

Research on the Improvement of Aerospace Software Test Model Based on V Model

Hu Xiaoxi, Niu Jing

Beijing Institute of Spaceflight Control Instruments, Beijing, China

Email address:

tinasabrina@126.com (Hu Xiaoxi), ruomuzi888@163.com (Niu Jing)

To cite this article:

Hu Xiaoxi, Niu Jing. Research on the Improvement of Aerospace Software Test Model Based on V Model. *Science Discovery*.

Vol. 6, No. 2, 2018, pp. 110-115. doi: 10.11648/j.sd.20180602.16

Received: April 20, 2018; **Accepted:** May 17, 2018; **Published:** June 22, 2018

Abstract: The software industry has endured a long and tortuous development process from the beginning to the present, from small scale, low complexity of software development to the large scale of application of software products, software testing plays an important role, the early software debugging has been evolved into software testing now. With the rapid development of space industry in China, the position of software in it has been more and more important, software testing is very significant throughout the software development process, therefore software testing needs more concern. Based on the theory of software testing in software engineering, combined with the practice in work, the current problems in aerospace software testing, how to find existing problems and new testing methods in order to improve the quality of aerospace software. The traditional V model requires that each testing phase could proceed only after the previous stage completed, and there is no backtracking. We cannot adjust and modify the model when problems are found. In order to overcome the shortcomings of traditional V model, the paper puts forward an improved method of aerospace software model based on V model. By iterating over the various stages testings of software development, this method enhances intersections and comprehensiveness, so as to achieve the aim of reducing the number of testing problems and providing effective solutions for the aerospace software testing problems.

Keywords: Test Model, V-Model, V Model Improved

基于V模型的航天软件测试模型改进研究

胡晓曦, 牛静

北京航天控制仪器研究所, 北京, 中国

邮箱

tinasabrina@126.com (胡晓曦), ruomuzi888@163.com (牛静)

摘要: 软件行业从起步到现在经历了一个漫长又曲折的发展过程, 从小规模的、低复杂程度的软件开发到现在大规模的软件产品应用, 软件测试起着举足轻重的作用, 早期的软件调试已经演变为现在的软件测试。随着中国航天事业的飞速发展, 软件在其中的地位愈来愈重要, 软件测试在整个软件开发过程中占有十分重要的位置, 必须给予高度重视。本文基于软件工程中的软件测试理论, 结合工作中的实践, 就当前航天软件测试中的重难点、如何发现存在的问题以及新的测试思路以便提高航天软件质量提出一些观点。传统的V模型要求每个测试阶段必须在完成上一阶段后才能继续进行, 且不允许进行回溯, 当发现问题时, 无法进行调整和修改。为了克服传统V模型的不足, 提出一种基于V模型的航天软件测试模型改进方法, 该方法通过对软件开发周期中各个阶段的测试进行迭代处理, 增强了测试的交叉性和全面性, 从而达到降低测试问题数量的目的, 为航天软件测试问题提供了有效的解决方案。

关键词：测试模型，V模型，改进V模型

1. 引言

为了发展航天事业，建设航天强国，完成历史赋予中国这一代航天人的重要使命，在工作中更要精益求精、精益求精，不能出现半点马虎和纰漏。随着中国航天事业的飞速发展，软件在其中的地位愈发的重要，而随着软件数量的增加、规模的扩大，导致软件的复杂程度也呈现出几何形式放大的特点[1]。

由于人的主观认识常常和客观现实不能完全相符，与工程相关的人员之间的协作也不可能完美无缺。所以，在软件生存周期的各个阶段都可能会发生问题，而且前一阶段发生的软件问题自然会为后一阶段带来相应的问题，因此，软件问题会积累起来[2]。此外，一般情况下，软件经过前一阶段后所产生的结果的细化是后一阶段的工作内容，因此，前一阶段发生的一个软件问题可能会带来后一阶段的几个软件问题，也就是说，软件问题不仅有积累效应，还有放大效应[3]。图1[4]展示了软件开发周期中软件问题的这种特殊效应。

