航天器软件测试技术与实践

候成杰　江云松　编著

国防工业出版社
National Defense Industry Press

航天器软件测试技术与实践

侯成杰　江云松　编著

国防工业出版社
·北京·

内 容 简 介

本书系统地阐述了软件测试的基本概念，讲解了航天器软件测试的各种基本方法和技术。在此基础上，介绍了航天器软件工程概况和研制技术流程，重点讲解了测试覆盖性分析技术、航天器软件测试环境搭建技术、资源访问冲突分析技术、代码更动影响域分析技术等航天器软件测试的几项关键技术，并介绍了这些技术和方法在航天器软件各阶段测试过程中的应用。此外书中还对一些航天器软件典型故障案例进行了分析，希望可以帮助测试人员积累经验，提高技术水平。

本书适合具有一定测试经验的人员使用，也可供从事航天器等领域的软件研制和管理工作人员参考。

图书在版编目（CIP）数据

航天器软件测试技术与实践 / 候成杰，江云松编著. —北京：国防工业出版社，2017.6

ISBN 978-7-118-11382-2

Ⅰ. ①航… Ⅱ. ①候… ②江… Ⅲ. ①航天器－应用软件－测试 Ⅳ. ①V4-39

中国版本图书馆 CIP 数据核字（2017）第 164390 号

※

国防工业出版社出版发行

（北京市海淀区紫竹院南路 23 号 邮政编码 100048）

三河市腾飞印务有限公司印刷

新华书店经售

*

开本 787×1092 1/16 **印张** 7¼ **字数** 198 千字

2017 年 6 月第 1 版第 1 次印刷 **印数** 1—2000 册 **定价** 50.00 元

国防书店：（010）88540777 发行邮购：（010）88540776

发行传真：（010）88540755 发行业务：（010）88540717

前　言

软件测试是航天器软件研制过程的一个重要阶段。随着航天器软件工程化的不断推进以及航天器软件规模和复杂性越来越大，对于航天器软件测试的技术要求也越来越高。航天器软件以嵌入式软件为主，因此面向嵌入式软件的各项基本测试技术都可以应用在航天器软件的测试过程中。航天器软件具有安全性和可靠性要求高、不易维护、实时性要求高等技术特点。航天器软件研制和测试人员在各类航天器软件研制的工程实践中积累了一些成功的技术经验，并对一些基本的测试技术进行了深化和发展，形成了一些独特的针对航天器软件测试领域的技术。

作者从事航天器软件测试工作 20 余年，先后负责过载人航天、探月工程、各种卫星等国家重点型号的数百个航天器关键软件的测试工作。从 2006 年起担任中国空间技术研究院软件专家组成员，先后独立或者参与编制多份航天器软件测试、航天器软件产品保证方面的技术标准。作为评审专家先后参与过多个重大型号的数百个软件的出厂专项评审工作，并先后为航天各院所、中电集团、中科院等从事航天器软件研制工作的单位软件研制和测试人员授课 50 余次。

本书结构合理，内容丰富，既有测试基本技术的介绍，又有结合最新航天器软件工程实践对基本技术的讲解，也有对最新航天器软件故障问题案例的分析和总结，对于从事航天器软件研制、测试的技术和管理人员提供了技术总结和经验教训分析。本书适用于从事航天器软件研制、测试的技术人员和管理人员、质量管理人员、型号两总系统人员，也可以作为软件研制和监督检查的参考书。希望本书能为提高航天器软件测试水平、保障型号任务成功贡献微薄力量。

侯成杰同志负责全书统稿，并负责编写了第 1～7 章的主要内容。董燕同志参与了第 4.1 节、4.4 节的编写工作，陈睿同志参与了第 5.2、5.3 节的编写工作，江云松同志参与了第 6.4、6.5 节的编写工作，高猛同志参与了第 7.1、7.2 节的编写工作，左万娟同志参与了第 7.3 节的编写工作。

本书在编写过程中，参阅了大量的国内外图书、标准、规范、报告、论文，吸纳并借鉴了许多专家和学者的研究成果和实践经验，在此表示衷心感谢！由于本人水平有限，书中难免有谬误和不妥之处，恳请同行专家、学者和广大读者批评指正。

侯成杰

目　录

第 1 章　软件测试定义

1.1　概　　述

什么是软件测试？软件测试的目的是什么？软件测试的目的到底是为了发现问题还是验证软件没有问题呢？

针对上述问题的答案一直存在争议。著名的软件测试专家 G.J.Myers 在《软件测试艺术》一书中给出了如下的观点：

（1）测试是为了发现程序中的错误而执行程序的过程。

（2）好的测试方案是最可能发现迄今为止尚未发现错误的测试方案。

（3）成功的测试是发现了至今为止尚未发现错误的测试。

根据这个观点，软件测试是以查找错误为中心的，而不是为了演示软件功能的正确性。

当然不要认为发现错误是软件测试的唯一目的，查找不出错误的测试同样具有价值。通过软件测试找出软件存在的错误，分析错误产生的原因和错误的发生趋势，可以帮助项目管理者发现当前软件开发过程中的缺陷，以便及时进行改进。

