

发动机试车控制软件工程化及质量管理

蒋 瑜, 王秋红

(西安航天动力试验技术研究所, 陕西 西安 710100)

摘 要: 为提高发动机试车控制软件的可靠性, 通过需求分析、概要设计、详细设计、软件测试等步骤实现了软件开发工程化。结合试验控制软件研制和使用特点, 对软件质量管理的具体步骤, 即从设计评审、测试、验证、文档及技术状态管理等方面对软件开发过程进行监督与管理, 实现了软件开发的质量控制, 达到了软件设计的透明性、继承性及高可靠性。

关键词: 火箭发动机; 试验; 控制软件; 可靠性; 质量管理

中图分类号: V434.3

文献标识码: A

文章编号: (2008) 04-0060-05

Approach for improving reliability of rocket engine test control software

Jiang Yu, Wang Qiuhong

(Xi'an Aerospace Propulsion Test Technique Institute, Xi'an 710100, China)

Abstract: To improve reliability of the control software for rocket engine test, an engineering method is introduced into the developing process of the software through requirement analysis, brief design, detail design and software testing. Also quality management approaches are carried out considering the features of development and application of the software, which are supervising and managing the developing process from the aspects of review, test, validation, documenting and technical status control. All the measures reach the goal of high reliability software development.

Key words: rocket engine; experiment; control software; reliability; quality control

1 引言

机试验的成败, 因此与控制系统的硬件一样, 软件可靠性是不容忽视的。

由软件故障引发的、造成灾难性后果的事件控制软件的可靠性直接关系着液体火箭发动 已有很多。例如, 1991 年海湾战争中, 美军使用

收稿日期: 2007-07-26; 修回日期: 2007-10-23。

作者简介: 蒋瑜 (1973—), 男, 工程师, 研究领域为液体火箭发动机试验测控技术。

爱国者导弹,在拦截伊拉克飞毛腿导弹中,出现过几次拦截失败事件,经查明是软件计时系统的累积误差所致,而且爱国者导弹也因软件问题误伤28名美国士兵。1996年欧空局首次发射阿里安5型火箭时因控制系统的软件故障而失败,直接损失近5亿美元,还使历经10余年、耗资已达80亿美元、上万人心血的开发计划推迟了近三年。

软件工程化是把系统工程的原理与软件特点相结合的产物。实践证明,实施软件工程化是保证软件可靠性的基础。目前试验控制软件的开发离工程化程度还很远,仅仅在需求、测试方面做了些工作,由开发设计人员“自行设计、自行编码、自行测试”,程序的质量和可靠性依赖于开发人员的技术水平。当软件需要维护时,只能由开发者自己“包办”,其它人员难以介入,形成所谓“三自一包”的生产模式。由于无法对软件开发过程进行有效的监督与管理,以至软件的透明性、继承性都较差。显然,这种状态难以适应高可靠性试验要求。

控制软件属于试验领域中的应用软件,突出特点是可靠性要求高,实时性强,以技术人员为研发主体,应用领域和具体用途保密,很难交流和横向合作或任务外包,现场更改需求也时有发生,特别是在研究性试验的过程中。这些特点决定着高可靠性和高质量的软件主要依靠本单位研发人员和质量管理部门的共同努力。

2 控制软件开发的工程化步骤

软件工程化是指使用软件工程的理论、技术、要求和管理等来规范软件开发过程中的全部活动。实际上,它是硬件生产流程的移植,软件工程就是把工程化应用于软件生产中。具体地说,一个基本的软件开发流程包括需求阶段、系统(概要)设计阶段、详细设计阶段和测试阶段等这些易于遵循的标准化开发步骤,每一开发步骤又被定义为里程碑。

2.1 软件需求分析

软件需求包括将要开发的控制软件所涉及的

概念、定义、依据、指标(通道数、定时精度等)、功能、控制逻辑、算法、硬件环境、现场环境、时序、执行过程和特点(发动机点火启动和关车控制、紧急关车处理和数据采集系统等分系统间的接口细节)、运行平台和编程语言、人机界面、可靠性、扩展能力以及没有明确提出来的现场需求变更和试车台工艺控制等隐含需求。有时需要咨询相关的发动机设计部门和试验领域专家,以免产生模糊概念和歧义。需求分析要求全面、细致和深入,不要因分析不周再从头修改软件,软件修改相当麻烦,往往有牵一发而动全身的问题。对于现场需求变更,要用软件的灵活性特点来适应这种情况。通过需求分析,最终产生软件需求规格说明书。