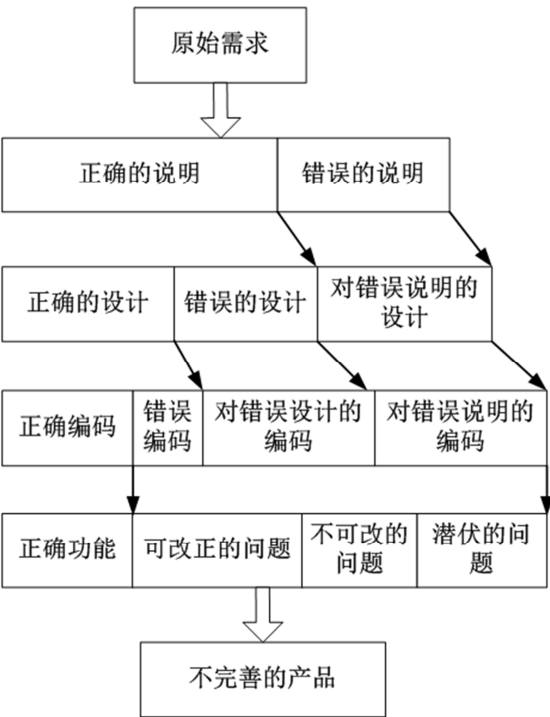


图1 软件问题的积累和放大效应。

由于航天工业的特殊属性，如果软件存在的漏洞在测试阶段没有被完全发现，产生的不可靠、不安全的因素会给运载火箭、空间飞船和空间站带来无法挽回的损失[5]。软件测试可以有效保证软件的质量，细致严谨的软件测试能够发现软件中更多隐藏的bug，减少了软件中遗留的安全问题[6]。据统计，国外软件开发机构花费在软件测试上工作量高达40%，用在软件测试上的开发费用也约为50%。

对于一些要求高可靠性、高安全性的软件，如航天系统内的各种软件，测试费用可能会是软件周期中所有其他阶段花费的3到5倍[7]。因此，要想开发出高质量低隐患的软件产品，一定要提高软件测试工作的重视程度，也因此系统有效、全面充分的软件测试工作也逐渐成为航天软件研制各个阶段中必不可少的组成部分。

本文基于软件工程中的软件测试理论，结合工作中的实践，就当前航天软件测试中的重难点、如何发现存在的问题以及新的测试思路以便提高航天软件质量提出一些观点，形成一套行之有效的航天软件测试问题解决方案[8]。

2. 航天软件的特点

从当前中国航天软件发展的现状来看，航天软件主要有以下特点：

(1) 航天软件的类型多为嵌入式软件，软件的代码与所用的航天仪器元器件结合紧密。航天计算机都是专业定制的嵌入式计算机，生活中常用的显示器、键盘和打印机等输入输出设备被大量的传感器、控制信号接口和数据采集接口所替代[9]。同一段软件代码，在同样功能但不同型号的元器件上仍可能出现漏洞。

(2) 航天软件对可靠性和安全性的要求极为严格。一个微小的漏洞隐患，在运行环境下一旦对控制系统造成影响，产生的严重后果是无法估量的。例如，1963年，美国一艘装载探测器飞往火星的运载火箭，由于程序员将Fortran程序中DO 5 I=1,3误写为DO 5 I=1.3，只是“;”误写成“.”，导致火箭升空爆炸，损失达1000多万美元[10]。

(3) 航天软件有非常苛刻的实时性要求。在航天仪器工作过程中，控制系统必须实时完成各项工作，不但要保证所采集的数据完整、真实和有效，而且要能够及时对各种突发情况进行加工处理[11]，不能出现任何延迟。发送和接收的控制信号有严格的时序关系，任何微秒级别误差都会导致控制信号的混乱。

(4) 航天软件复杂的结构、广阔的应用领域给航天软件的测试带来重重的困难。航天仪器所处的环境跟地面的办公室有着天壤之别，超长的工作时间、外太空的辐射射线、高温高压的设备舱、气象条件恶劣等等[12]，都是航天软件在设计时需要考虑的因素，其应用范围和复杂程度远非正常应用软件所能比拟。

3. 航天软件测试面临的难点

上述航天软件的4个特点为软件测试人员在测试过程中带来了极大的困难。众所周知，航天软件的测试过程十分复杂，要反复经历重复迭代的过程。航天软件的研制过程分为模样、初样、试样、正样等多个研制阶段[13]。在每个研制阶段都要进行单元测试、集成测试、系统测试、确认测试，有时还常常需要进行第三方测试。由于其需要反复重复迭代的特殊性，测试过程会重复很多次。而且在

