文章编号:1671-4598(2019)10-0036-05 DOI:10.16526/j.cnki.11—4762/tp.2019.10.008 中图分类号:TP311.5 文献标识码:A

航天嵌入式软件测试用例典型设计缺陷研究

左万娟[1,2],虞砺琨[1,2],王小丽[1,2],黄 晨[1,2]
(1.北京控制工程研究所,北京 100190;2.北京轩宇信息技术有限公司,北京 100190)

摘要:作为动态测试充分性的基本评价指标,覆盖率分析只能帮助修正因输入不足而导致的测试用例设计缺陷;针对航天嵌入式软件测试过程中不影响覆盖率统计结果的用例设计缺陷,从测试步骤和预期结果两大测试用例核心要素开展研究,提出十个典型缺陷,分别予以分析,并进行缺陷修正;工程实践证明,这些缺陷的发生率高,具有典型性;修正这些缺陷后,可以有效检出软件设计缺陷;与用例执行后的覆盖率统计数据分析相结合,可以有效提高测试充分性。

关键词:测试用例;典型;设计缺陷;覆盖率

Typical Test Cases Design Faults Research of Aerospace Embedded Software

Zuo Wanjuan[1,2], Yu Likun[1,2], Wang Xiaoli[1,2], Huang Chen[1,2]
(1. Beijing Institute of Control Engineering, Beijing 100190, China;
2. Beijing Sunwise Information Technology Co. Ltd., Beijing 100190, China)

Abstract: As a essential evaluation criteria, coverage analysis can only help to revise the test case design faults due to input deficiency. The research is aimed at test case design faults in the aerospace embedded software testing which do not influence coverage. From the two core elements, test steps and anticipations, ten typical faults are proposed, analyzed, and revised. The evaluation results show that these faults are typical because of high happen rate. After the revision, software faults can be found effectively, and test adequacy can be improved through the connection with the coverage analysis.

Keywords: test case; typical; design fault; coverage

0 引言

随着航天器在轨设计寿命的不断延长,可靠性要求越来越高,软件设计在满足航天器在轨运行基本功能需求基础之上,还要承担硬件的故障诊断与重构功能,软件规模不断增大、复杂度不断提高,对软件质量的要求也日益严峻。

软件测试是保证软件质量的重要技术手段,按照是否运行被测软件,分为静态测试和动态测试,在航天工程实践中,二者缺一不可、相辅相成。测试用例设计是动态测试有效性和充分性的根本保障。

针对测试用例设计,目前研究方向多集中于测试用例自动生成技术和测试用例集约简技术,主要包括基于遗传算法的测试用例自动生成[1-3]、基于蚁群算法的测试用例自动生成[4]、基于组合测试算法的测试用例自动生成[5]、基于符号执行的测试用例自动生成[6]、基于专家系统的测试用例自动生成[7]等等。测试用例集约简技术[8-9]则主要着力于解决测试效率问题。

以上研究以及工程实践中,通常以覆盖率作为动态测试充分性的基本评价指标。但是,通过长期的工程实践研究,发现部分测试用例虽然存在设计缺陷、导致软件设计缺陷未检出,但是并不妨碍覆盖率统计结果,即,存在覆盖率分析无法发现的测试用例设计缺陷,而业界对此却鲜有研究。因此,针对覆盖率分析无法发现的测试用例设计缺陷开展专题研究,具有非常重要的工程实用价值。

收稿日期:2019-03-24; **修回日期**:2019-04-16。

作者简介:左万娟(1971-),女,四川人,硕士,高级工程师,主要从事嵌入式软件测试方向的研究。

1 名词解释

对本文用到的名词解释如下:

