Bug 管理的经验和实践（下）

孟岩：刘振飞，你好。在这个系列的前面两篇文章里，我们先是探讨了 Bug 管理的理念和意义，然后又从软件系统的构建角度更进一步探讨了 Bug 管理技术层面的问题。这次我想我们应该来探讨 Bug 管理中　人　的问题了。当然，所谓人的问题，就是管理制度的问题。有了先进的理念、坚实的软硬件基础，还需要有相应的管理制度与之相配套，否则 Bug 管理就只是一个摆设。你认为一个软件开发团队应当制定严格的 Bug 管理制度吗？没有一个相配套的管理制度，会有怎样的后果？

刘振飞：我们在第一篇文章中讨论过，微软的软件研发可以总结为以下两点：
（1）．需求（PM）、开发（Dev）、测试（Test）三权分立，分工明确、各司其职
（2）．每个产品的每个版本遵循同样的模式：流程＋工具＋人，并不断反馈（以改进流程、升级工具并提高团队/员工的能力）

回到你这个问题，Bug 管理制度其实就是定义 Bug 处理流程，有了好用的工具之后，我们需要这样的流程去明确指导如何对 Bug 进行管理。但是一个软件开发团队应当制定严格的 Bug 管理制度吗？坦率的讲，不需要。严格的制度在软件行业就意味着教条、负担和不切实际，让一帮聪明的大脑陷入无边无际的条条框框不能自拔，明知道是包袱还要去背、是火坑还要去跳，直到有一天终于受不了，最终结果不外乎三个：过劳累到、对付一天是一天或者干脆辞职换个工作。因此我觉得应该用　Bug 管理指导原则（guidance）　来替换　Bug 管理规章制度（rules & regulations）　这个词。

所以我认为 Bug 管理就是去制定适合自己团队实际情况的 **Bug** 处理流程和指导原则，制定者（管理层）应该起到真正指导的作用，他们要非常清楚下面这些问题的答案：

- 我们需要测试什么：哪些软件（网站）、哪些模块
- 测试人员的分工：什么人负责什么模块
- 测试工具和环境：巧妇难为无米之炊。你不能安排一个测试人员去测某个模块，而没有给他提供必要的软硬件环境
- 测试的进度安排：干这一行加班是不可避免的，但是需要有度，人不是机器，长期的劳累谁都扛不住。如果时间很紧，那只能去抓重点，要有所不为
- 发现一个问题时，如何用 Bug 管理工具去创建一个 Bug：标题如何写、严重程度、详细重现步骤、错误状况、期望结果、现场附件、这个 Bug 去分配给谁
- 当一个 Bug 被处理掉时，测试人员应该如何验证并关闭
- 当一个 Bug 的解决方法有争议时，谁来仲裁
- 定期的 Bug 提醒，比如当前每个人的 Bug 情况
- Bug 状态报告：Bug 数目的变化趋势及我们应该采取的行动
- 阶段性的总结反馈：哪些地方我们做的好，哪些做的不好，为什么、如何改进
- … …

没有这样配套的 Bug 处理流程和指导原则，再好的工具都将会是一个摆设、甚至是添乱的工具。就像一个乐队有非常出色的各种乐器，但乐队指挥是个外行（就像成龙电影《双龙会》一个镜头），那么演奏出来的一定会是混乱的乐章。

孟岩：根据你的了解，国内中小型软件开发企业中 Bug 管理制度方面有什么缺陷？主要的问题是什么？

刘振飞：我想目前中小软件企业的 Bug 管理主要存在的问题是：

1. 不重视测试，认为测试人员无关大局，随便测测就行了。当然这种情况正在逐步好转，因为大家都开始意识到了测试重要性；
2. 有些企业，认识到测试的重要性后，却走向极端 --- 制定了极其严格的规章制度：无数琐碎、难用的测试工单、非常严密的一级级权力控制，在 Bug 管理系统中谁能看到什么信息、谁可以解决、谁可以关闭等等，非常严格。一个需要灵活变化的工作变成了工业制造车间流水线的一个工种，让测试人员陷入制度的泥潭，不能把主要精力投入测试工作本身；
3. 管理层自身没有制订明确的 Bug 处理流程及相关指导原则，让测试人员在黑暗中摸索，走到哪儿算哪儿，不能给他们以切实有效的指导和帮助；
4. 管理层把软件的质量保证责任一股脑推到测试人员身上，任何问题都去指责下面的测试人员，殊不知测试仅仅是研发的一个环节，前面需求、开发两个环节如果没有好好做，测试将会极其被动，比如：没有需求文档，怎么测试？这是一个系统工程；
5. 错误的考核标准：管理层用 Bug 个数去衡量测试人员的工作成绩，谁发现的 Bug 多谁的工作就做的好。这是一个十分危险的倾向，将直接导致测试人员去重视 Bug 个数这个数字本身、而不是产品的真正质量。