测试过程中,尤其在模样和初样阶段,用户需求经常发生变化。软件测试的频繁回归过程中将会产生大量的测试用例,这样就会出现很多无效的测试用例[14],这为整个测试过程和软件开发周期的工程化管理带来很大的不便:

(1) 需要很长的周期来培养同时精通航天专业和软件工程专业复合型人才。从事软件测试的工程师绝大多数是计算机或软件工程专业,要较好地完成航天软件测试工作,除了运用本专业的知识外,还要熟练掌握与航天技术相关的各项专业知识,如航天器设计、流体力学和自动控制等。此外,在搭建软件测试平台的过程中,软件测试工程师还要有较强的计算机硬件知识,以便能够提出相应的测试需求和设计测试用例。但从目前看,能够将几项专业融合起来的复合人才培养周期过长,难以在短时间内形成大批骨干力量。

(2) 市场上常见的测试工具在很多方面并不能满足航天软件充分测试的需求。例如, 航天软件对实时性要求较高, 且不能改变接收和发送数据的时序顺序, 但现有的测试工具在代码插桩环节可能会改变时序关系, 这样的测试用例就毫无用处。

(3) 航天软件的迭代周期短, 因此测试的工作量极大, 同时对测试的质量要求非常高。由于很多航天软件不能完全依靠已有的测试工具进行测试, 只能通过的方法来进行人为的排故操作。人工代码走查面对的是海量的代码和测试文档, 而且还会面临不熟悉软件设计背景, 不了解设计思想, 甚至文档的不完整等各种问题, 但是同时还要求测试工程师要读懂、读熟、读透代码, 更要能火眼金睛地查出问题。因此, 航天软件测试要求测试工程师不但要

熟练地掌握和运用相关的专业技能，更要有过人的耐心和毅力。

4. 基于V模型的航天软件测试模型改进

软件测试模型是软件测试工作的框架，它描述了软件测试过程所包含的主要活动，以及这些活动之间的相互关系等。目前主要的软件测试模型有V模型、W模型和X模型等，但随着软件测试技术的发展，以上几种传统测试模型开始逐渐显露出各种局限性，单纯地使用传统测试模型已经不能完全满足现在软件测试的需要。

4.1. 传统V模型

传统V模型由软件开发瀑布模型变化而来，主要反映测试各阶段与软件开发各阶段的对应关系，从左至右，展示了最基本的软件开发和软件测试过程[15]。传统V模型同时也被称为RAD（Rap Application Development，快速应用开发）模型，而且由于其模型的图示很像字母V，所以又被人们称软件测试的V模型[16]。它通过同时进行软件开发和测试工作，从而达到缩短整个软件开发周期、提高软件开发效率的目的。可以看出，V模型包括了低层测试和高层测试，低层测试保证了源代码的正确性，高层测试确保软件与用户提出的需求相符[17]。

V模型的设计思路是将测试过程划分为数个阶段，在每个阶段分别进行不同的测试工作。如下图所示，沿着左边部分下降的是软件开发的各个阶段，而沿着右边部分上升的是相应的软件各个阶段，很明显，软件开发各阶段与软件测试各部分是相互对应的关系。

