

BugFree is free software under the terms of the FreeBSD License.

自由软件 BugFree 官方网站是 <http://www.bugfree.org.cn/>。

我们将保持自由的心态去不断完善这款自由软件。BugFree 是一款自由软件，为了保持其纯洁性，经过讨论，我们严肃的承诺：BugFree 官方网站(<http://www.bugfree.org.cn/>)所属网页(含论坛)，以及官方网站发布的 BugFree 版本，将拒绝放置或携带任何商业广告 --- 以保证 BugFree 的用户看到的是一个单纯、干净的自由软件及其讨论社区。

工具是思想的载体，[《Bug 管理的经验和实践》\(Thinking in BugFree\)](#) 就是 BugFree 所蕴含的软件研发流程思想的全面总结。

自由软件 BugFree 简介 --- 借鉴微软公司软件研发理念的 Bug 管理系统

刘振飞 2004-10-11

1. BugFree 简介

1.1 BugFree 的来源

BugFree 是借鉴微软的研发流程和 Bug 管理理念，使用 PHP+MySQL 独立写出的一个 Bug 管理系统。简单实用、免费并且开放源代码(遵循 FreeBSD License)。

如何有效地管理软件产品中的 Bug，是每一家软件企业必须面临的问题。遗憾的是很多软件企业还是停留在作坊式的研发模式中，其研发流程、研发工具、人员管理不尽人意，无法有效的保证质量、控制进度，并使产品可持续发展。

针对这个问题，我们独立做出了 BugFree，并且半年多来每天都在使用。我们公司就是用它来管理 Bug，不断提高产品质量的:-)

1.2 BugFree 名称的含义

命名 BugFree 有两层意思：一是希望软件中的缺陷越来越少直到没有，Free 嘛；二是表示它是免费且开放源代码的，大家可以自由使用传播。

1.3 为什么开放 BugFree 的源代码呢？

根据半年多的实践，觉得 BugFree 非常有用，我们公司的日常工作已经离不开它了。虽然没有微软的 Bug 管理系统(以前叫 Raid，现在是 Product Studio)的功能那么强大，但是处理方法和思想是完全一致的，起码我自己用起来的感觉和在微软时基本一样，值得向大家推荐。

我们是用开放源代码的 PHP+MySQL 开发的，目的就是希望跟大家分享 BugFree。而且开放源代码之后，期待高手不断改进它，大家都能用到更加强大的功能。也算为中国的软件业做点小小的贡献:-)

BugFree 代码在我们的“数字神经系统”中非常独立，很容易拿出来给大家共享。

1.4 BugFree 仅仅是个工具

不过坦率的讲，BugFree 仅仅是个工具而已，重要的是掌握其中蕴含的软件研发的流程思想，才能用好这个工具。如果你以前没有用过 Bug 管理系统，那么一开始的时候也许你会觉得这个工具是在浪费时间，因为一个测试人员需要费神把发现 Bug 的详细步骤记录下来，有时还要贴一张示意图，这一切都不如当面说来得直接。

但是使用一段时间，你会发现 BugFree 很有用，它忠实的记录着每个问题的处理过程，不

断提醒你存在的问题，永远不会丢失和忘记。如果你参与过较大软件项目或产品的研发，就会理解它对软件可持续发展是至关重要的。而且研发的规模越大，BugFree 的作用就会越大。

2. 如何安装使用 BugFree

BugFree 的安装和使用都非常简单，请参考 <http://www.bugfree.org.cn/help/>。

希望你能喜欢 BugFree，就像我一样。

3. 为什么开发 BugFree

一言以蔽之，「工欲善其事，必先利其器」。

3.1 我在方正的困惑

我曾在北京大学计算机研究所(方正技术研究院)学习、工作了七年，那里有我很多师长和好友，我参与研发的方正 RIP (PSPNT) 至今仍是方正的核心软件产品。当自认为成了一名合格的 C++ 开发人员后，我发现面临一个无法解决的难题：如何有效地控制软件研发流程以保证产品质量和进度。我意识到做好一个软件，只靠技术好是很不够的，必须要有一套好的研发流程和配套的研发工具。恰好那时我在吴士宏的《逆风飞鹰》中读到了微软的“数字神经系统”及相关叙述，很吃惊，想去那里看看。