2.2 控制软件设计

控制软件设计包含概要设计和详细设计。概要设计是指将控制软件功能模块初步划分,并给出合理的流程和资源要求,构造出软件结构,结构中的每一部分都是意义明确的模块,每个模块与需求相对应,对每个模块的工作进行具体的描述,编制软件框图。设计说明书应当提供具体的模块(黑箱结构),使得系统整体模块化程度达到最大,一份好的概要设计说明书,可以使编码的复杂性降到最低。有的开发人员往往跳过了概要设计阶段,而是先有编码,为了检查才补充设计,事倍功半。

详细设计又称代码设计。详细设计是概要设计的继续,主要目的是完成概要设计对象内部逻辑的实现,在开始软件编码之前应完成所有的设计细节,避免在编码过程中进行设计工作,编码是软件详细设计的一种再现。软件要清晰直观,不要过于曲折。软件越透明,就越容易正确。消除那些无效多余代码,例如设置一个变量后却不使用,多余代码的出现反映出设计人员思路不清,因此有可能在此附近出现错误。有时开发人员往往为了节约开发时间拷贝成熟代码时并没有对被拷贝代码的含义做足够的研究,这些代码有可能在当前软件中出现互斥,引发死锁或者竞争。下一步是对软件代码进行编译和初步测试,初步测试包括静态测试和单元测试,静态测试就

是对源程序每条指令进行书面走读审查。单元测试是测试中最小单位的测试，在面向对象编程中，又称为类测试，它需要从程序的内部结构出发设计测试用例。例如设计一些用例测试其内部的控制点（如：条件判断点、循环点、选择分支点等），再如对于一个函数，输入自变量测试其因变量是否正确，可用弹出信息窗口输出数据验证等等。只进行了单元测试的软件，对代码的测试尚不完整，未覆盖的代码可能遗留错误。进行充分的单元测试是提高软件质量的必由之路。单元测试可看作是编码工作的一部分，应该由程序员完成。代码及其链接、集成和构建必须通过编译，使用开发工具所含的编译功能对软件源码进行检查，分析和寻找源码存在的问题。编译通过只是说明了它的语法正确，但无法保证它的语义也一定正确，语义正确要得到验证和确认。通过了单元测试、编译、验证和确认的代码才是已完成的代码。上述过程完成后，最终形成用某一种特定程序设计语言表示的源程序清单。

2.3 软件测试

包括静态测试和动态测试，这里仅指动态测试中的确认测试和系统测试。动态测试的主要目的是通过在相似环境或相同环境中执行软件，以证实软件需求是否正确实现，并通过测试找出所有的错误。软件可靠性测试是软件可靠性保证过程中非常关键的一步。从工程的角度来看，一个高可靠性软件不仅意味着该软件的失效率低，而且意味着一旦该软件失效，由此所造成的危害也小。试车控制软件测试的侧重点不同于一般的软件功能测试，其测试实例设计的重点是寻找对可靠性影响较大的故障。

确认测试又称有效性测试或功能测试。软件需求规格说明书中包含的信息就是软件确认测试的基础。进行有效性测试是在模拟的环境下，控制对象用模拟等效器代替，将软件看作一个不能打开的黑盒子，在完全不考虑软件内部结构和内部特性的情况下，运用黑盒测试的方法，验证被测软件是否满足需求规格说明书列出的需求，软件是否能正确地接收输入数据而产生正确的输出信息。

系统测试把经过确认的软件纳入实际运行环境中，与其它系统组合在一起进行测试，亦即发动机试验准备期间和开车前所必须进行的试验系统综合测试。系统测试的目的在于通过与系统的需求定义作比较，发现软件与其它系统的定义不符合或与之矛盾的地方。软件测试不是一个固定的呆板的框框，而是一个有弹性的概念。要根据实际情况，科学合理有重点地去做，有时需要追加其它的测试。完成验收并完成最后的测试文档，工程化才算告一段落。

3 软件开发工程化的质量管理

在软件开发中虽然引入了工程的概念、原理、技术和方法，这种思想在一定程度上解决了软件生产过程中遇到的质量问题，没有软件工程，软件质量不可能得到保证，但有了软件工程并不意味着质量就一定能够保证。前述的阿里安5型火箭在发射后不到40秒爆炸，事后调查发现，错误发生于当一个64位浮点数转换为16位带符号整数时出现异常。仅这个极微小的错误，导致十余年的努力毁于一旦。欧空局和法国空间局共同发布的阿里安5运载火箭事故调查委员会的调查报告称：“如果对飞行控制系统进行了充分分析和测试，则有可能检查出这种潜在的失败因素”。软件质量管理工作不完善、不严格，是造成潜在事故的原因之一，它给人们敲响了重视软件研制过程质量管理的警钟。为确保软件工程化的有效开展，使保证软件质量的任务贯穿在整个工程化过程中，就要紧紧抓住需求分析、设计、测试、评审、验证与确认等几个主要环节的质量管理。通过保证每个环节的工作质量来保证最终的软件质量。

