

Design of BUG Tracking System

陈文刚 (东方通信股份有限公司 100085)

摘要 本文介绍了一种基于web方式的bug跟踪系统。通过这套系统,项目管理人员、软件开发人员和软件测试人员可以跟踪每一个被发现的软件错误(bug)及其更改、调整的历史过程;通过这套系统,管理人员可以及时考察相关软件技术人员的工作执行质量,进行数据挖掘,从而不断改进软件开发过程。

关键词: bug 错误跟踪 数据挖掘
B/S 结构

1 前言

对软件开发、调试、测试中出现的错误(bug)建立一套有效的跟踪系统是成熟的软件企

业在软件工程化管理中必然要实施的制度之一。

开发一个大型的软件产品,在软件模块测试、集成测试、系统测试中会发现大量的错误,将这些错误进行统计、归类分析以及进行必要的跟踪对企业的软件管理会起到很好的促进作用。据国外某知名公司自己的统计,他们在研制某大型系统产品的过程中发现了近10万个错误,这些错误记录都是利用数据库来管理和控制的。同样,国内任何软件企业在开发产品的过程中,也会发现大量的错误,这些错误记录在不同的企业有不同的处理方式,这些错误记录可能会存放在测试记录、调试记录、测试报告、数据库、自定义的文件、邮件记录,还有的甚至就放在相关人员的脑子中,这些不同的载体分部存放着软件企业最具价值的错误记录,但是这些记录没有用统一的方法来整理归类,没有被充

分的共享利用,更不会有人系统的去分析这些错误产生的原因和规律。

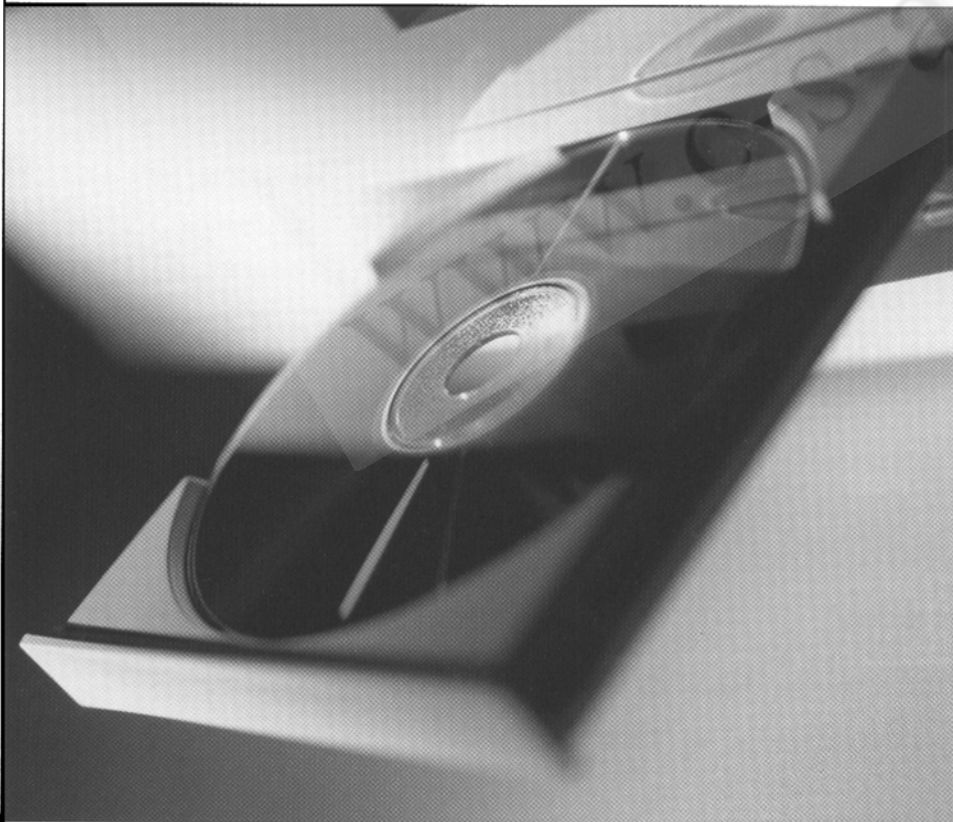
在我们公司的软件开发过程中我们发现,有些错误现象往往是多次出现而测试人员、软件开发人员每次却总得一一应对,如果用数据库的方式将这些错误记录统一存放,利用关键字索引可以快速察看已往的解决记录,用最快的时间找到解决问题的人和办法。另外,测试过程中很多出现的错误由于没有跟踪记录系统,没有得到有效的解决和重视而被人遗忘……这些问题集中起来都可以利用信息化的方式来处理,开发一套基于web方式的错误跟踪软件可以摆脱上面的困境,为加快和规范软件开发过程发挥作用。

2 错误记录字段的构成设计

不同企业对每个错误记录的描述详细程度是不同的,还有即使是同一企业内部不同项目所要求的错误记录的描述也是不尽相同的,作为一个规范的软件企业应该在开发bug跟踪系统之前对本企业作详尽的调查,查阅公司的历史记录,寻找出那些对企业项目管理最有帮助的错误记录字段。

经过我们企业的内部调查,归纳出以下错误记录字段作为一个错误记录的构成。

- bug名称(简单描述bug现象)
- bug提交时间
- bug解决时间
- bug的位置(项目名、文件名、模块名等)
- 软件版本号
- bug严重程度(轻度、一般、严重、致命等)
- bug处理方式
- bug错误类型(拼写错误、逻辑错误、系统错误等)



