系统软件研发中程序、文档、测试用例的同步技术

孙赫楠,陈 榕
(同济大学 基础软件工程中心,上海 200092)

摘 要:系统软件处于相对底层的位置,支持上层软件的开发和运行,因此系统软件的研发过程需要更准确更清晰的文档支持和更有效更全面的测试保障。但系统软件的研发是一个中间存在着很多变化的庞大而反复的过程,因此,确保文档和测试用例能够快速而准确地与其同步是十分困难的。研究了系统软件研发中程序、文档和测试用例的同步技术,提出了一套同步机制和管理方案,并将其运用于Elastos嵌入式智能手机操作系统的开发过程中,使得该系统软件的开发更加高效,研发周期大大缩短。

关键词:系统软件研发;同步;Bug管理系统
中图分类号:TP311.52 **文献标识码**:A **文章编号**:1673-629X(2008)11-0051-04

Synchronous Technique of Procedures, Documents, Test Cases in System Software Development

SUN He-nan, CHEN Rong
(System Software Engineering Centre of Tongji University, Shanghai 200092, China)

Abstract: The system software is located on the base layer, supporting the development and running of upper level software. Consequently, the development of system software needs more accurate, clearer documents as supports and more effectively, more overall tests as guarantees. But, the development of system software is a huge and repetitious process with a lot of varieties. So it is hard to keep procedures, documents and test cases synchronous quickly and accurately. Researches the synchronous technique of procedures, documents and test cases in the system software development and puts forward a set of synchronous mechanism and management project. This set of synchronous mechanism and management project is used in the development of a smart mobilephone operating system—Elastos, and this makes the development of the system software more efficient and the development cycle shorter.

Key words: system software development; synchronous; bug management system

0 引 言

系统软件的研发是一个涉及到程序设计、测试和维护的同步开发过程,与应用软件相比,系统软件位于更底层的位置,向上层提供各种应用程序设计和开发的支持,因此它的开发更加需要文档和测试用例来说明系统软件的方方面面,并测试它的性能及其它质量指标[1]。

一直以来,程序和文档以及测试用例的同步是程序开发过程中极其艰难的一件事,文档的撰写让大多数程序员感到困难。原因不在于开始撰写一份文档或编写一个测试用例,而是所有的文档和所有的测试用例必须跟随项目的变化而变化,写出来的文档越多,需要被持续维护的文档也就越多,需要反复检查的文档间可能存在的矛盾也就越多,开发人员的负担也就越来越大。于是,一些项目组将程序和文档完全分离开,程序设计人员只管代码的开发,文档的撰写留给文档人员来做,这无疑又会加大程序和文档之间同步的难度,因为一旦项目变得越来越庞大,真实的设计情况存在于代码中,而同步出来的文档可能由于错误的理解而包含了错误的东西。

在系统软件的开发工作中,有必要保持一个合理有效的同步管理机制,形成一个流畅的,能够及时同步的工作机制。

文中以Elastos智能手机操作系统的研发过程为背景,提出了一种在系统软件开发流程中保持程序、文档和测试用例同步的解决方案。

收稿日期:2008-02-27
基金项目:国家863计划资助项目(2001AA113400);国家移动通信产品研究开发专项项目(财政部(财建[2005]182号),信息产业部(信部请函[2005]297号))
作者简介:孙赫楠(1982-),女,吉林长春人,硕士研究生,研究方向为嵌入式操作系统、系统软件支撑技术;陈 榕,博士生导师,教授,研究方向为嵌入式系统、构件技术。

1 Elastos 系统的开发模式

Elastos 是国家 863 计划《基于中间件技术的因特网嵌入式操作系统及跨操作系统中间件运行平台》(课题编号 2001AA113400)支持下的一个项目,是国家移动通信产品研究开发专项项目:"面向服务的 3G 手机软件平台开发"(财政部(财建[2005]182 号)、信息产业部(信部请函[2005]297 号)),是上海科泰世纪公司历经六年半时间开发的一款网络增值业务类嵌入式系统,目前 Elastos 平台已经完成了产品化开发,并成功完成了 TD-SCDMA 双模智能手机和 GPRS 智能手机的软件整体解决方案。

Elastos 操作系统的研发包括:程序开发、文档部门、测试部门、技术支持部门、商务部门等。

系统软件开发的工作模式,是以程序设计为中心的工作模式,文档和测试随着程序的改变与其保持同步[2]。程序开发部门主要负责 Elastos 系统的研究开发并提供对所写程序的注释。文档作为技术支持部门的一部分负责针对 Elastos 的开发,搭建出符合开发人员参考和用户培训要求的文档帮助系统框架,并使用这个框架,对各研发部门开发出来的程序进行详细说明[3]。测试部门依据文档描述设计测试用例,采用各种测试方法和技术,对操作系统各部分的性能进行测试,掌握系统性能,发现 Bug,将结果通过文字和图表表达出来,并运用 Bug 管理系统提出 Bug,将修改 Bug 的要求提交给相关人员,从而引起程序或文档的改变,程序改变又要求文档和测试用例等与其同步。可以看出,这是一个以程序为中心,程序、文档和测试用例相互同步的过程。