   遗憾的是，即使在微软内部，各个地方研发中心也有这个倾向，比如经常出现大陆、台湾、韩国、日本四个地方某个软件的测试人员虎视眈眈的在半夜盯着某个版本的问世，一旦下载到最新的 Build，马上安装测试，把表面上的 Bug 赶快 抢 到、记录进 Raid/Product Studio 中，然后心满意足的打车回去，很高兴比另外三个对手多上了几个 Bug。我记得微软内部有个专门的培训曾认真的研讨过这个问题：不能用 Bug 数目来衡量 Tester 的工作。但是微软太大了，当某地方或部门不能找到更合适的标准的时候，Bug 数目本身就是最快捷的答案了。

   这是我现在经常思考的问题之一。

孟岩：能否请你比较系统地阐述一下微软的 Bug 管理制度？

刘振飞：其实前两篇文章已经陆续谈过微软的 Bug 管理指导原则了，这里系统的总结一下：

- 管理层要求所有的 Bug 都要通过 Raid（Product Studio）来跟踪处理。这个看起来很简单的 Bug 管理工具是每个员工和其他同事有效协作的重要保证
- 每个产品都细分模块（Area，SubArea），每个模块都有明确的需求定义者（PM）、开发工程师（Dev）和测试工程师（Tester）这三个角色。一个问题出现了，一定会落实到某个人的头上去跟踪处理，绝不能出现 无主 的 Bug
- PM 负责书写的 Spec 是这个功能特征（Feature）的 合同 ，以此 Spec 来指导开发和测试。当 Dev 和 Tester 就某个 Bug 发生争执的时候，PM 负责给出一个明确的说明
- 测试不仅仅是 Tester 的事情，尽管那是他们的专职工作。研发团队中的所有人每发现产品的问题时候，都有义务把这个问题告知负责这个模块的测试人员去记录跟踪这个 Bug，或者干脆自己新建一个 Bug 来跟踪
- 你可以创建一个 Bug 指派给自己，以跟踪某件事的处理。比如开发人员把源代码中的某处问题用 Bug 记录下来，以后抽出时间来进行处理
- 团队中的所有人都可以创建、指派和更改 Bug 的状态

- 当你创建一个 Bug 的时候，描述一定要足够详细，让下面处理 Bug 的人和其他关心这个 Bug 的同事能够通过 Bug 描述准确的重现这个问题，而不是猜测某些步骤或者跑过来当面问你
- 通常一个 Bug 的处理过程是这样的：
  1. Tester 发现一个问题，到 Raid 中创建一个 Bug，描述这个 Bug 的详细信息，比如重现步骤（Repro Step）、错误结果（Result）、期望的改动（Expect）、运行版本等；然后把这个 Bug 指派给负责该模块的 Dev Lead
  2. Dev Lead 判断后把这个 Bug 指派给某个特定的 Dev
  3. Dev 处理掉这个 Bug 并返还给原 Tester，或者请求 PM 给出一个澄清说明
- 管理层通过 Raid 来跟踪整体进度，以及每个员工、团队在其中的贡献
- 有专人定期给相关同事发出 Bug 的状态报告
- 每个人都可以方便的自助查询 Bug 的分布处理情况。Bug 管理系统对所有的团队成员都是毫无保留的敞开大门（除了你不能删除 Bug，另外所有的操作都被忠实的纪录下来）
- 随着时间的推移，管理层要逐步给出明确的 Bug 解决指导方针：哪些 Bug 是可以不理睬的（Won't Fix），哪些是可以推迟到下个版本处理（Postponed）。比如在最终 Build 到来前的几周，也许非常严重的 Bug，像数据丢失、程序崩溃之类的也都要推迟到下个版本再解决了。
- 当一线员工出现争执、无法达成一致意见的时候（尽管这种情况不多见），管理层要快速给出处理意见