遥测:地面观察航天器在轨运行状态的一种手段。通过特定的天地通路,将航天器在轨运行状态传送回地面,供地面观察、分析。

三取二:航天器常见的可靠性安全性设计手段。对重要数据分三区保存,使用前对三区内容进行比对,任意两区一致方可使用。

物理输出:特指软件的外部接口输出,通常表现为指令码、控制脉冲等等。

最小指令间隔:指相同指令或不同指令之间的间隔最小值。一般应作为测试约束条件,并在测试用例设计时按照最小指令间隔设置指令输入,从而验证软件对指令响应的及时性。

2 测试用例典型设计缺陷

测试步骤和预期结果是测试用例设计的两大核心要素。测试步骤主要表现为软件各种输入及其组合,预期结果主要表现为软件输出。测试步骤和预期结果的设计缺陷,将对测试有效性和充分性造成不良影响。

2.1 指定地址范围的操作覆盖性缺乏验证

以航天嵌入式软件常见的 RAM 区自检功能需求为例,

对此项典型缺陷进行说明。

2.1.1　需求概述

采用写入与读出值比对的方式，对 RAM 区指定地址范围进行自检。

2.1.2　用例设计缺陷分析

用例设计时，考虑写入与读出值比对一致和不一致的情况，执行测试，并通过观察相应的遥测来确认软件实现的正确性。

用例执行后，通过覆盖率分析确认，自检相关的语句和分支均已覆盖。即，从覆盖率分析的角度而言，用例设计不存在缺陷。

但是，进一步分析可以发现，以上用例无法检出如图 1 所示的软件设计缺陷。

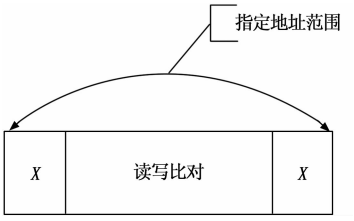

图 1　读写比对未覆盖指定地址范围缺陷示例图

图中，*X* 表示未做读写比对。

如图 2 所示，该用例设计无法检出以下两个软件设计缺陷：

1）对指定地址范围未覆盖完全，仅对其中的部分地址范围进行了自检。

2）自检操作范围超出指定地址范围。

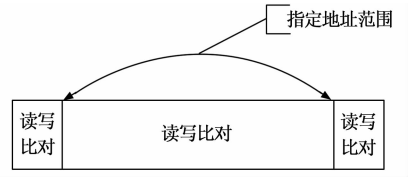

图 2　读写比对超出指定地址范围缺陷示例图

因此，该用例设计存在缺陷。

2.1.3　用例设计缺陷修正

针对上述用例设计缺陷，修正用例设计：考虑写入与读出值比对一致和不一致的情况，考虑比对不一致的地址为指定地址范围的首地址－1、首地址、中间地址、尾地址、尾地址＋1 的情况，执行测试，并通过观察相应的遥测来确认软件实现的正确性。

用例修正后，图 1 和图 2 所示的软件设计缺陷可以被检出，证明用例修正有效。

2.1.4　应用说明

此类用例设计缺陷，较易发生的测试场景如下：

1）指定地址范围自检功能测试。

2）区域性三取二测试。

2.2　通过修改局部变量值实施故障注入、覆盖故障分支

以航天嵌入式软件常见的寄存器校验功能需求为例，其故障分支，一般只能通过故障注入的方式才能够激励运行。

2.2.1　需求概述

对指定寄存器进行校验，若寄存器值不为指定值，则重置寄存器。

2.2.2　用例设计缺陷分析

测试用例设计中常见的设计缺陷如图 3 所示。

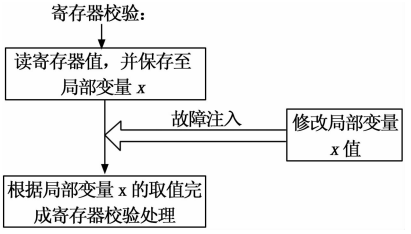

图 3　故障注入用例设计缺陷示例图

如图 3 所示，用例设计时，通过修改局部变量值的方式实施故障注入，存在设计缺陷。此类故障注入手段将导致寄存器校验功能验证不充分，软件是否校验了指定寄存器，未验证到。