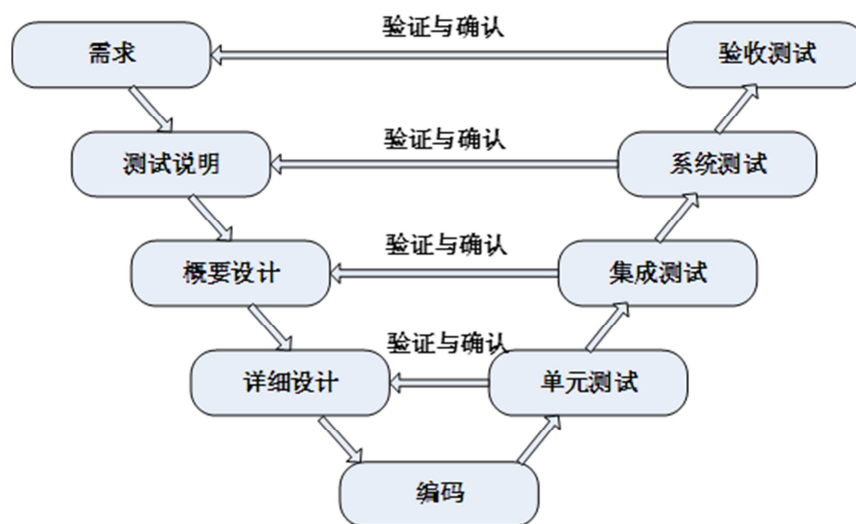


图2 V测试模型。

如上图所示，传统V模型分为如下阶段：与软件开发过程相关的需求分析、概要设计、详细设计、软件编码，以及软件测试过程中的单元测试、集成测试、系统测试、验收测试等必要的测试过程。

需求分析：在开启整个软件开发周期之前，首先要明确客户真正的需求，对最终的软件产品有什么具体要求，软件需要实现哪些功能，这就要求软件分析师和客户进行

彻底的交流沟通，做到完全理解客户提出的要求。为了后续的软件开发工作，软件分析师要使分析结果形成一份涵盖客户所有需求的需求规格说明书。

概要设计：在这个阶段形成软件的主体架构，主要包括架构的搭建、各软件模块实现功能的描述、各模块接口之间如何连接以及数据流的处理等。

详细设计: 对概要设计阶段中的各个模块进行进一步的细化和分析, 同时还要对这些模块的不同组合进行分析, 在软件开发的详细设计阶段要形成伪代码或达到同等程度的级别, 具体详细地描述程序功能的具体实现方法及现象。其中需要包含数据库设计说明。

软件编码: 根据详细设计中具体描述的软件模块, 选择合适的编程语言用代码实现相应的功能。

单元测试: 选取相应的最小测试单元来进行单元测试, 这个测试阶段主要是对软件程序代码进行测试, 确保软件各单元做到被正确的编译。不同的单位和不同的软件对单元的划分不尽相同, 例如可以具体到模块, 也可以具体到类或函数等。

集成测试: 在单元测试之后, 把单元组合成系统, 对组合后的功能进行测试, 同时要测试模块接口、数据流是否出现问题等, 集成测试的主要目的是确保各单元模块之间的接口正确性。按照集成测试步骤, 组合软件的单元或模块, 之后运行单元或模块组合后的系统来测试是否出现软件问题。

系统测试: 在对最小软件单元以及单元模块的组合进行充分的测试后, 真正搭建软件系统, 根据软件的需求规

格说明, 测试软件的功能、性能、安全性等是否符合用户的需求, 并且分析这时运行的系统是否有漏洞。

验收测试: 在这个测试阶段, 软件已经交付给用户, 用户持有软件, 在实际的使用环境中一句需求规格说明书和任务书设计相关的测试用例, 以确保软件达到预期效果。

图2中可以看出, 与单元测试相对应的是详细设计阶段, 即设计单元测试的测试用例是和软件开发的详细设计是同时进行的, 也就是说, 在软件开发人员做详细设计的同时, 相关的软件测试人员要完成这个软件的测试用例; 集成测试与概要设计相对应, 在分析模块的具体功能和接口以及数据流的时候, 要同时完成集成测试阶段的测试用例, 这样在后续真正进行集成测试时可以直接使用这些测试用例; 系统测试本身就是由需求分析产生的, 在分析人员分析系统的时候, 测试人员要一句客户的需求规格说明书, 完成系统功能相应的测试用例, 为最后的系统测试做准备。最后的验收测试是一句需求来测试, 所以与用户需求相对应。图3展示了软件测试的4个步骤: 单元测试、集成测试、系统测试和验收测试。