等等。

孟岩：在微软，如果违反了这些制度，会有什么后果？

刘振飞：哈，这个问题有意思。我还真没有仔细考虑过，如果一个研发人员在微软违反了这些 Bug 管理制度 ，会有什么结果？他一定不会因此被开除J ，不过他肯定会努力学习和适应这些 Bug 处理原则，他的直接上司也会指导他如何做才是正确的。

让我们换个角度去考虑这个问题。Bug 管理是研发管理的有机组成部分，而研发管理是微软企业管理极其重要的部分，只有好的企业管理才能把业务做好，业务做好了，公司就有好的利润，这样公司发展了员工也跟着赚钱了。微软可以很奢侈的在众多求职者中招聘到合适的员工，这个员工进来后不可能对微软近 30 年研发总结出来的 Bug 管理制度 发生抵触情绪、甚至有意去违反破坏这些处理原则，他所能够做的只能是快速去体会、理解、适应这些流程和指导原则。

我曾听到这样一个故事：国内某大型软件企业研发老总访问微软，询问如何进行研发管理，微软一位研发高层答曰：很简单啊，定期看看 Email 发来的 Bug 报告和曲线图，然后通过 Email 告诉各软件负责人，下一步应该注意哪些问题就可以了。我们国企老总很愕然，百思不得其解。如果没有相关的软件研发流程和指导原则、配套工具以及熟悉这些的员工，管理层无论如何达不到这样的 轻松自在 --- 用 Email 进行研发管理。

有时候想，我们需要 拿来主义 。与其羡慕 Bill Gates 的钱袋、痛骂微软帝国的 霸权 ，还不如好好研究学习人家的研发管理和企业管理：如何把几万个聪明的脑壳有效的管理起来？如何让分布全球的几千名研发人员每隔上 18 个月生产出新版本的 Office 和 Windows？为什么我们上百人、几十人甚至几个人的软件企业就管理不好？

孟岩：在整个 Bug 管理的系统中，测试人员是非常关键的一个角色。以前测试人员在团队里的形象好像是灰色的，这两年各公司都开始重视测试工作和专业测试人员的培养。你觉得

测试人员在 Bug 管理体系中处在一个怎样的位置上？测试人员与开发人员之间的关系如何？

刘振飞：测试人员在整个软件研发管理体系中都是一个十分重要的、无法替换与省略的角色。经过多年产品研发的体会，我现在无法想象一个软件企业没有或者不重视测试和测试人员。就像没有经过质检人员检验过的流水线生产出来的电视机，能出厂吗？会有人买吗？

测试人员和开发人员是对立统一的关系。说对立，是因为测试人员需要专门挑出开发人员做出来的功能模块的毛病、发现其考虑不周的地方；说统一，这两个角色需要努力协同工作，把负责的模块做好。只有每一个模块问题减少了，整个产品才能提高质量；好质量才有好价钱；公司赚到钱了大家才能有好收入。所以开发和测试是同一战壕里的战友，只有共同努力才行。

孟岩：现在很多开发工程方法提倡 全员测试 ，很有点类似日本企业里流行的 全员质量管理 。特别是单元测试已经将功能性测试变成开发人员的一项职责，这是否与 Bug 管理体系有冲突？全员测试是否会导致责任的不清晰？你觉得单元测试与 Bug 管理是否矛盾？能否协同工作？

刘振飞：一点也不矛盾。开发人员把一个功能模块送去测试的时候，应该已经把最基本、最常用的功能逻辑测试通过，否则测试人员发现这些基本问题后，很快还得退回去给开发人员，这样双方都费时费神。

我所理解的 全员测试 就是每个人当发现问题的时候，不能说 这是测试/研发人员的事情 而置之不理，而应该把这个问题记录到 Bug 管理工具中，或者告诉相关的测试人员去跟踪。产品是公司的产品，是大家共同的饭碗。当然测试人员的专职工作就是去分模块测试，而且测试得有计划、有条理、有总结归纳；别的同事可能不是那么系统化而已。