另外，通过用例执行后的覆盖率分析确认，寄存器校验相关的语句和分支均已覆盖。即，覆盖率分析无法发现此类用例设计缺陷。

2.2.3　用例设计缺陷修正

修正后的用例设计如图 4 所示。

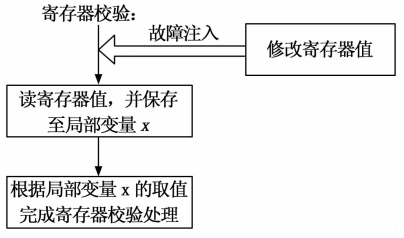

图 4　修正后的故障注入用例设计示例图

如图 4 所示，在软件读寄存器之前，通过修改寄存器值实施故障注入。修正后的用例设计可以检出未读指定寄存器（即，读错寄存器）的代码设计缺陷。

2.2.4　应用说明

当软件中存在外部接口的正常、异常输入均无法激励运行的语句和分支，必须采取特殊的故障注入手段执行测试时，较易发生此类用例设计缺陷。测试场景多见于：

1）寄存器校验功能测试。

2）三取二测试。

3）指定地址范围自检测试。

2.2.5　故障注入注意事项

结合该典型缺陷分析，总结激励故障分支运行的各种故障注入手段的优先级依次如下：

1）外部接口输入流。（最高优先级）

2）修改外部 RAM 指定地址内容、或寄存器值。

3）修改全局变量值。

4）修改局部变量值。（最低优先级）

注意，通过修改局部变量值实施故障注入的优先级最低！仅当排除所有高优先级手段的可行性之后方可使用。

2.3 连续和累计计时考虑不全面

以航天嵌入式软件常见的总线无通讯故障处理需求为例，对此项典型缺陷进行说明。

2.3.1 需求概述

连续 16 s 无 CAN 总线通讯，则重新初始化 CAN 总线。

2.3.2 用例设计缺陷分析

用例设计时，考虑连续 20 s 无 CAN 总线通讯的情况，执行测试。测试结果表明，软件在连续无通讯 16 s 时对 CAN 总线进行了重新初始化，测试通过。另外，覆盖率分析确认，相关语句和分支均已覆盖。

但是，如果软件设计存在缺陷，将连续计时需求设计成累计计时，则用例运行结果如图 5 所示。

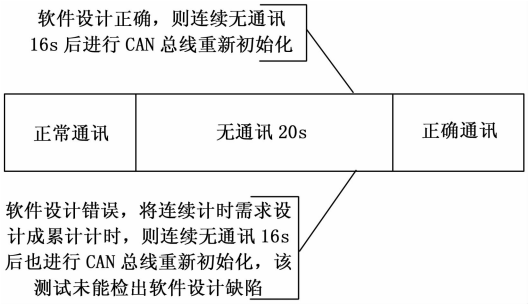

图 5 考虑连续计时工况时的运行结果示例图

如图 5 所示，针对连续计时需求，用例设计时，如果仅考虑连续计时情况，则无法检出将连续计时需求设计成累计计时的软件设计缺陷。

2.3.3 用例设计缺陷修正

对缺陷用例进行修正，分别考虑连续 16 s 和累计 16 s 无通讯的情况。则累计无通讯情况下的运行结果如下：

如图 6 所示，用例修正后，将连续计时需求设计成累计计时的软件设计缺陷可以被检出，证明修正有效。

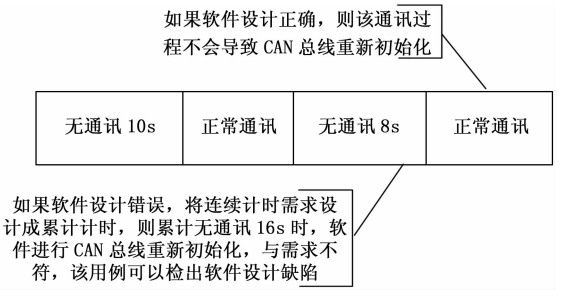

图 6 考虑累计计时工况时的运行结果示例图

2.3.4 应用说明

在所有含连续计时需求的用例设计中，均应排除此类用例设计缺陷。

2.4 未执行正确与错误输入的穿插测试

错误格式指令测试，是软件接口测试的基本测试要素。GJB2725A 军用软件评测实验室测评过程和技术能力要求 5.1.2.9 条款中明确规定：对每个外部输入/输出接口必须做正常和异常情况的测试[10]。