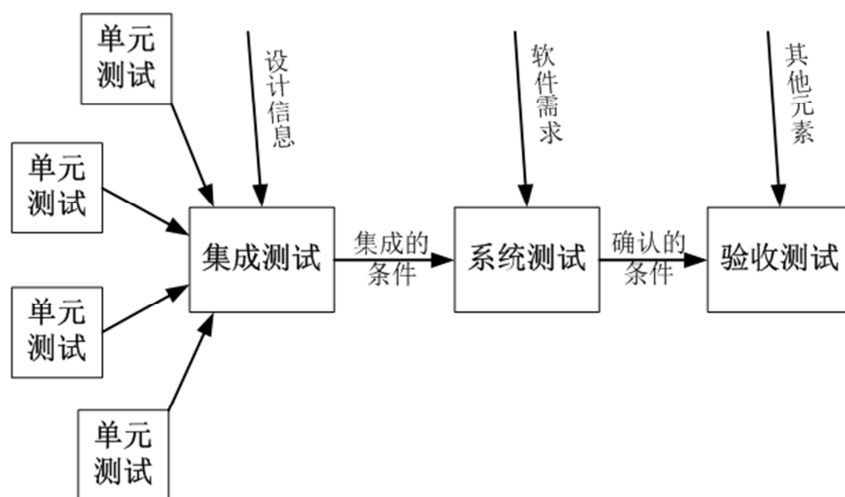


图3 软件测试过程。

软件需求提出和可行性研究阶段要与项目验收测试对应, 验证项目各版块是否完成相应功能; 软件需求规格说明和软件验收测试阶段相对应, 验证软件内部外部接口及各项性能指标是否符合规格说明书的要求; 概要设计和详细设计与单元测试和集成测试一一对应, 验证代码是否与设计思路有偏差。V模型要求每个测试阶段必须在完成上一阶段后才能继续进行, 且不允许进行回溯, 当发现问题时, 无法进行调整和修改。

从传统V模型中可以明确看到软件开发到软件测试过程中的不同阶段, 右侧上升部分的单元测试、集成测试、系统测试和验收测试, 与每一个开发级别一一对应。实践表明传统V模式比较适用于部分传统信息系统应用的开发, 但是存在一些存在高性能高风险的系统或互联网软件, 或者很难做到具体模块化的系统, 就很难符合传统V模式展示的各个具体软件开发和软件测试阶段, 这时就需要更具有迭代特性的开发模型。

传统V模型只是把测试过程作为在软件开发各阶段之后的阶段, 忽视了软件测试需要反复对需求、系统设计进行验证, 是否真正符合用户的需求只能在最后的验收测试阶段才能被验证。在软件开发完成后才去考虑软件测试过程把, 前期的需求分析产生的错误只能在软件开发周期的最后才能被发现。传统的V模型认为测试的对象只是软件代码, 没有对需求和具体的软件设计进行及时的测试。传统的V模型忽视了代码形成前的测试, 不符合“尽早地和不断地进行软件测试”原则[18]。

由此可知, 该模型的缺陷是非常明显的。它将软件测试工作局限成为一种在编码开发完成后所进行的接续工作。其目标是查找代码在编码过程中的语法错误。这样做会带来在软件设计初期的可行性和需求分析、概要设计等阶段产生的逻辑错误无法被发现或只能在验收阶段或用户使用一段时间后才能被发现的隐患[19]。基于传统V模型的这个缺陷, 本文提供一种解决思路, 当一个软件开发

的时候,研发人员和测试人员经过协商,同时工作,反复回归迭代,在对软件做需求分析同时完成测试用例的设计,这做到及时高效地发现程序和文档中的错误,从而提高程序的质量,最大程度的减少成本,满足用户的实际需求。

4.2. 改进V模型

如上所述,传统V模型将软件测试工作局限成为一种在编码开发完成后所进行的接续工作。其目标是查找代码在编码过程中的语法错误。为整个软件开发周期带来了在

软件设计初期的可行性和需求分析、概要设计等阶段产生的逻辑错误无法被发现或只能在验收阶段或用户使用一段时间后才能被发现的隐患。可见,传统的V模型要求每个测试阶段必须在完成上一阶段后才能继续进行,且不允许进行回溯,当发现问题时,无法进行调整和修改。

依据“越早发现错误,修复的成本越低”的原则,考虑设计一种改进的V模型来解决上述问题。改进的V模型如下图所示:

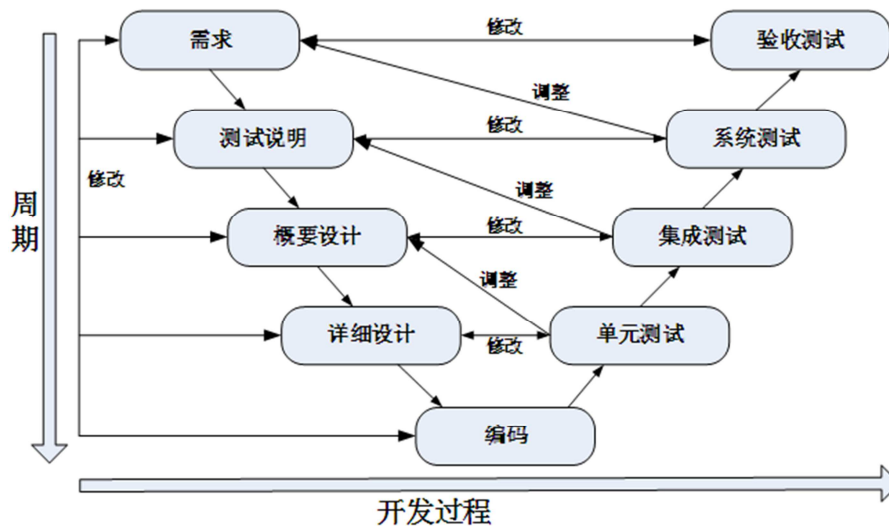


图4 改进的V测试模型。

在新的改进版本中,和传统V模型相比,最大的区别在于没有一个阶段是相互独立的。可以从图中看出,每个阶段之间都是相互交互、协作的关系。每一个软件开发阶段,在发现问题后,能够及时调整和修改各自的开发活动,这样在一定程度上就能够并行开展工作。

单元测试阶段,如果检查出软件存在问题,就可以迅速追溯到详细设计阶段,根据单元测试来修改详细设计,并同时调整概要设计,以防出现本文之前提到的软件问题积累和放大的问题。同样,在集成测试阶段,依据测试中出现的软件问题,修改概要设计的同时,调整测试说明,做到尽可能同时修改所有涉及到的内容,从而减少后期的测试迭代工作,降低软件开发成本,提高测试效率。

设计阶段的工作可以逐步转移到单元测试用例的设计中来;编码阶段并行执行单元测试用例,同时对已集成的部分软件模块进行集成测试。在改进的V模型中,执行测试后将错误及时反馈给开发人员,然后测试工程师根据软件的改正情况,修改或复用前一版本的测试用例。这样可以实现在后一阶段的软件开发代码发生变更时,及时修改和完善前一阶段的测试用例以便对软件进行更充分更完善的测试。改进V模型的优点主要体现软件版本发生变化时,软件测试与开发活动之间进行的不间断交互迭代。例如并行执行制定测试计划、设计测试用例以及软件开发过程中的需求分析、详细设计和代码编写等工作,提高了测试的工作效率,对软件代码的缺陷分析更有针对性。由于这个模型中需要不断地对已有测试用例进行复用与修

改,因此要配合软件工程化管理,严格控制软件代码和相应测试文档的版本。

4.3. 应用实例分析

针对同一开发项目,分别根据传统V模型和本文中改进的V模型来进行代码缺陷的分析整理。本文摘取开发项目中一部分模块的静态代码,通过Klocwork测试软件[20]对这部分模块代码进行缺陷分析检查。

使用Bat批处理脚本和Shell脚本后的缺陷分析结果如下图所示:

缺陷级别	缺陷类型	测试版本	
		V模型	改进V模型
致命级	数组越界	8	5
	内存泄露	3	1
	判断浮点数是否相等	4	4
	资源泄露	5	3
重要级	数据流异常	2	1
	空指针引用	25	11
	冗余代码	18	13
	变量未初始化	7	5
一般级	Else If语句分支不全	12	10
	Switch 语句缺少	0	0
	Break		
	合计	84	53

图5 缺陷分析结果。

在整个软件开发周期中,通过静态分析进行代码检查远比动态测试更有效率,能够通过词法分析、语法分析、

控制流分析等技术对程序进行扫描,验证代码是否满足规范性、安全性、可靠性、可维护性等指标的代码分析,快速分析对软件架构、编程规则的违反情况,找到软件质量缺陷、安全漏洞方面的缺陷,如果操作得当,可以发现30%-70%的代码设计和缺陷,而且这种代码检查看到的是问题本身而非征兆[21]。