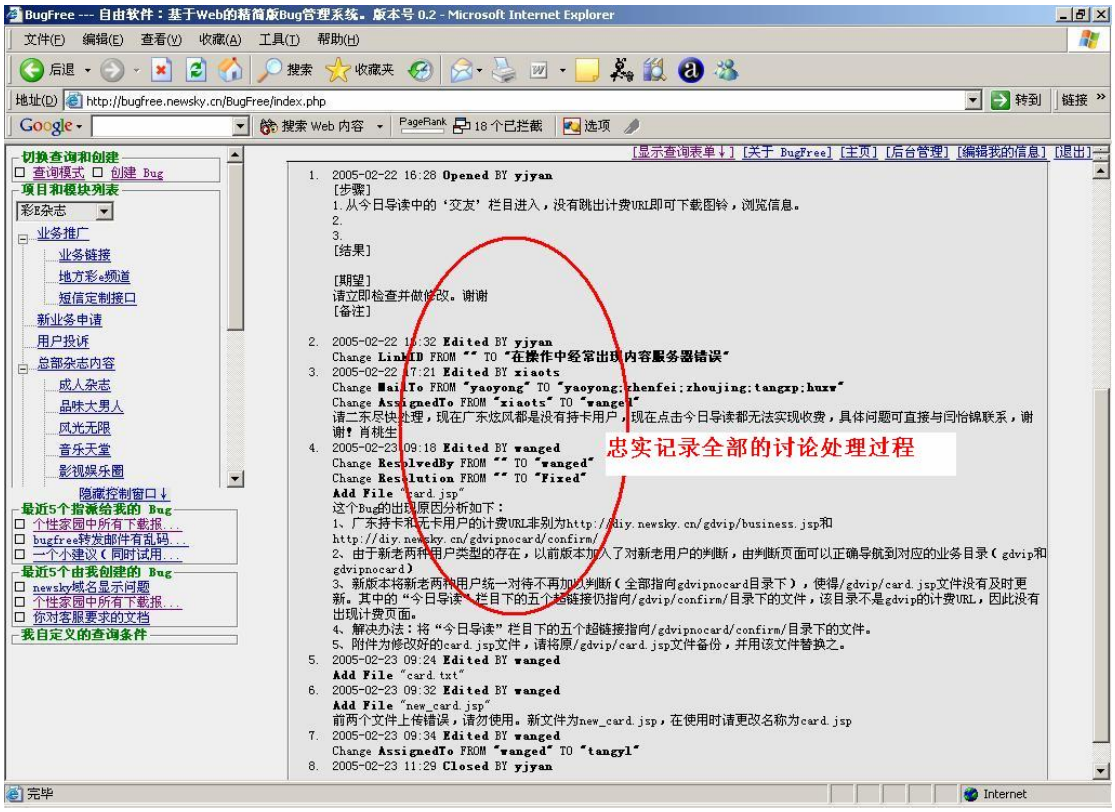
在 Office 2003（Office 11）发布后启动的下一版 Office 12 研发伊始，微软 Office 组的管理层就根据大家的反馈，启动了一项叫做 Engineering Excellence 的活动，全面总结上一版研发流程中经验教训，提出了十多条大的、具体的过程改进办法全面执行，其中有一条叫做 Feature Crews ，就是加强测试：在把源代码 check in 到代码库之前，就开始测试一个功能特征（Feature）。该 Feature 对应的 PM、Dev、Tester 紧密合作在本地 Build 上，当一个 Feature 进入总产品代码库的时候应该经过认真测试、非常稳定可靠，就是说把测试工作大大往前（开发阶段）提了。当然 Office 组有专人立即设计、开发相关的支撑工具去保证 Feature Crews 这个新方法能够顺利执行。当我第一次看到这份 Engineering Excellence 活动说明时，真是佩服得五体投地！！很多像这样具有优秀管理、执行能力的各个小组织，组成了微软公司优秀的大团队。

孟岩：这样吧，假设你领到一个团队进行软件开发，以 BugFree 为基础进行 Bug 管理，能否简单地介绍一下你打算制定怎样的 Bug 管理协作制度？比如，有哪几个角色，角色之间如何协作，那些规定需要作为硬性要求保证执行，如何保证等等。

刘振飞：我现在还真是正在带领着一个团队去这么干（北京金环天朗通信技术发展有限公司 http://www.newsky.cn）。我所计划的 Bug 管理指导原则是：