3.2 微软数字神经系统的妙处

2000 年我加入微软(中国)研发中心，真的很快在这里找到了答案。四年来，我所理解的微软数字神经系统在软件研发中的妙处是：

3.2.1 研发人员分工明确

主要的三个角色：PM (Program Manager)、Dev (Developer)、Tester 三者分工明确、接口清晰，PM 来定义需求、书写出来每个功能特性 (Feature) 的设计文档 (Spec)，Dev 写代码来实现这个 Spec，Tester 来测试 Dev 做出来的东西是否符合 PM 定义的 Spec，三个角色之间并无必然的上下级关系，只是分工合作完成某个功能 (Feature)。我将之形容为“三权分立”，三者之间有效合作并制衡。国内企业好像还没有 PM 这个角色，而测试人员又往往成为开发人员的附庸，一个 Bug 是否要被解决全由开发人员说了算，这很糟糕，就像政治上一个权力没有被有效的制衡一样，一定会产生各种问题。

3.2.2 研发工具很配套

PM 将写好的需求设计文档 (Spec) 保存到 SharePoint 【见附录一】文档库中，所有相关的人都可以随时查看；Dev 用 Source Depot (功能类似 CVS 的微软内部源代码管理工具) 来保存源程序；Tester 把发现的 Bug 记录到 Raid 中以有效跟踪这个问题的处理流程。

3.2.3 分阶段的研发流程

和任何软件公司一样，微软的研发无非也分为规划、开发、测试、发布等几个阶段。但是微软的研发流程不走形式，可以统一产品组所有员工的思想，并且能够有效地控制住进度。做完一个版本后，还会让所有员工匿名投票，找出这次研发过程中出现的各种问题以便在下个版本中解决 (此过程称为 Postmortem，挺吓人的一个词)。

可以这么比喻，微软这套研发模式是让其中的每个人都成了一架高速运转的机器上的各种零件，少数零件坏了不要紧，可以随时更换。当然微软有许许多多技术高手，但我认为更重要是其研发模式保证了软件产品的高品质、可持续发展。

3.3 微软的每一天

我在微软经历了三个产品的研发：**Office XP**、**Project 2002**、**Office 2003**。微软所有产品的研发都遵循同样的研发模式、使用同样的研发工具来进行管理。在所有的工具中，我最佩服的就是 **Raid**。可以说，遍布全球的微软研发人员能够保持统一的思维模式、做事及语言习惯，与整个研发流程的配套工具密不可分，其中最重要的就是通过 **Raid** 把整个产品的研发有机的联系起来。阅读每个 **Bug**，你可以详细的看到大家讨论解决该问题的完整思路。

我曾读过微软 **Project 2002** 产品的 **Architect** 写的一个备忘录，其中提到“**Raid** 是微软最伟大的产品，为 **Bill Gates** 节省了无数金钱”，当时我“心有戚戚焉”，立即给这哥儿们发 **Email** 表示赞同之意:-) 他回信说希望 **Project** 能够做的像 **Raid** 一样成功，但可惜他要离开微软自己开公司了。

在微软上班，我每天第一件事是打开 **Outlook** 来处理有关自己的重要邮件，第二件事就是打开 **Raid** 来看看有关自己的 **Bug** 情况，赶快处理。我一直纳闷，微软为什么不把这个 **Bug** 管理系统作为软件来出售，那可是任何一家软件企业都需要的啊！

3.4 在朋友公司的实践

2004年4月我加盟朋友的公司做网站，发觉自己已经习惯了微软的研发模式，于是建议这几个朋友先做一个“数字神经系统”(BugFree 就是其中有关软件研发的 **Bug** 管理部分)，其目的是让一切可以数字化、文档化的信息被记录下来，为公司的进一步发展和决策提供基础信息支持。该系统可以用八个字来概括：数据、文档、自助、自动。其表现形式就是一个包括六个子系统的企业内部网：

3.4.1 员工管理系统

每个员工都有唯一的 **UserID**，验证密码后方可登录数字神经系统，访问公司内部信息，查看上下级关系、每个员工的个人公开信息等，此处学习 **SharePoint**、**Outlook** 和 **Exchange** 中的员工管理和展示；