同时，对于软件错误的分析也能帮助测试人员改进测试方法，优化测试用例，提高测试效率。没有发现错误的测试同样有价值，完整的测试是评价软件质量的一种方法。

1.2　软件测试的基本原则

软件测试的基本原则，包括独立性原则、尽早开始原则、正常异常组合原则、可复现原则、80-20 原则、有序原则等，下面分别进行说明。

1.2.1　独立性原则

独立性原则是指：为了达到最佳效果，最好由独立于开发者的其他人员来测试软件。这种独立既可以是同组的其他人员，也可以是不同的软件小组，当然也可以是不同的部门甚至是独立的第三方测试机构。

1.2.2　尽早开始原则

尽早开始原则是指：越早开始测试，发现缺陷后修改的代价就越小。同时在测试开始执行前提早进行被测软件的背景理解、计划制定、用例设计和测试环境搭建等准备工

作。例如，测试计划可以在需求分析一完成就开始，测试用例设计可以在设计一完成就开始。

1.2.3 正常异常组合原则

正常异常组合原则是指：设计测试用例时不仅要考虑合理的输入条件，更要注意不合理的输入条件。而且要注意：不同的异常输入条件之间要输入一组正常输入条件使软件恢复正常，否则很有可能出现发生了异常，但是却无法判断是前次输入的异常输入条件还是后输入的异常输入条件起了作用。

1.2.4 可复现原则

可复现原则是指：软件测试过程应该可复现，测试中发现的软件缺陷都应该能以规范的方式重现，而不能依赖于某些随机因素。应对测试用例和测试环境进行改进，以确保软件测试能够复现。

1.2.5 80-20 原则

80-20 原则是指：发现问题多的地方，隐藏的问题也多。因此针对发现问题多的软件或者模块，应进行重点测试。

1.2.6 有序原则

有序原则是指：软件测试应该是个有计划、有组织的过程，不同阶段的测试活动侧重于从不同层次发现软件不同方面的设计缺陷。如单元测试侧重于软件详细设计和代码质量，确认测试侧重于软件功能和性能与需求的符合性。不进行单元测试而直接进行确认测试，并寄希望在确认测试中发现单元测试能发现的问题，是不现实也是不科学的。

1.3 基 础 概 念

首先需要明确一些软件测试的基础概念，包括测试目标、测试对象、测试依据、软件缺陷等。理解了这些概念，才能对后续内容有更好的理解。

1.3.1 测试目标

在抽象模型中，测试目标包括指定的被测软件、要考察的质量属性和对测试对象如何使用测试结果信息的描述。在实际测试中，可以具体化为发现软件的缺陷、验证软件的功能、评价软件的性能及提高软件的可靠性等。

1.3.2 测试对象

测试对象指具有被测属性的软件实体。测试对象具有被测属性和软件实体两个重要

属性。

软件实体是一个内涵十分广泛的词语，它可以定义为软件产品的集合，包括程序代码和各种规格说明。具体来说，典型的测试对象包括软件代码或代码的子集（某个函数或者程序模块）、需求规格说明、软件的体系结构、软件的设计文档以及其他软件生命周期中的相关产品。

被测属性是测试对象的核心，它是指软件实体具有的可以被测试的某种固有属性。一个软件实体具有许多固有属性，例如功能属性、性能属性等。只有在明确指定了测试中要考察的属性类型时，一个软件实体才可以被称为测试对象，否则它仍然只是一个软件实体。测试属性是决定测试中其他对象的基础。例如，如果只说要对一个嵌入式软件进行测试，而没有说明对软件的何种属性进行测试，这种测试是无从下手的，因为并不知道是进行功能、性能还是强度测试，也无法确定测试的依据是什么，更不要说采用什么样的测试策略了。

1.3.3 测试依据

测试依据是指根据测试目标制定的能够对测试对象做出明确判断并得到测试结果的标准。

测试依据一般要具备两个条件：合法性条件和唯一性条件。

（1）合法性条件要求测试依据与测试对象之间要具有某种合理的关系，使其可以作为测试结果的判断标志。一般来说，测试依据的内容一定要包含测试对象的相关信息，尤其是测试对象中测试属性的详细资料。例如，通常选择需求规格说明作为软件功能测试的依据，选择某种成熟的覆盖率准则作为单元测试的依据。合法性条件是测试依据存在的前提。