应严格按照软件工程要求进行软件开发，把好每一阶段的质量关，没有经过评审、验证与确认，不得转入下一阶段的工作。

3.1 准确分析软件需求

需求分析的目的是使软件设计人员对控制软件有全面和深入的理解，以明确试验所需的究竟是一个什么样的软件。因此，需求分析是软件生

产过程中的一个首要步骤。一份规格说明书的质量优劣,取决于分析人员理解需求的正确性、完整性、合理性和一致性的程度。软件需求分析的过程,也是软件设计方案的酝酿过程。通过分析应得出需求的正确性、合理性和完整性的结论;同时,也应得出软件付诸实现的可行性、可靠性的结论。需求分析是软件可靠性设计中最容易被忽视、但又最容易导致不可靠的一个重要因素。满足需求是软件最基本也是最主要的质量目标。

完成软件需求规格说明书的编写,通过正式的评审和确认、完成归档,使其成为软件设计、调试和测试的基础以及评审、鉴定和验收的依据之一。

3.2 设计是质量重点

软件是产品,从产品的意义上说,所谓软件设计开发就是软件生产。软件和硬件一样,它的质量也是设计、生产出来的。因此,质量控制的重点应放在设计阶段。一个好的设计基本上决定了产品的最终质量。设计是把需求转换成产品的一个关键步骤,它把自然语言描述的需求变成用计算机语言构建的软件,编码是软件详细设计的一种再现,需要编程人员全神贯注地操作。根据软件单元测试结果对软件进行必要的修改,形成软件设计说明书、软件框图、概要与详细设计说明书和源程序清单等文档,并通过静态复测(书面走读)、审查、评审。

3.3 测试是质量保证的关键

控制软件在投入使用前,对其进行严格的测试是质量保证的关键,它体现了对需求分析、规格说明、设计和编码的最终复审,是实现可靠性的必要手段。在面向产品的质量控制活动和面向过程的质量保证活动中,软件测试是实现软件质量目标的最重要方法。不对控制软件进行严格测试,就无法排除许多隐藏在软件中的错误和缺陷,甚至是某些致命的错误。对软件测试的每一步,都要有严格的质量管理,由质量部门建档,并监督落实每个问题改进的情况。对软件测试常常要求同时进行硬件检测,对响应时间、处理精度、定时精度、时间累积误差、数据输入/输出的

正确性等需求指标必须进行严格的测试。软件的可靠性是规定条件下的可靠性,即实际运行条件下完成任务的可能性,不能仅满足于几个测试用例演示,软件仅停留在正确性层次上已远远不够,还需要在一定的环境条件下(模拟试验环境)进行测试,有时甚至需要在更恶劣的电磁干扰、温湿环境下进行所谓强度测试。

对于试验控制软件,暴露出那些在试验特定环境、特定条件下才能产生的错误,特别是人为或环境造成输入数据异常和控制时序异常而反映出的软件错误要特别注意。对于暴露的问题,持续不断地运用 PDCA 循环进行 FMECA(可靠性的故障模式、影响及危害性分析)、FTA(故障树分析)、改进设计,直至错误完全归零。

3.4 坚持进行阶段评审

要求在软件研制阶段的里程碑点进行软件评审。主要有需求评审、概要设计评审、详细设计评审、测试评审和验收评审等。评审的目的则是在过程中及早地消除缺陷,减少后续阶段的返工,避免上一个阶段引入的缺陷遗留到下一个阶段,尤其在需求分析和软件设计等重要阶段进行严格的评审是发现错误、提高可靠性的有效办法。对评审出的问题进行整理、分类和汇总,不忽视任何一个细小的疑点。

3.5 软件验证与确认

软件验证(Verification)和确认(Validation),简称为 V&V。验证与确认是贯穿于软件开发过程中的软件检验活动。转入下一阶段之前必须对本阶段结果作出确认。验证和确认的主要方法有:测试、评审和正确性证明等。

硬件/软件协同验证是一种验证和确认控制系统中硬件和软件质量的技术,协同验证系统由另一个系统组成,它不仅能验证软件的正确性和可靠性,而且能验证控制系统硬件的质量。目前使用的阀门电流信号采集系统正是这种技术的应用,是质量验证的有效方法。它通过对执行对象产生的电流波形进行采集、分析,验证控制软件的时序、定时精度、时间累计误差、功能是否满足需求,是控制软件投入发动机试验的最终验证和确认。