对代码缺陷检查的结果进行对比,可以发现,通过列举的缺陷类型与种类,改进后的V模型的Bug数量明显少于依据传统V模型运作的代码测试结果统计。而这些是我们采取静态分析代码检查方法时想要得到的结果,因为这些问题不仅会为软件的易读性、维护性等方面带来大量问题,特别情况下,还可能导致出现程序死机、异常退出等严重问题。我们知道,通过Klocwork软件所发现的缺陷大多是比较隐蔽、容易被忽略的问题,如数组越界、资源泄漏等,它们会带来不可预估的损失和隐患。

本文对传统V模型的改进可以有效帮助开发人员、质量保证人员查找代码中存在的结构性错误、安全漏洞和代码缺陷等问题,从而保证软件的整体质量。改进的V模型同时能够在代码开发的全周期协助开发人员优化代码,缩短项目周期,降低研发成本,提高代码质量,优化软件工程化管理。

5. 结论

本文结合航天软件的特点,基于软件工程中的软件测试理论,结合工作中的实践,就当前航天软件测试中的重难点、如何发现存在的问题以及新的测试思路以便提高航天软件质量提出一种基于V模型的航天软件测试模型改进方法,克服了传统的V模型要求每个测试阶段必须在完成上一阶段后才能继续进行,且不允许进行回溯,当发现问题时,无法进行调整和修改的缺陷,通过对软件开发周期中各个阶段的测试进行迭代处理,增强了测试的交叉性和全面性,从而达到降低测试问题数量的目的,全文从实际问题出发进而反推解决办法,最后将该方法应用于某航天软件的测试过程中,实验表明新方法使测试过程中发现的问题数量减少了大约36.9%,达到了预期的效果。

参考文献

- [1] 张新华,何永前.软件测试方法概述[J].科技视界,2012(04):35-37。
- [2] 习颖慧.航天软件测试技术的研究[J].第二十六届全国空间探测学术研讨会,2013年10月。
- [3] 王蓁蓁.软件测试理论初步框架[J].计算机科学,2014,41(03):12-16+35。
- [4] 谷潇.计算机软件安全及其防范的研究[J].信息记录材料,2018,19(04):56-58。
- [5] 滕俊元,徐忠宾,高猛.软件产品化在航天领域的应用与管理[J].质量与可靠性,2018(01):39-42。
- [6] 单锦辉,姜瑛,孙萍.软件测试研究进展[J].北京大学学报(自然科学版),2005(01):134-145。
- [7] 顾吉堂.基于失效数据的软件可靠性测试方法[J].电脑知识与技术,2018,14(01):122-123。
- [8] 王强,田涛,刘昕昀.软件测试能力评定模型研究[J/OL].计算机技术与发展,2018(08):1-7[2018-05-04].<http://kns.cnki.net/kcms/detail/61.1450.TP.20180427.1644.074.html>。
- [9] 陈银平,王荣.第三方软件测试流程以及实施中存在的问题和建议[J].中国管理信息化,2018,21(08):71-72。
- [10] 孙春生,卫翔,王金树.软件测试在软件质量管理中的应用分析[J].电子技术与软件工程,2015(12):69-70。
- [11] 刘宇轩.软件测试方法研究[J].科技风,2018(04):53。
- [12] 蒲冬梅,孙虎.如何做基于需求的测试[J].电子技术与软件工程,2018(01):38。
- [13] 肖文涛.自动测试系统软件技术的通用性分析[J].通讯世界,2017(24):333-334。
- [14] 冯亚娜,刘星妍.基于移动平台的APP软件测试[J].信息技术,2015(04):206-207+210。
- [15] 王象刚.软件测试过程管理工具的设计与实现[J].软件,2014,35(02):96-97。
- [16] 哈清华,刘大有,沈湘衡,刘遵.基于需求模型的航天软件测试用例生成方法[J].光学精密工程,2016,24(05):1185-1196。
- [17] 孙莹,王华伟.软件测试中存在的问题及对策[J].软件导刊,2015,14(01):50-51。
- [18] 林永峰,陈亮.面向安全性分析的嵌入式软件测试方法研究[J].现代电子技术,2016,39(13):80-83。
- [19] 李霁.软件测试中选择有效测试技术的一种方法[J].科技通报,2015,31(01):217-220+224。
- [20] Klocwork 测试软件[Z].<https://wenku.baidu.com/view/9468ef0069eae009581beca9.html>。
- [21] 朱晓敏.软件测试的相关技术应用研究[J].电子测试,2017(01):122-123。