- bug处理优先级(立即处理、可稍后处理等)
- 测试用例编号
- bug状态(提交、已解决、关闭等)
- bug提交人
- bug解决人
- bug具体现象及记录(详细)
- 备注信息
- 附件文件(可多个)
- 自定义记录(多个)

从中可以看出,想要详细的描述一个错误的状态和现象等需要这么多的字段来完成,显然,利用数据库的方式来管理这些错误记录是再合适不过的。其中的“自定义记录”充分考虑了不同项目的特点,由相关人员自己定制。项目经理可以根据项目的特点从这些字段中挑选中某些必选字段,可选字段在实际工作中使用。

此外每个字段的内容要充分结合公司现有的流程来设计,例如“bug状态”字段,按照通常的更改流程,一个bug从被软件测试人员提交的开始,一个bug就被记录到数据库中,此时它的状态是“提交”;随后,项目主管根据错误的范围指定责任人,此时bug的状态变更为“分配”;相应bug责任人接到分配任务后将bug的状态修改为“打开”,表明该bug已经在分析和解决之中;经过调试修改后责任人将bug更正,然后设置bug的状态为“解决”;软件测试人员在看到状态变化为“解决”后要确认,当测试人员确认这个bug已经修改正确完毕后,将bug的状态修改为“关闭”。刚才描述的只是一个正常的流程,实际上bug的状态还可能包含“推迟”、“拒绝”等多个状态,这些内容要完全结合公司的流程来制定,按照实际经验,开发bug跟踪系统的主要精力都应该放在这一阶段的详细分析。

3 数据挖掘设计

利用数据库建立一个表和多个表来描述错误记录只是开发bug跟踪系统的初衷,实际上项目管理人员更多的希望从这些大量的数据中得到更有价值的东西,例如有的项目经理会关心一些问题:软件开发人员最容易犯的是那些错误?哪些

软件开发人员的代码质量好?公司在软件设计的那方面需要进一步加强?这些问题没有科学的数字统计分析恐怕是很难准确回答的。那么利用公司长期的历史错误记录对项目经理的问题进行定量的分析,正是bug跟踪系统的另一大功能。

下面这些统计分析都可以从错误记录数据库中获得:

- 某个测试后(或进行阶段中)的bug统计(记录列表)
- 某个测试后bug错误类型统计(饼图,柱状图)
- 某个测试后bug严重程度统计(饼图,柱状图)
- 某个测试后每一个开发人员的bug统计分析(柱状图)
- 某个测试后每个测试人员发现的bug统计分析(柱状图)
- 某个测试解决一个bug需要的时间统计分析
- 某个测试没有解决的bug的列表分析
- 千行bug数统计分析
- 测试周期、测试投入与产出统计分析
- 发现的bug数目的时间、数量图形分析(曲线图型)
- 通过关键字的搜索可以列出相关的bug列表信息,以供查询。
- 可以通过记录的优先级别不同进行bug排序
- 针对某个项目、模块、文件发现的bug数的统计
- 针对公司历史上所有的bug进行以上分析
-

这些统计信息可以根据项目管理人员的需求不断调整,不断挖掘出用于改进软件开发过程管理需要的信息。

4 系统实现的设计

开发一套bug跟踪系统采用传统的B/S三层结构,客户端使用普通的web浏览器,web服务器可以利用公司内部intranet的web服务器资源,后台的数据库有多种选择(例如Oracle,SQL

Server等)。由于这个软件在公司内部使用,实际使用中要采用的资源最好充分结合公司自身的特点,尽量采用公司已有的资源,降低开发这套系统的软硬件成本。

对于客户端的设计,由于采用B/S方式,省去了客户端程序的安装过程,合理的安排页面布局结构,使相关人员可以方便快捷的向后台数据库增加bug信息,同时又可以找到自己需要的各种信息——这些过程是客户端界面开发的主要课题。由于该系统的用户有不同的身份,譬如项目管理人员、软件开发人员、软件测试人员、质量控制人员等,即使是同类型身份的用户由于级别的不同所关注的内容也会千差万别,例如,对于软件测试人员,他们在使用该系统的过程中最常使用的是新bug的提交以及bug的确认等过程,所以在设计提交bug的页面中要多征求测试人员的使用习惯,将常用的、必选的记录字段放在前面,不常用的、可选的记录字段放在后面等;对于项目管理人员更关心一些宏观的数据,比如一些数据挖掘的统计分析,设计实现的时候可以按照项目管理人员的习惯不同设计多种图型方式供管理人员选择;软件开发人员往往是该系统被动的使用者,当软件开发人员登录该系统后,系统立即显示最近的和该软件开发人员相关的bug列表,免去软件开发人员再深入的查找的过程。

5 小结

开发基于web方式的bug跟踪系统还可以有更广的延伸,小公司的intranet,大型项目分布式开发还可以采用internet方式,在开发该系统的过程中采用自动邮件通知的方式可以使该系统增值不少,同时在运用过程中要注意安全防护等措施。项目管理人员本着“不断改进”的思想对该系统进行不断调整,必然会给企业自身带来益处。■

参考文献

- 1 Glenfor J. Myers (The art of software testing).
- 2 Stephen Wynkoop (SQL Server 6.5 开发使用手册), 机械工业出版社, 1998.7.