ü 产品（WAP、彩 e 或彩信杂志、网站等）中碰到的所有问题都要用 BugFree 来跟踪处理

- ü 有一个专职的测试小组
- ü 团队中每个同事发现一个问题时，都有义务去告知相关的人员或者直接创建一个 Bug
- ü 需求、开发、测试三个角色的定位要非常明确。特别的，提出需求的人要把需求认真考虑好、细化成文档，然后才能提交正式开发、测试
- ü 发现一个 Bug 时，测试人员提交给某个开发小组长，他来负责指派给具体的开发人员；产生争议的时候由需求定义者来出面说明； 矛盾 很大时我来协调和仲裁。Bug 的处理过程都要用 BugFree 记录下来：



ü 每天系统自动通知头上有 Bug 的人自己还有几个问题。我会检查这些 Bug，看到不合适的地方就去添加我的意见

ü 每周系统自动通知所有人前一阶段 Bug 的整体情况；同时测试小组要汇总上周的 Bug 测试情况，告诉团队中所有同事哪些模块容易出问题、主要有哪些类型的问题

上面这些我能够作为 硬性要求 的，只能是前两条：

ø 任何人再向开发人员反映问题的时候，开发人员会告诉他们：创建一个 Bug 来跟踪

ø 刚刚成了一个测试小组

其余的只能融化在日常工作中，管理层不断在很多细节上要求、甚至亲自示范（比如我会使用不同的产品，发现问题上 Bug），去教会大家测什么、如何测、发现问题怎么办、Bug 解决后怎么办。

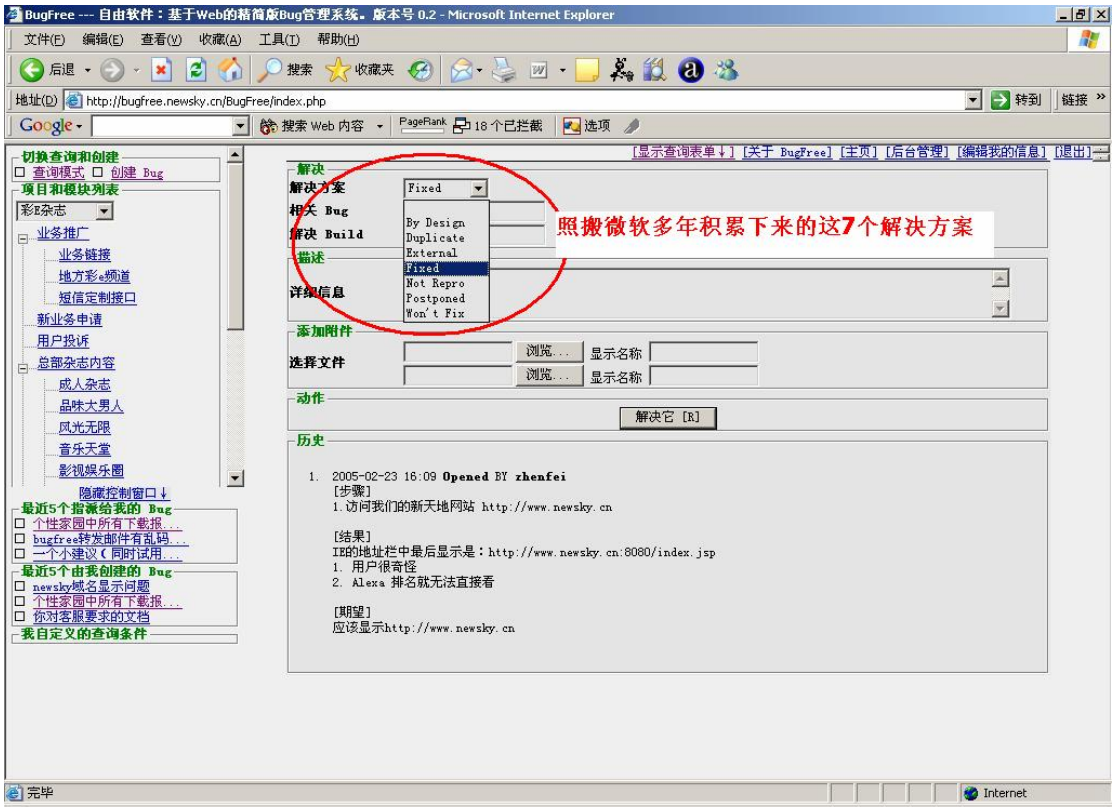
因为整个研发团队刚招聘了不少新人，由于历史的原因以前也没有重视测试工作，大家这方面的经验相对而言比较少，所以目前我最重要的工作是给大家不断灌输这些意识，手把手去教他们如何创建一个 Bug、解决一个 Bug 时应该怎么描述、提供哪些信息等等。坦率的讲，这个过程会非常辛苦劳累。去年我在的公司比较小，一切我都可以从零开始设计规划。但现在不同，因为公司业务有很多、人员也有了一定规模、以前的程序有很多，而且基于这个行业自身的特点，新的需求很多、变化非常频繁，所以在这种情况下如何把研发流程理顺、Bug 管理到位，对我也是很大的挑战。我会根据对业务的深入理解去不断调整、细化上面提到这些 制度 。

