doi: 10.7690/bgzdh.2014.05.026

基于可信性思想的实时测控软件系统开发方法

贾海艳
（中国人民解放军 92941 部队，辽宁 葫芦岛 125000）

摘要：根据新形势下实时测控软件系统的要求，提出基于可信性思想的系统开发方法。从软件可信性的角度，探讨实时测控软件系统的开发方法，按照可信性要求制定出软件开发和测试规范，采取简化设计、容错设计与自启动技术、配置管理技术相结合的方法来确保软件可信性，并结合实例给出具体的设计方法和测试方案。结果表明：该方法能提高软件整体质量和开发速度，保证软件开发的可信性。

关键词：可信性；软件开发；软件测试；配置管理
中图分类号：TJ02　　**文献标志码**：A

Development Method of Real-Time Measurement and Control System Based on Dependability

Jia Haiyan
(*No*. 92941 *Unit of PLA*, *Huludao* 125000, *China*)

Abstract: According to the requirement of real-time measurement and control system in new condition, this paper proposes the methods of system developing based on dependability. In the view of dependability, the developing methods of real-time measurement and control system are analyzed, the specification of software development and test are made in term of dependability, in detail, the technology of simple design, fault tolerant design and self-restart, configuration management are developed for assuring the dependability, a practical example for design and testing schemes are proposed. Results indicate that the methods introduced could enhance the quality of software and the speed of development, meanwhile assuring the dependability of software research and development.

Keywords: dependability; software development; software test; configuration management

0　引言

在航天测控领域中，实时测控软件占有关键的地位。软件失效所产生的后果无法估量。实时测控软件系统是一个具有多进程多线程结构、信息量大、接口复杂的大型应用软件系统，主要完成航天器的测量和控制任务。软件可信性要求综合了对软件可靠性、安全性的要求，更能体现测控软件的特点。其可信性包括实时性、可靠性、可维护性、机密性、可用性。实时性是实时测控软件系统的最基本要求。可用性是软件系统能够正确完成需求规定的功能。可靠性是在规定的运行环境中和规定的时间内软件无失效运行的机会。可维护性是软件具有重用、修改方面的能力。机密性指软件系统应具备抵抗泄密的能力。

随着新形势下对测控系统需求的提高，对实时测控软件系统的开发也提出了更高的要求[1]。传统的开发方法是面向功能的，主要注重的是软件的可用性。根据新形势下实时测控软件系统的要求，笔者针对软件质量管理和软件开发技术 2 方面，从软件可信性的角度，探讨了实时测控软件系统的开发方法，以提高软件整体质量和开发进度，促进软件开发技术的发展。

1　基于过程控制的软件开发

软件开发过程直接影响着软件可信性。实时测控软件系统的开发过程首先由专业知识领域的人员组成项目组，之后按照软件工程要求在规定时间内完成工作[2]。而软件系统的开发规范是保证软件系统可信性的首要因素。针对实时测控软件系统的特点，软件系统开发的每一步骤都应严格制定相应的软件开发管理规范，对软件的开发过程和质量实施全程监控，这是保证软件质量的基本要求。软件开发过程通常采用改进的 V 模型，如图 1 所示。

首先制定总体方案，分析需求，编写《软件需求说明》，由专家组对项目需求文档进行评审，项目组中的软件测试人员按照《软件需求说明》编写《软件测试需求说明》并评审通过。接下来进行概要设计和详细设计的编写和评审。测试人员也应编写测试计划，进行测试概要设计和详细设计的编写和评