2 系统软件开发中程序、文档和测试用例的同步

在 Elastos 操作系统的研发过程中,制定了一整套系统的同步机制和解决方案,包括文档生成工具、Bug 管理系统等,并对系统软件研发中的程序、文档和测试用例的同步机制进行了完善,使同步工作得到了明显的改善。

2.1 程序与文档的同步

程序设计人员提交代码的同时提交对这段程序的注释,文档人员首先理解这些程序和注释,然后使用文档自动生成工具生成出一部分文档资料,再添加一些必要的背景知识,按照一定的表达方式建立文档系统。

有一些文档自动生成工具可以帮助文档人员自动生成文档,例如可以将 C# .NET 编译生成的程序集和对应的 doc XML 文档,自动转换成如 .NET Framework SDK 类库文档或者 MSDN Library 在线 .NET 类库文档形式的代码文档的 NDoc 等。这些工具为文档的同步工作提供了以下方便:

(1)为文档的自动生成提供了标准,可以初步标准化文档的表达方式,例如使文档框架、文档格式、文档的查询方式等得到统一;

(2)为程序开发人员提供了统一的原始文档资料的编写方法和规范,一定程度上规避了文档生成的差异性。

在 Elastos 的研发过程中,程序和文档的同步工作中主要用了 Doxygen 文档自动生成工具来同步程序和文档以及组织一部分原始的文档资料。

Doxygen 是一个开源的、可配置的程序文档产生工具,可将程序中的特定注释转换成为说明文件,目前的版本是 1.5.4。用 Doxygen 工具生成文档,只要程序设计人员在编写代码时,按照它所制定的规定,在适当的地方加上注释,就可以利用工具依据程序的结构及注释生成文档。Doxygen 的使用分为两大部分:首先是特定格式的注释撰写;第二便是利用 Doxygen 的工具来产生文件。目前 Doxygen 可处理的程序语言包含:C/C++,Java,IDL (Corba,Microsoft 及 KDE-DCOP 类型),而可产生出来的文件格式有:HTML,XML,LaTeX,RTF,Unix Man Page。其中还可衍生出不少其它格式,如有了 LaTeX 文件后,就可以利用一些工具产生出 PS 或是 PDF 档案[4]。Doxygen 可处理下面几种类型的注释:JavaDoc 类型,Qt 类型,单行型式的注释,此外还有特别规定了的注释的写法。要生成文档,先要通过配置 Doxygen 档案来设计组织符合自己需求的文档生成格式。在配置 Doxygen 时,选中 HTML 标签中的 GENERATE_HTMLHELP 项并在 HHC_LOCATION 中设置 hhc.exe 文件的目录可以直接生成 .chm 帮助文档。用 Doxygen 工具自动生成文档的好处就在于:

1)程序设计者可以随时更改程序的注释。当需要更改一个接口说明时,不用再打开 WORD 文档去查找该接口,然后再对其进行更改。

2)注释可重用。

3)文档自动格式化。

4)可对 Doxygen 工具进行改造,扩充进自己需要的特殊功能,生成符合特定需求的文档,使文档工作变得更加方便。

对于文档人员来说,程序开发人员提供的原始文档资料只是文档雏形。程序开发者提供的原始文档资料只是针对开发的这段程序写出的注释,其背后的知识积累不可能全部呈现出来,其表达方式也存在一定

的个性。详细来说,文档人员同步程序和文档的工作包括:

(1)修改原始注释。将程序设计人员提交的原始程序注释、资料和某个特定程序开发人员的习惯表达方式重新组织整理,修改成规范标准的表达形式。

(2)填充文档资料。程序开发人员给出的注释和文档原始资料往往只是针对与其开发相关的某个点的表述,文档工作就要将必要的知识积累包含进文档,例如将这些资料编写成 html 或 xml 文件,完善扩充成一整套资料全面、表述清楚准确的文档资料。

(3)编译生成文档系统。这些整理好的文档是零散的文档资料,需要通过编译,组织成符合要求的文档系统。Elastos 的开发中将已经编写好的 html 或 xml 文件编译成一个.chm 帮助系统并将其发布于文档门户,可以使全体技术人员阅读和下载。

2.2 文档和测试用例的同步

开发人员开发出系统软件的某个模块后,测试人员要根据文档说明针对这个模块编写测试用例,并通过观察测试用例在 Elastos 平台上的执行情况掌握该模块的性能。在整个过程中,测试人员可能会发现文档描述有错误,这时测试人员就要提出文档 Bug,要求修改。