孟岩：BugFree 在设计上为此作了什么特别的考虑？

刘振飞：我想主要有这么几点：

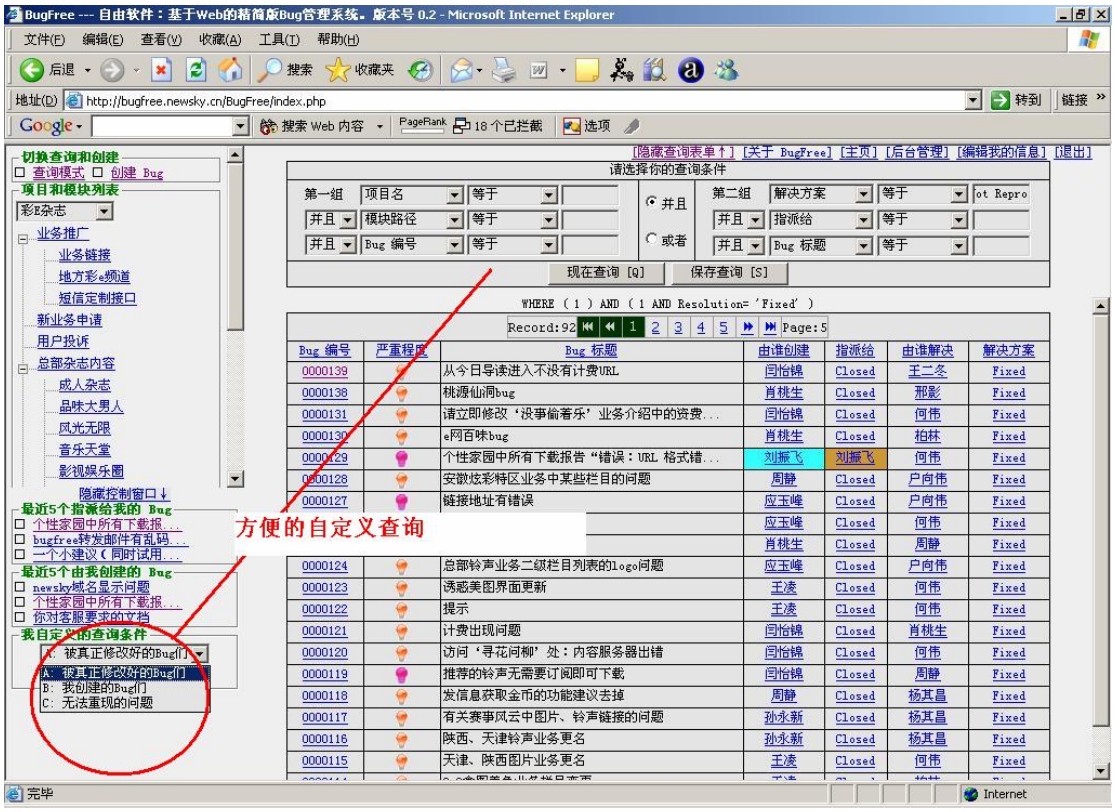
（1） BugFree 是基于 Web 的 Bug 管理系统，我们研发人员很容易上手使用、简单方便

（2） 一个 Bug 从创建到关闭这个 生命周期 的处理过程，BugFree 全面借鉴微软内部工具 Raid 的处理流程，处理方法甚至一些词汇都和 Raid 一样，代表着 先进的生产力 J



（3） 当一个 Bug 被指派给你的时候，系统会自动给你发一封邮件，提示有个 Bug 需要你处理，这样结合 Email，不断提醒研发队伍 Bug 的存在和进展。我们还增加了两个 Bug 统计功能：一是每天定时（比如早上 8 点钟）每个同事都会收到一封 Email，告诉他/她头上还有多少 Bug 等待处理；二是每周一中午给所有人发一封邮件，告知上周 Bug 的处理情况和到目前为止所有 Bug 的统计数据