测试过程中，设置错误输入，执行错误格式指令测试，首先是为了验证软件对错误输入的屏蔽情况，即，软件不应响应错误输入；其次是为了验证软件对错误输入的处理是否会影响其后续对正确输入的正常响应处理。因此，必须执行正确与错误输入的穿插测试。

此类用例设计缺陷，一般表现为如下两种形式：

1）连续执行错误输入步骤。

2）以错误输入作为用例的最后一个测试步骤。

上述用例设计缺陷，均应予以修正，以确保测试有效性。

2.5 连续若干个测试步骤的预期结果一样

当连续若干个测试步骤的预期结果（或其中部分关键输出的预期结果）一样时，无法确认连续步骤中第一个步骤之外的后续步骤是否正确执行，因为同样的执行结果可能意味着软件并未正确响应后续测试步骤，而仅仅是维持着与之前测试步骤相同的执行结果，从而导致软件未能正确响应后续测试步骤中指定输入的缺陷无法检出。

因此，在设计测试步骤时，应使每个步骤的预期结果均不相同，以充分确认软件对当前测试步骤响应的正确性，从而确保测试验证的有效性。

需要注意的是，初始运行状态，即，测试用例的初始设置，与测试步骤 1 之间，也属于连续步骤，即，测试步骤 1 的预期结果应与初始运行状态有所不同。

2.6 最小指令间隔测试不足

执行最小指令间隔测试，是为了验证软件对指令响应及处理的及时性。最小指令间隔测试不足缺陷，含如下两种情况：

1）测试步骤未考虑最小指令间隔情况。

2）预期结果仅观察最后一条指令的执行情况，或者仅观察指令计数情况，未观察每条指令的执行结果。

最小指令间隔测试不足，将导致软件响应及处理不及时缺陷无法检出。

针对最小指令间隔测试，测试步骤设计应考虑如下两种情况：

1）相同指令的最小间隔。

2）不同指令的最小间隔。

最小指令间隔测试的预期结果设计，务必包含每条指令的执行结果，这是最小指令间隔测试的核心意义所在。

2.7 预期结果错误

用例设计过程中，预期结果错误的原因有如下两种：

1）测试人员疏忽，对预期结果未考虑清楚。

2）测试人员对需求的理解有误。

预期结果错误，在用例执行时将面临如下两种情况：

1）软件设计正确，导致执行结果与预期结果不一致。通过对结果的分析、确认，发现预期结果设计错误，更正预期结果后，用例执行通过。

表 1　测试用例典型设计缺陷汇总表

序号	测试用例典型设计缺陷	缺陷位置	缺陷性质	需求相关性
1	指定地址范围的操作覆盖性缺乏验证	测试步骤	输入组合不足	相关
2	通过修改局部变量值实施故障注入、覆盖故障分支	测试步骤	输入手段不合理	相关
3	连续和累计计时考虑不全面	测试步骤	输入组合不足	相关
4	未执行正确与错误输入的穿插测试	测试步骤	输入组合不合理	—
5	连续若干个测试步骤的预期结果一样	测试步骤/预期结果	输入组合不合理	—
6	最小指令间隔测试不足	测试步骤/预期结果	输入组合不足	—
7	预期结果错误	预期结果	预期结果错误	—
8	预期结果仅观察遥测、未观察物理输出	预期结果	预期结果不足	—
9	预期结果仅确认了合法输出、未确认非法输出	预期结果	预期结果不足	—
10	预期结果无依据	预期结果	预期结果不合理	—