对于文档,所有技术人员都可能会查阅,来了解各个模块的信息,这也是一个检查文档正确与否的机会,任何人员如果发现文档有问题都可以通过 Bug 管理系统提出 Bug,要求对文档进行修改。文档人员对文档修改完成后,再通过 Bug 管理系统提交修改,并通知相关人员检查修改内容,测试人员按照修改后的文档重新编写测试用例。

2.3 程序与测试用例的同步

在系统软件的开发过程中,需要测试人员对程序进行大量的测试。测试帮助项目经理了解产品存在的问题,帮助开发人员减少 Bug,产生高质量的代码,帮助文档人员完善文档,帮助市场人员理解产品。

测试工作中,测试人员首先根据文档系统中对程序的描述,对被测试对象进行理解,并采用特定的测试技术和辅助测试工具对系统各部分的性能进行测试。在 Elastos 的测试中,需要测试构件的使用情况,CAR 构件支持的各种数据类型、测试参数、内存使用情况、CPU 使用情况。然后再通过对测试用例的执行情况进行汇总和分析,掌握程序的性能,将结果通过文字和图表表达出来,之后将测试结果发布,使所用的技术人员都可以看到这些汇总的数据,改进自己的程序。例如,表 1 是对 Elastos Debug 版进行测试的测试汇总:其中,总共有 6812 个文件,执行了 6598 个文件,有 214 个有 Bug 的文件未被执行,6598 个文件中执行通过的有 6597 个,有 1 个是新产生的错误,除此以外,还有 103 个 Bug 和 111 个未被处理的错误。

表 1　系统软件测试汇总表

测试汇总							
时间	文件总数	可执行文件数	跳过文件数	通过文件数	错误数	未解决的错误数	新错误
01-14-2008 07:00:22	6812	6598	214	6597	103	111	1

测试部门会利用各种文字、表格、图表的形式将测试用例执行的数据和结果反馈出来,并运用 Bug 管理系统,将修改 Bug 的要求提交给相关人员进行修改。这使得所有的技术人员对 Elastos 系统的运行情况有所把握。

2.4 Bug 管理系统的作用

在以上的同步过程中可以看到,完善的 Bug 机制在软件开发流程中起着重要的作用。但是除了完备的 Bug 管理机制,更重要的是要有良好的 Bug 管理软件和坚实的基础平台来支持,使管理与技术结合起来[5]。在 Ealstos 操作系统的研发中,为了更好地进行多人合作开发,管理工具小组开发了一套软件工程管理工具,有 Bug 系统、任务管理系统等。其中,Bug 管理系统主要包括以下一些功能:

(1)简单的用户认证和管理。能够增加和删除用户,直接使用已有的管理系统中的用户认证,不需要单独一套用户认证系统,管理起来比较简单。

(2)方便的项目和模块管理。一个 Bug 管理系统中可以有多个项目,每个项目有多个模块,能够方便地添加、删除和修改。

(3)全面的 Bug 管理功能。用户可以增加、删除 Bug,将 Bug 指定给某个人,指定 Bug 的优先级等。

(4)完备地记录、跟踪 Bug 的生命周期。从创建新的 Bug,记录相关人员寻找产生 Bug 的原因的讨论,找到处理办法到关闭 Bug,同时也要允许 Bug 的重现,继续其生长过程。

(5)提供各种 Bug 统计信息。如统计 Bug 的总数,某个人有多少个 Bug 要负责等功能。

Bug 管理系统是开发集体内部开发部门、测试部门、文档部门之间的同步保障机制,它的存在能够确保程序开发过程中程序、测试用例、文档的同步性。

3 测试数据

前面的论述中提出并详细说明了系统软件开发流程中的同步技术。目前,Elastos 系统平台研发过程中正是采用了这套同步技术来确保软件开发过程中各个环节的同步性。Elastos 系统平台研发过程中共产生文

档4326个,测试用例6598个,采用了这套同步技术后,其文档和测试用例的生存周期都有了明显的提高。

如图1和图2显示了在Elastos系统研发过程中测试用例和文档生存周期。可以看到,由于恰当运用了同步技术,测试用例和文档生存周期都有所延长,这样无疑可以大大节省软件开发中财力和物力的投入,缩短软件开发时间。