（4） 很方便的去自定义查询条件，以后轻轻点击一个按钮随时查看你关心的 Bug



（5） BugFree 是用开源的 PHP+MySQL 写成的，规模很小、代码也很规范，所以需要的时候很容易定制或扩充功能

孟岩：非常感谢你，刘振飞。我们通过这三期访谈，已经较全面地谈到了 Bug 管理的方方面面。从前两次的反应来看，很多读者都对这个话题非常感兴趣，你能否用三句话总结一下你的主要观点？

刘振飞：

1. 测试是软件产品研发的重要一环，需要 IT 企业的高度重视，就像重视开发一样
2. 选择一个得心应手的 Bug 管理工具,比如使用最接近微软内部 Bug 管理系统的开源软件 BugFree ，是免费的！J
3. 明确 Bug 管理的流程和指导原则，并把这些意识逐步灌输到每个研发人员头脑中；同时根据企业的具体情况不断去完善测试流程和方法

也真诚感谢你做这次访谈，通过这么多有启发性、很有条理的问题，我算是有机会把这么多年的软件研发经验、特别是 Bug 管理的体会系统的总结一下，给自己留下一份很有意义的记录。同时借这个机会，也感谢给我 Email 探讨 Bug 管理实践以及 BugFree 系统的读者朋友表示感谢，如果这三篇访谈能对大家的实际测试管理工作有所帮助、BugFree 能够被真正使用起来，那我就非常自豪和快乐了。

另外，我也会根据自己的使用经验和网友们的反馈，逐步完善 BugFree，让它成为一个长期的、有生命力的开源项目。

孟岩：最后一个题外话。我知道你现在从事研发管理工作，招聘过不少技术人员，对那些未走出校门的大学生和刚刚踏入社会的大学生们，有什么意见和建议？

刘振飞：春节后我刚刚完成我们部门今年的第一轮招聘，一共收到了 1000 多封简历、面谈过近 100 人，最后录取了 9 名新同事。去年也曾招聘过一些新人；一方面是很多大学生工作不好找、另一方面是很多企业招到一个合适的人也很费劲。在招聘过程中，真是什么情况都碰到过。

作为一名有几年工作经验的老毕业生，我想对年轻的学弟学妹们提几点建议以共勉：

1. 争取做一个善良的人、多站在别人的角度上去考虑问题。
2. 要树立为自己努力工作的心态，你不仅仅是为老板打工。如果对工作不满，赶快换一个，千万不要耗着；我们比上一代人幸运、可以有更多的选择工作机会，所以不要浪费自己的青春年华。
3. 珍惜在学校读书的四年宝贵时光，打好专业基础（比如计算机专业的起码应该把离散数学、数据结构认真搞明白）、提高基本素质（比如诚信、沟通表达及团队协作能力）。
4. 如果你想朝技术方面发展，多去钻研钻研那些优秀的开源软件，学习别人的智慧；
5. 摆脱那种非常纯的技术情结，要逐步明白在市场经济的企业中，业务、管理最重要，技术是一种　后勤支持　，没有我们想象的那么重要。
6. 不断学习来提高综合素质。技术之外的书籍：哲学、历史、经济、文学等都需要好好读一读。古人云，　世事练达即文章，处处留心皆学问　。比如我看电影和话剧的时候，觉得这两样和软件研发非常相似，剧本就是需求、演员就是开发人员、彩排类似测试，导演呢就像一个项目经理，一个票房很高的电影就像很受欢迎的软件产品一样。
7. 身体是革命的本钱。毛主席他老人家的这句话千真万确，健康的身体是扛住工作和生活压力的重要保证。
8. 想办法去慢慢培养自己金钱和管理意识，碰到合适的机会也可以尝试自己创业，即使失败了也可以学到很多书本之外的知识。难道陈天桥生下来就注定要当中国大富豪吗？王侯将相，宁有种乎？J

顺便提一句：热烈欢迎感兴趣的同仁加入我们公司！我们一起经历这充满挑战性的过程！

孟岩：好的，我们关于 Bug 管理这个话题的讨论就告一段落。感兴趣的读者可以通过 liuzf@pku.org.cn 直接与刘振飞本人联系探讨，也可通过 http://www.okooo.com/OpenSource 去体验、试用并下载 BugFree。