（2）唯一性条件是明确测试结果的必要条件，也是任何一个测试依据所必须具备的。实际上测试依据和测试对象有许多重叠的部分。软件开发是一个递进的过程，一个阶段的产品往往会作为下一个阶段产品的依据，同时也被作为测试依据。因此同一个软件产品可能会在不同阶段作为测试对象和测试依据两种角色。

1.3.4 软件缺陷

一般将测试对象与测试依据之间的不匹配称为缺陷。在软件测试活动中有多个术语用来表示软件与相关依据之间的不一致情况，包括失效、故障、缺陷等，这些概念经常容易混淆，下面分别进行解释说明。

（1）失效（failure）。失效是软件运行时的表现。

（2）故障（fault）。故障是发生失效的原因所在。

（3）缺陷（defect）。缺陷是程序中一种潜在不安全因素，特定的输入触发了程序的缺陷时，软件才会发生故障继而导致失效。

一般可以采用简化的关系，认为缺陷存在可能导致软件表现出失效。

1.4 软件测试的分类

软件测试通常有以下几种分类方法。

按照对被测对象内部实现情况的了解程度，软件测试可分为：

（1）白盒测试。

（2）黑盒测试。

（3）灰盒测试。

按照是否执行被测系统，软件测试可分为：

（1）静态测试。

（2）动态测试。

按照被测内容，软件测试可分为：

（1）功能测试。

（2）非功能性测试（如性能测试、接口测试等）。

按照测试过程逐步推进的角度，软件测试可分为：

（1）单元测试。

（2）组装测试。

（3）确认测试。

（4）系统测试。

（5）验收测试。

1.5 几个容易混淆的概念

下面再介绍几个测试人员容易混淆的概念，包括测试级别、测试类型、测试项等。

1.5.1 测试级别

测试级别也称测试阶段。是按照测试过程逐步推进的方法进行的分类，对比软件开发过程，测试级别包括单元级（针对模块）、集成级（针对部件）、配置项级（针对单个软件配置项）、系统级（针对由多个软件及硬件组成的系统）。

1.5.2 测试类型

测试类型是一种综合性的分类方法，既包括按照被测内容进行的分类，即功能测试、非功能测试（性能、接口测试、人机交互界面测试、强度测试、余量测试、可靠性测试、

安全性测试、恢复性测试、边界测试、数据处理测试等）；又包括按照对被测对象内部了解程度的分类，如逻辑测试（白盒测试）；还包括静态测试的几种不同类型，如文档审查、静态分析、代码审查、代码走查等。

1.5.3 测试项

测试项就是分解后的测试功能项和非功能项，即从测试角度描述的被测软件各功能和性能。

第 2 章　航天器软件工程概况

2.1　航天器软件工程概况

2.1.1　概况

航天器领域的软件工程化工作起步早，为国内其他行业软件工程化的发展起到了重要的引领作用。参照国际标准、国家标准、军用标准，航天领域逐步结合航天型号软件研制特点形成了行业软件标准。各研制单位在此标准体系下结合各自业务活动建立了院标、所标。

2.1.2　载人航天工程的软件工程

在航天软件工程发展中，载人航天工程作为我国航天发展史上规模最大、系统组成最复杂、技术难度最大、安全性最高的一项大型系统工程，对软件工程的探索起步最早，对航天软件工程化的发展起到了促进作用。

目前，航天各型号均制定了软件工程化管理要求及技术文件（标准）。其中软件研制管理要求作为顶层文件，定义了软件研制相关方及其职责，确定了软件研制过程和要求；软件研制技术标准作为管理要求的支撑文件，详细描述了软件研制技术流程、各阶段的工作活动和技术要求，以及测试、评审等工作细则。

2.1.3　航天器软件分级分类管理

航天器软件的一个重要特点是分级分类管理，即把航天器软件按照安全关键等级进行分级管理，按照软件继承性进行分类管理。

2.1.4　软件安全关键等级

按照安全关键级别，可以分为 4 个等级，即 A、B、C、D（安全关键要求从高到低）。其分类依据是软件类别、软件失效后带来的危害以及软件发生失效的概率。

（1）A 级软件是失效可能导致灾难性危害的软件。

（2）B 级软件是失效可能导致严重危害的软件。

（3）C 级软件是失效可能导致轻度危害的软件。

（4）D 级软件是失效可能导致轻微危害的软件。

2.1.5 航天器软件分类

按照软件继承性可以把航天器软件分为Ⅰ类、Ⅱ类、Ⅲ类和Ⅳ类（继承性由高到低），每一类软件的定义及其所采用的不同研制流程参见本书的 2.2 节。后续本书中会多次提到软件安全关键等级和研制类型。简单来说，航天器软件研制实际上就是针对不同安全关键等级、不同类型的软件采用不同研制流程并进行实施的过程。