收稿日期：2013-12-09；**修回日期**：2014-01-19
作者简介：贾海艳(1974—)，女，辽宁人，硕士，工程师，从事数据处理研究。

审。最后进行代码的编写与测试的实施，并对测试结果进行分析总结。

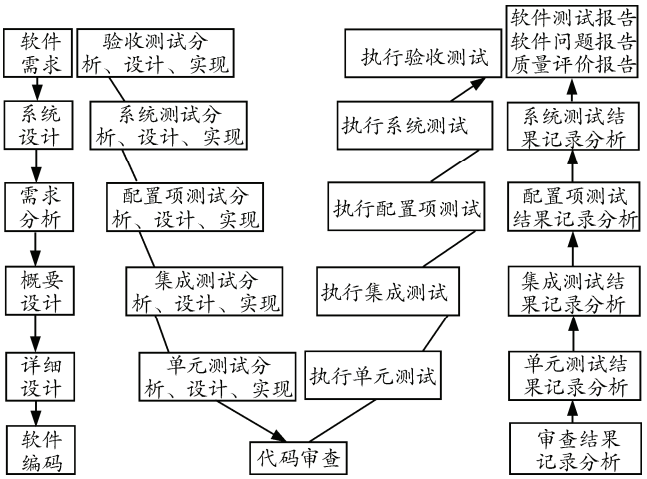

图 1　软件开发 V 模型

采用上述制度化、规范化开发过程，相关人员会对软件开发过程有一个宏观的控制，减少了人为造成的问题和故障，加快了开发进度，保证了软件开发的可信性。

2　可信软件设计方法

2.1　可靠性软件设计

实时测控软件系统规模庞大。软件系统的复杂性降低了软件的可用性、可靠性；因此，在软件设计时，采用层次化和模块化设计方法及软件重启技术来保证软件可靠性。

采用层次化和模块化设计方法，使软件结构简单，缩小了程序规模。实时测控软件系统按照功能分为前端控制和后端数据处理 2 层。前端控制层分为记录控制、目标控制等模块，后端数据处理层分为目标运动参数处理、引导数据处理等模块。每个模块采用类的处理方法，例如将外测数据处理模块设计为 1 个类，每个成员函数只实现 1 个功能。程序模块应该只允许访问那些与自身实现相关的数据，其他程序数据对模块是隐藏的，因为隐藏的信息不会被无关模块破坏。这样使开发设计思路清晰，降低了程序的复杂度，保证了软件的可靠性[3]。

软件重启技术的基本思想是在采用双工和双机备份的系统条件下，在软件运行期间，周期性进行重启，清除系统内部状态变量，从而对软件故障进行预防。尽管所设计的软件系统经过优化设计和严格测试，但软件的属性决定了其不可能完全无差错，并且也无法避免各类缺陷的存在，只是从缺陷的数量上实现了控制。实践结果表明：在软件长时间运行后均会发生一定的错误，如内存泄露、文件块未释放、指针错误、舍入误差、系统资源数据操作冲突等，最终导致性能下降，影响软件可靠性，这类缺陷无法通过设计方法解决，通过测试的方法找出这类缺陷，需要耗费大量的人力和物力，其性能提升效果并不明显。根据环境多样性设计原则，采用了自恢复技术进行解决，操作包括切换、停机和重启动操作。操作周期的选择则根据实际工作经验和优化方法决定，以最小化系统损失[4]。

2.2　实时性软件设计

在保证软件可用性、可靠性的同时，必须兼顾实时测控软件系统的实时性。容错设计可保证实时测控软件系统的实时性。容错设计核心方法是 N 版本编程方法和恢复模块方法，其利用软件设计多样性技术，由不同小组使用不同版本的软件来完成同一功能，以时间和空间的冗余性来达到容错的目的。

N 版本编程方法其工作原理如图 2 所示。针对同一软件需求，由项目不同人员实现一组不同的版本，所有版本都同时进行计算，并用 1 个输出比较器选出可信的输出。输出比较器可以是 1 个简单的软件模块，用某种选举机制来选择输出。在实时测控软件系统设计中，要求从不同版本选出的输出结果的所有过程必须在某个时帧内完成，以确保软件响应的实时性。

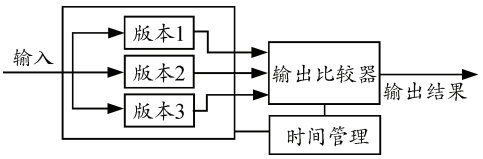

图 2　N 版本编程

恢复模块设计是通过建立还原点并使用可接收测试和后向恢复实现动态容错。其伪码实现如下：

```
ENSURE      acceptance_test
BY          {   module_1    }
ELSE BY     {   module_2    }
…
ELSE BY     {   module_m    }
ELSE ERROR
```

2.3　可维护性软件设计

实时测控软件系统的特点要求经常需要根据任务需求的变化，对相应功能模块进行调整，也就是说对软件的可维护性提出了很高的要求[5]。软件配置管理是一种标识、组织和控制变更的技术。软件维护过程中采用的配置管理技术可以使因变更而引

起的错误降为最小，有效地保证软件产品的完整性和生产过程的可视性。在软件生存周期内所产生的各种管理文档和技术文档、源代码列表及其可执行代码，以及运行所需的各种数据都属于软件配置管理项。配置管理有专业的配置管理员和管理工具，但管理工具一般都缺少流程控制，缺少本地化和易用性，因为实时测控软件的机密性决定开发是由参与航天试验任务的人员来完成，一般都属于小团队开发，采用符合自己特点的管理工具更加方便、清晰、完整。如图 3，软件配置管理登记表就是方法之一。