3.4.2 信息管理系统

内部的信息发布展示平台，有点象 **BBS** 一样，可发布公司正式通告、员工也可自由匿名发帖；

3.4.3 Email 系统

现在 **Email** 的重要性对一个企业不言而喻，我们采用免费 **Qmail** 来搭建；

3.4.4 文档管理系统

一个集中管理公司所有文档(包括研发过程书写的各种文档)的地方，学习 **SharePoint** 中的文档库；

3.4.5 源代码管理系统

集成优秀且免费的 **CVS**；

3.4.6 BugFree

虽然网上有免费的 **Bug** 管理系统，但是我看后觉得都不好使，和我在微软用的差别太大，科泰世纪公司的 **Bug** 管理系统【见附录二】倒也很像微软的，但是要花钱买。于是决定用 **PHP+MySQL** 借鉴微软的研发流程和 **Bug** 管理工具自己开发一个，以便对我们开发新网站、声讯软件、客户端软件和公司事务管理中出现的问题进行有效的跟踪处理。

当然这个小小的“数字神经系统”还不全面，而且当时设计的时候更多的是站在为方便研发人员的日常工作而考虑的，以后还需要挂靠更多的公司业务相关模块。即使这

样，它已经发挥了很大威力，现在公司每个员工都养成这样的工作习惯并乐在其中：每天使用邮件，到文档库中查找文档，发现问题上 Bug 等等。

3.4.5 BugFree 代码的完成

“数字神经系统”中的 BugFree 是用开放源代码的 PHP+MySQL 写成、基于浏览器方式运行的。我以前没有任何 Linux+Apache+MySQL+PHP 的开发经验，但我很幸运的招聘到两名优秀的 Web 程序员，可以在短短的两个月时间内搭建起这样的系统。其中 BugFree 是由我的同事王春生<wwccss@263.net>开发的，他用了不到一个月的时间就把代码写完，让我很是惊讶，从而认识到基于 Linux 的 Web 开发魅力。

之后我们测试一个多月，就可以在实际工作中使用。现在 BugFree 已经成了我们日常工作最重要的工具，每个员工也都习惯用 Bug 来记录跟踪事情，不仅仅是代码中的缺陷可以上 Bug，新的需求、设计变化等都可以用这个 Bug 管理系统有效的管理起来。其实 Bug 不仅仅可以用来记录软件中的缺陷，也可以用来跟踪公司的日常事务。比如在公司网上报销系统还没有建立之前，我们就用 BugFree 来处理报销的事情。甚至，一个同事给我上了这样的 Bug：你的桌面太乱了，请整理一下:-)

3.4.6 和 Raid 比较起来，BugFree 有如下特点：

- Raid 是 Windows 客户端软件，BugFree 是基于浏览器的。基于此，Raid 有很强大的编辑展示功能，而 BugFree 简单、方便、易用；
- Raid 可以进行极其复杂的组合查询，BugFree 的查询功能相对弱一些，但我觉得已经够用了；
- 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 Raid 的处理流程，处理方法甚至一些词汇都和 Raid 一样（所以我现在用 BugFree 处理 Bug 的感觉和在微软时候基本一样）；
- BugFree 还有一个独创的功能：当一个 Bug 被指派给你的时候，系统会自动给你发一封邮件，告诉你有个 Bug 需要你处理，这样结合 Email，BugFree 被完美使用起来，成为我们现在网站开发、运行、维护必备的工具。我们还增加了两个 Bug 统计功能：一是每天早上 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；
- BugFree 程序规模很小，一个中等水平的 PHP 程序员就可以在 1~2 周内看懂所有的代码，然后就可以根据自己的需要做相应的定制了；
- 最最重要是，BugFree 是免费并且开发源代码的。你可以体验到微软的 Bug 管理精髓，按自己需要自由地增加功能、修改代码而不用担心版权问题:-)

4. BugFree 目前的开发情况

和任何软件系统一样，BugFree 中还存在着一些 Bug、还有很多地方可以改进、还可以增加很多新的功能。但是基于人力的关系，没有太多精力去进一步完善它。开放源代码之后，希望有热心人不断完善 BugFree。以下是目前我想到的可以改进的地方：