2.2 航天器软件研制技术流程的划分

一方面为了保证新研软件的质量，要制定研制全过程的技术流程；另一方面为了充分发挥软件重用带来的效率，同时降低软件重用的风险，必须对重用软件的技术状态进行控制和把关，并且根据技术状态制定软件重用的流程。

根据航天器软件研制的特点，以及软件配置项相对于基线配置项的技术状态，航天器软件的研制技术流程共分为 4 类型：

（1）沿用软件。

（2）参数修改软件。

（3）少量功能修改软件。

（4）新研软件。

2.2.1 沿用软件定义

沿用软件也叫做Ⅰ类软件，即已经成功完成飞行试验任务，不加修改即可再次使用的软件配置项。

2.2.2 参数修改软件定义

参数修改软件也叫做Ⅱ类软件，即不更改软件可执行代码的内容，仅修改软件配置参数即可满足任务要求的软件配置项。配置参数通常包括编译时绑定的宏和常量定义，以及固化时写入的配置文件。

2.2.3 少量功能修改软件定义

少量功能修改软件也叫Ⅲ类软件，或称适应性修改软件，指根据任务要求，进行适应性修改、完善设计以及提升关键等级的软件配置项。

2.2.4 新研软件定义

新研软件也叫Ⅳ类软件，即不属于上述 3 类状态的新研制软件配置项。

把航天器软件研制流程进行分类的目的是，对继承性较强的软件可以简化工作过程，从而避免不必要的重复劳动。

2.2.5 新研软件技术流程

针对新研的航天器软件，一般具有如图 2.1 所示的基本研制流程。

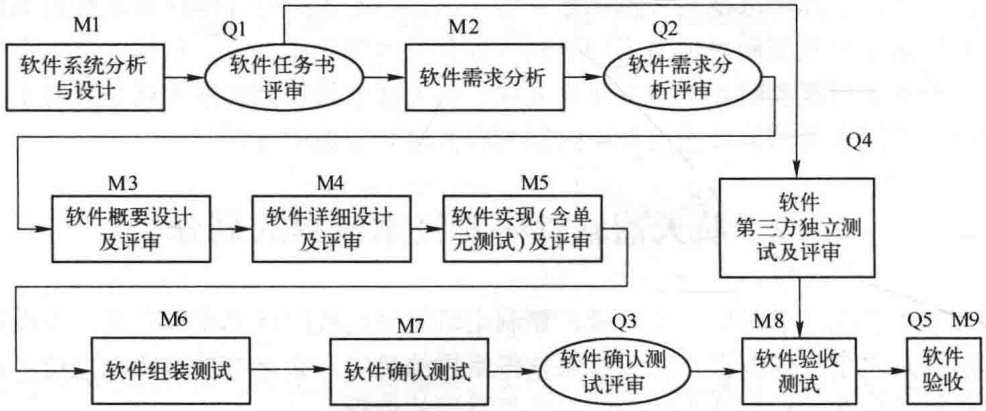

图 2.1 新研软件研制基本流程

新研软件的技术流程是一个完整的型号软件研制技术流程，其他 3 类流程均是在此基础上裁剪而成。

从图 2.1 中可以看出，在软件研制流程中，涉及的测试阶段有单元测试、组装测试、确认测试、第三方独立测试、验收测试。

其中软件验收测试是参照硬件提出的概念，其目的是在软件验收前对软件进行一次完整的测试，以确认软件功能、性能、接口、可靠性、安全性等与软件需求规格说明（任务书）的一致性，针对安全级别要求高的软件还要进行边界测试、强度测试等。由于航天器软件在研制过程中均已进行了各阶段全面测试，一般还要进行第三方独立测试，因此在实际操作过程中验收测试可不再实际进行，而是审查所提交的各阶段测试计划和测试分析报告（包括第三方独立测试的报告），其审查所遵循的要求详见下面。

2.2.6 沿用软件技术流程

对于沿用软件，其简化的研制流程如图 2.2 所示。相比完整的新研软件研制流程，沿用软件技术流程将软件的研制简化为沿用可行性分析和技术状态复核两个工作项目。

图 2.2 沿用（Ⅰ类）软件研制流程

2.2.7 参数修改软件技术流程

对于参数修改软件，其研制技术流程如图 2.3 所示。参数修改的软件原则上不需重新进行软件设计，但是需重新对有代码更动的部分及受影响的部分重新进行影响分析和回归测试。