<table>
<tr><td>时间</td><td colspan="2"></td><td>软件配置管理组成员</td><td></td></tr>
<tr><td>软件项目名称</td><td colspan="4"></td></tr>
<tr><td rowspan="14">软件更动说明</td><td rowspan="5">软件现行配置</td><td colspan="2">软件文档名称及存储路径</td><td></td></tr>
<tr><td colspan="2">软件程序名称及存储路径</td><td></td></tr>
<tr><td colspan="2">软件数据名称及存储路径</td><td></td></tr>
<tr><td colspan="2">软件预想更动策略</td><td></td></tr>
<tr><td colspan="2">软件更动所需时间</td><td></td></tr>
<tr><td rowspan="2">软件评审情况</td><td colspan="2">软件评审组成员</td><td></td></tr>
<tr><td colspan="2">软件更动批准时间</td><td></td></tr>
<tr><td rowspan="2">软件开发情况</td><td colspan="2">软件开发组成员</td><td></td></tr>
<tr><td colspan="2">软件开发开始及完成时间</td><td></td></tr>
<tr><td rowspan="2">软件测试情况</td><td colspan="2">软件测试组成员</td><td></td></tr>
<tr><td colspan="2">软件测试开始及完成时间</td><td></td></tr>
<tr><td rowspan="3">软件验收情况</td><td colspan="2">软件测试发现问题</td><td></td></tr>
<tr><td colspan="2">软件验收组成员</td><td></td></tr>
<tr><td colspan="2">软件验收时间</td><td></td></tr>
<tr><td rowspan="2">软件更改后配置</td><td colspan="2">软件文档名称及存储路径</td><td></td></tr>
<tr><td colspan="2">软件数据名称及存储路径</td><td></td></tr>
<tr><td rowspan="2">软件维护说明</td><td>软件适用范围</td><td colspan="3"></td></tr>
<tr><td>软件使用时间</td><td colspan="3"></td></tr>
</table>

图 3　软件配置管理登记表

图 3 中程序的标识需有程序的首部，在程序的首部应有程序的标识符、程序的作用、程序的作者、程序的完成日期及上次修改程序的人员姓名、日期及其原因。数据标识应注明本软件项目开发和使用过程中用到的各种基准参数值。软件问题说明应详细说明问题的症状、性质、预计的影响范围(本次修改可能涉及哪些文档、软件、数据)。软件更动类型填写纠错、改进或扩充。软件更动涉及范围填写文档、程序、数据、系统功能及性能。软件更动所需时间指软件从计划更动到投入使用所需时间。

3　分析对比

在某次实时测控软件系统的升级改造中，严格执行了上述软件开发过程。实际结果表明：本次软件开发代码编写规范，软件开发时间缩短，软件可信性提高。表 1 为其与某次规模类似的软件系统开发的对比。可信性值是采用上述方法计算的结果，分为优秀、良好、合格、不合格 4 个等级。图 4 是软件使用中缺陷的统计。

表 1　软件系统开发结果比较

名称	文中开发方法	传统开发方法
开发周期/y	3	4
使用中最初 6 个月出现问题数	5	9
可信性值	89(优秀)	79(良好)

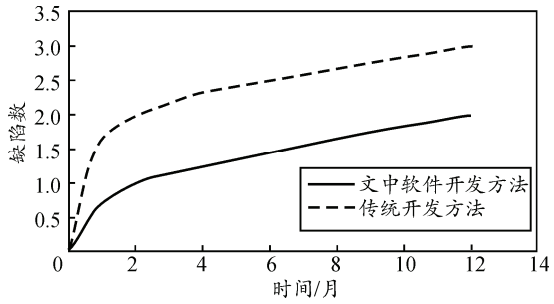

图 4　软件缺陷统计

4　结束语

软件可信性可以通过建立多层次的综合评估体系进行评测。首先赋予各功能模块相应权值，对于那些与主要功能密切相关的功能模块，赋予它较高的权值，在量化结果中体现出各模块的不同侧重。然后建立评价指标，通过专家打分和计算各指标不同的权值系数得出实时测控软件系统的可信性。无论是何种测试手段都无法测出软件中的所有缺陷，尤其是对实时测控软件系统这样规模庞大的应用软件系统，因此在使用软件的过程中要搞好维护性测试，以确保实时测控软件系统安全可信。

参考文献：

[1] 吴集, 沈雪石, 赵海洋. 新一代软件技术发展及其军事应用展望[J]. 兵工自动化, 2012, 31(6): 86-89.

[2] Ian Sommervile. 软件工程[M]. 北京: 机械工业出版社, 2012: 5-100.

[3] 苗扬. 软件可靠性测试与评估方法的改进[D]. 上海: 上海交通大学, 2010: 3-20.

[4] Jeffrey Voas, Jeffrey Payne. Dependability certification of software components[J]. The Journal of Systems and Software, 2000, 52: 165-172.

[5] 黄学德. 导弹测控系统[M]. 北京: 国防工业出版社, 2000: 5-50.