重构，可能不够！

14楼 [liangtu](#) 2012-09-17 16:06发表

 引用“goof”的评论：
[quote=liangtu33]加强沟通吧。一个成熟的团队，不应该在这些问题上，认识偏差这么大。 [...]

这个也是个习惯问题，很多技术性（对技术执着、不爱交流的人）的人，都喜欢自己独立解决问题。我刚开始也这样，经常耽误进度，后来才改过来。

13楼 [Tim_包](#) 2012-09-17 15:58发表



引用"liangtu33"的评论：
 加强沟通吧。一个成熟的团队，不应该在这些问题上，认识偏差这么大。

在团队成员与领导的沟通中，应该有个定期回报机制，防止出现到最后半天或者进行不下去的情况下才回报工作。

12楼 Damon_wenhao 2012-09-17 15:24发表



我觉得多方面入手吧。这些东西不是一个人，或者一时半会能搞定的。

Re: wmnothing 2012-09-17 16:23发表



回复Damon_wenhao：我经历过不同的团队，发现差别不小，但问题的共性也很多。我一下子想不出统一的方案。我相信这个应该是各有不同的，也希望可以听到更具体的故事。

11楼 liangtu 2012-09-17 14:11发表



加强沟通吧。一个成熟的团队，不应该在这些问题上，认识偏差这么大。

10楼 beme 2012-09-17 12:31发表



团队的培训和code review都应该定期进行，哪怕再忙，也需要，俗话磨刀不误砍柴工

9楼 ylgm44 2012-09-17 10:37发表



楼主看来与成员关系没搞好啊。。。

8楼 子达如何 2012-09-16 21:24发表



重构需要老板支持，当你花了两天时间去重构一组代码，然后告诉老板我在优化代码，他肯定很不能理解；

7楼 LogicTeamLeader 2012-09-16 09:46发表



进行重构的人员要三个条件：一是能力，二是驱动力，三是时间。
能力很多人有，驱动力只有为该项目负责的人有，时间就看公司如何使用这个人。
所以，最后这件事一般落到项目负责人身上。

6楼 jAmEs_ 2012-09-16 09:00发表



引用"zengzhengliang"的评论：
 重构的成本是比较大的，客户应该不会给你多少机会..
 还是在日常工作中，把code review做好点...

赞同，可惜程序员的能力和态度参差不齐，很难搞

Re: nscboy 2014-01-21 20:28发表



回复jAmEs_：客户关心的是疗效.如果真有效.我想大部分客户还是不难说话的.客户也是人.你投入了精力,还有效果.客户还有啥不满意的呢.
当然,找借口拖延工期的就算了.投机取巧不是对待客户的态度.

5楼 zz962 2012-09-15 11:31发表



前面有几位说得很到位
1、文中所说的不是重构的该做的，而是code review该做的
2、严格说没必要亲力亲为，应该培养骨干、建立机制来贯彻。

4楼 huoyingfans 2012-09-13 09:51发表



学习了。

3楼 liguonan110 2012-09-12 15:00发表



!!!! ¥ R T

2楼 sayume 2012-09-12 14:51发表



楼主把问题已经分析得很清楚了，只是缺乏相应的机制去执行而已。
放权是一种很好的管理方式，可以最大限度发挥部下的创造性，但是放权的前提是做好机制和制度建设。
我可以给楼主提两个意见：
1. 建立里程碑，在项目完成到一定程度时进行定期code review。
2. 培养部下重构能力和习惯。找一些骨干重点训练，能后让他们扩散技术，形成整个团队的习惯。

Re: 争光 2012-09-17 15:27发表

关闭



回复sayume：赞同。

1楼 [zengzhengliang](#) 2012-09-12 10:41发表



重构的成本是比较大的，客户应该不会给你多少机会..
还是在日常工作中，把code review做好点，及早的发现问题，防患于未然。代码写的不规范的，早点重写。
到后期如果真的要重构，那就是没办法继续修改了。

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker Op
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP j
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS I
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack FIC
coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

关闭