4.1 宣传推广 BugFree 的使用。

一般中小企业没有很好的 Bug 管理系统，刚好可用 BugFree 来填补这个空白。其实 BugFree 是一个非常简单的工具，可以让非技术人员快速上手使用。开源的目的是让它被广泛使用，只有被大家天天用，我们才能找到下一步不断改进这个软件的真实需求。有人天天用是软件生命力的保证。

4.2 和源代码管理系统 CVS 有效的统一起来。

当代码的改动是因为解决一个 Bug 引起的时候, 系统可以自动把代码的 Check-in 结合起来, 微软的源代码管理 Source Depot 好像早就有了此能力。可以考虑通过 Eclipse 把 CVS 和 BugFree 连通起来。

4.3 搭建研发工具套件。

更进一步, 把 BugFree 和需求设计文档、开发文档、测试文档、测试用例等联系起来, 做成一个软件研发工具套件, 即可通过一个 Bug 方便找到对应的文档、代码、测试用例等等, 那将是一个比较完美的境界。微软现在的 Product Studio 正在向这个方面努力。不过, 达到这一步, 必须先做一个好的文档管理工具, 就像微软的 SharePoint 中的文档库一样, 不仅仅存放文档, 还要方便检索 (比如 Word 文档内容) 等, 那可不是一件容易的工作。我目前初步的想法是, 先借鉴“基于 Java 的全文索引引擎 Lucene”去做一个易理解、易扩展的基于 C++ 的全文索引引擎, 我在微软做过 SPS 的 PM, 对搜索引擎这一块还是比较熟悉、可以做出来的。有了这样的引擎为基础后, 再借鉴 SharePoint 做出文档库及门户功能。

这些工作都需要花很多精力和时间, 而且也很难做。

5. 谈点别的: 关于我的梦想

程序员开发软件的时候, 需要一行行的去书写代码、调试、测试。我总在想, 如果哪一天我们不需要写代码就好了, 但你把需求确定并细化后, 计算机直接把程序生成出来, 那该多省事! 令人兴奋的是, 我见识过这样的系统, 那就是我的好朋友林燕锋历时八年多所开发的『KCOM 商业工程』, 尽管目前这个 KCOM 平台目前只能做到对信息管理系统(MIS)和工作流 (Workflow) 两者应用做到自动代码生成 (请注意不是生成代码框架, 而是根据你的需求设计自动生成所有最终代码, 你不用写一行代码, 也不用测试了, 直接可以使用!)

这个 KCOM (3A 软件网 <http://www.3acesoft.com>) 平台可以这么理解:

1. 从底层核心看, 它有自己的语言 (KCOM Basic) 和虚拟机 (KCOM Stage), 非常像 Java 和 .NET;
2. 从组件技术看, 它有自己的组件机制 KCOM, 类似 COM 和 COBRA
3. 从编程角度看, 它提供类似 Visual Basic 和 Delphi 的编程能力;
4. 从建模工具看, 它提供类似 Rose 的建模能力;
5. 从最热门的软件工程技术看, 它已经在 MIS 和 Workflow 两个领域实现了模型驱动体系架构 (MDA, Model Driven Architecture) 所设想的自动代码生成!

牛吧! 做出以上任何一个都不容易, 而林燕锋却全部实现了!!

我认识林燕锋已经四年多了, 当第一次看到他演示的自动代码生成功能, 实在震惊, 没有想到一个计算机软件开发的理想境界居然在他手里开始实现了, 不是模型, 而是实实在在的可用的软件系统!

尽管现在他的 KCOM 还没有得到市场的广泛认可, 但我始终相信这个产品有巨大的市场价值和社会意义。若能争取到更多的扶持、加大市场开发力度, 将会对加快我国企业、政府、军队的信息化建设具有不可估量的推动作用。

我设想如果使用 KCOM 来做一个类似微软 SharePoint 的企业门户软件, 那将是一个非常有趣的事情。

方正七年、微软四年, 我现在的梦想是:

1. 在不久的将来, 可以有机会跟林燕锋合作, 不断完善 KCOM 软件平台, 扩展自动生成代码

的领域；更进一步，能够创造一个适当的环境，可静下心来实现他最早设想的知识库系统，每个人都可以通过 Internet 来共享自己的知识，共同构建一个互联的知识库系统，就像现在可以通过 Google 来搜索网页一样，以后你可以通过这个知识库系统来搜索共享的或可交易的知识。