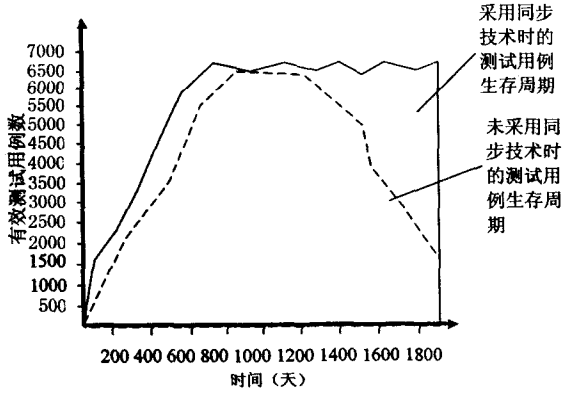

图1 测试用例生存周期的变化

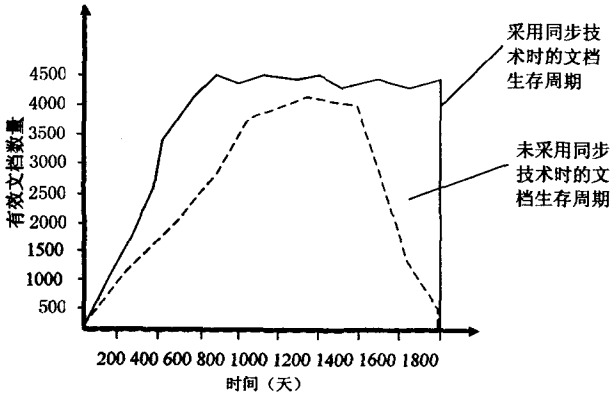

图2 文档生成周期的变化

4 结束语

程序、文档和测试用例的同步工作也是以程序的设计和修改为中心引发同步修改文档、测试用例的同步工作方式。三者通过包括文档同步工具和Bug管理系统在内的诸多技术来进行同步。这个同步机制可以通过图3来表示(箭头中斜体字表示引发同步的事件;非斜体字表示实现同步的同步机制和解决方案)。

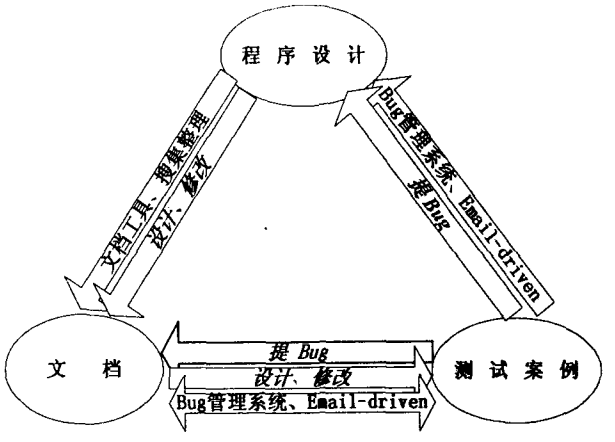

图3 程序、文档和测试用例的同步机制和管理系统

通过这些同步机制和管理方案,可以比较及时地保证系统软件研发中程序、文档和测试用例的同步。在程序发生变化的情况下,可以通过文档自动生成工具准确地同步出原始文档资料,经过文档人员将这些原始资料形成文档系统后,由测试人员开发测试用例对系统软件进行测试,验证文档和程序的正确性,如有错误再通过Bug管理系统进行文档和程序的修改和同步。这个同步过程能够使程序、文档和测试用例三者相互完善,步步同步,最终使系统软件达到成熟,呈现给用户和开发人员。

参考文献:

[1] Sommerville I. Software Engineering[M]. 北京:机械工业出版社,2003.

[2] 栾 跃. 软件开发项目管理[M]. 上海:上海交通大学出版社,2005.

[3] 田晓辉,戴金龙,沈雪芳. 如何使开发文档臻于完善[DB/OL]. 2008. http://www.uml.org.cn/bzgf/bzgf.asp.

[4] Doxygen usage[DB/OL]. 2008. http://www.stack.nl/~dimitri/doxygen/doxygen-usage.html.

[5] 孟 岩,刘振飞. Bug管理的经验和实践(中)[J]. 程序员,2005(2):38-42.

(上接第50页)

[6] Boehm B. COCOMO II model definition manual[DB/OL]. [2007-09-17]. http://sunset.usc.edu/research/COCOMOII/Cost Estimation.

[7] Fewster M, Graham D. Software test automation[M]. Boston, MA: Addison-Wesley, 1999:143-167.

[8] Skoglund M, Runeson P. A case study on testware maintenance and change strategies in system evolution[DB/OL]. 2004-11-14[2007-09-18]. http://www.ieeexplore.ieee.org/iel5/9383/29793/01357831.pdf.

[9] RoleModel Software, Inc. Drawlets[DB/OL]. [2007-09-17]. http://www.rolemodelsoftware.com/drawlets.

[10] Rajlich V, Gosavi P. A case study of unanticipated incremental change[C]//Proceedings of the International Conference on Software Maintenance (ICSM'02). Los Alamitos, CA, USA: Computer Society, 2002:359-368.