2. Internet 促进了整个世界的发展，引发了人们生活和工作方式的重大变革，这仅仅只是开始。我想有机会做做和 Linux/Internet 相关的分布式系统方面的技术，比如分布式系统、电信和 Internet 结合、超大邮件系统等等。开源软件对我们国家而言是个巨大的机会，但愿我这个技术人员能够为此做些有益的工作。

3. 利用我现有的软件研发管理经验和体会，使用我最喜欢的 C++ 语言，带领一支有战斗力的队伍，做一些大的、能够影响很多人对社会有较大贡献的软件，就像我曾经参与的方正 RIP (PSPNT)、微软 Office 一样。

P.S. 我觉得斯皮尔伯格编剧并导演的电影《A.I.》抓住人工智能的核心，人与机器最终的分界线可能就是“爱”。让机器拥有智能应该是不太遥远的事情，但是我现在无法想像如何让机器拥有意识和懂得“爱”。人类的“爱”是这个宇宙最伟大的力量，机器终究也会有人一样智能，但“爱”或许是人类抵御机器的最后一道防线。

6. 附录

6.1 微软的 SharePoint

微软的 Windows、Office 广为人知，但是 Office 家族中的一个企业级产品 SharePoint 大家可能还不很熟悉，它分为两个版本 SharePoint Team Service(现改为 Windows SharePoint Service, 简称 WSS), 是为一个小组建立网站所用；另一个叫 SharePoint Portal Server (简称 SPS), 是为了建立企业门户所用，SPS 基于 WSS, 增加了强大的门户管理功能、文档搜索功能。可以这么说，互联网上的 Web 页搜索你有很多选择，如 Google、Yahoo、百度等等；企业内部网上你好像只有选择 SPS, 尽管这个软件不便宜。

其实判断微软产品的优劣，只要看看微软自己在内部使用哪些产品好了。从 Office XP 发布以来，SharePoint 已经广泛地在微软内部部署使用起来，现在各个部门、小组的网站几乎都建立在 SharePoint 上了。微软也在不断投入力量加强这个产品的研发，考虑到 Office 家族的客户端产品(如 Word、Excel、InfoPath 等)可以无缝连接到 SharePoint 上，非常完美的集成在一起，不夸张的说，SharePoint 又将是微软一棵金灿灿的摇钱树了。现在 Office 家族的产品都在向 SharePoint 挂靠，这样每个独立的看似没有发展空间的客户端产品如 Word、Excel 等，一下子都有新的发展余地，开始迈向真正的企业级应用解决方案。不佩服微软那么多聪明的脑瓜不行啊！

6.2 科泰世纪公司的 Bug 管理系统

据我所知，还有一家公司也在学习微软的研发模式:科泰世纪(www.koretide.com.cn)，他们的软件工程管理工具像我们一样在借鉴微软的经验。科泰的首席科学家陈榕在美国微软工作多年，现在回来做操作系统，动手之前先把研发工具和研发模式学习微软的样子搭建起来。我想凡是在微软做过研发、理解了这套研发模式的开发人员，如果离开微软继续做开发的话，一定会做一个类似的 Bug 管理工具，“英雄所见略同”:-)有意思的是，我们和科泰世纪都用 PHP+MySQL 来开发 Bug 管理系统、用 CVS 来管理源代码，但不同的是他们把 Bug 管理工具当作软件来卖，我们免费并开放源代码了:-)

在 2002 年 10 月底 C++ 泰斗(咱的偶像) Bjarne Stroustrup 来北京访问的那两天，我跑到清华和北大聆听大师的三次讲座。北大的那一场就是陈榕做的翻译，我当时还跟陈榕聊了一会，后来又通过几封 Email，逐步了解了他们公司，也阅读了陈榕的 ezCOM 资料，很佩服他的技术思

路，更佩服他做操作系统的勇气，前两天看报道说上海市政府给科泰 2500 万元投资，而且他们的系统开始在国产 3G 系统试用，很看好科泰的前途，尽管产业化过程会充满艰难。但一旦成功，将有巨大的社会意义和经济收益--- 想想微软当年是怎么发展壮大的吧。祝福陈榕，祝福科泰。