注：需求相关性，用于定义该项用例设计缺陷是否与特定需求相关。相关则说明该项缺陷仅当软件有特定需求时才会发生。

2）软件设计错误，导致执行结果与预期结果一致，用例执行通过，最终导致软件缺陷未检出。

可见，无论是何种原因所导致的预期结果错误，均应及早予以修正。在确保预期结果正确的前提下，执行测试用例，否则，可能因“负负得正”而导致软件缺陷无法检出。

2.8　预期结果仅观察遥测、未观察物理输出

针对航天嵌入式软件而言，物理输出才是其核心功能需求，是确保航天器在轨正常运行的根本，而遥测则是供地面观察和分析航天器在轨运行状态的记录。预期结果设计时不能顾此失彼，应同时确认物理输出和遥测，既要确保航天器在轨运行状态的正确性，又要确保地面可以准确获知航天器在轨运行状态。预期结果中任一方面的缺失，都可能导致软件相关设计缺陷无法检出。

2.9　预期结果仅确认了合法输出、未确认非法输出

软件设计，在正确响应外部输入产生合法输出的同时，可能也产生了非法输出。如果预期结果中仅确认了软件合法输出，对是否产生了非法输出未予确认，则导致软件产生了非法输出的缺陷无法检出。

因此，预期结果中，不仅要确认软件响应当前测试步骤时所应当有的输出，还要确认软件响应当前测试步骤时不应当有的输出，即，确认软件无非法输出。例如，针对应无计数累加、无控制输出、无应答等的测试步骤，预期结果中应予以明确，以确保测试执行时对非法输出予以确认，确保测试有效性和充分性。

2.10　预期结果无依据

预期结果无依据，指预期结果中含需求未明确规定的软件输出。针对这种情况，应确认是预期结果设计缺陷，还是需求分析缺陷，并进行相应的修正。

总之，不能以无依据的软件输出作为用例的预期结果。

2.11　小结

作为动态测试充分性的通用评价指标，覆盖率分析可以指出测试未覆盖的语句和分支，供测试人员分析，从而找出测试用例中因输入不足而导致的设计缺陷，并据此补充、完善测试用例设计，提高测试覆盖率。但是，覆盖率分析并非万能，针对覆盖率分析所指出的测试已覆盖的部分，也并不能简单地认定其测试充分性。真正的测试充分性，归根结底，要通过完善的测试用例设计来保障。测试用例设计过程中，尤其要注意消除那些不影响覆盖率分析结果的潜在缺陷。综合上述研究，对具有潜在性（即，不影响覆盖率分析结果）的测试用例典型设计缺陷汇总如表 1 所示。

3　应用分析

3.1　统计数据

针对上述测试用例典型设计缺陷，随机抽取了 3 个测试项目开展同行评审，评审中发现的典型缺陷统计如表 2 所示。

表 2　同行评审数据统计表

序号	测试用例典型设计缺陷	缺陷用例个数		
		项目 a	项目 b	项目 c
1	指定地址范围的操作覆盖性缺乏验证	1	—	1
2	通过修改局部变量值实施故障注入、覆盖故障分支	2	1	3
3	连续和累计计时考虑不全面	1	2	1
4	未执行正确与错误输入的穿插测试	3	0	2
5	连续若干个测试步骤的预期结果一样	2	0	1
6	最小指令间隔测试不足	1	0	1
7	预期结果错误	7	10	3
8	预期结果仅观察遥测、未观察物理输出	2	1	5
9	预期结果仅确认了合法输出、未确认非法输出	0	3	1
10	预期结果无依据	1	1	2

注：单元格中的“—”，表示缺陷条目在该项目中不涉及。

3.2　数据分析

对统计数据分析如下：

1）通过对 3 个项目的同行评审，共发现并纠正测试用例典型设计缺陷 58 个，应用效果显著。

2）预期结果错误，发生次数最多，应予以重点关注。

3）指定地址范围操作覆盖性缺乏验证、修改局部变量值实施故障注入、连续和累计计时考虑不全面、预期结果错误、预期结果仅观察遥测未观察物理输出、预期结果无依据，上述缺陷在项目中的发生率为 100%。

4）其余缺陷，在项目中的发生率约为 70%。

应用结果表明，上述测试用例典型设计缺陷的提出，对于确保测试充分性、提高测试质量，是切实有效的。

3.3 应用说明

工程实践中，为了消除测试用例典型设计缺陷，可以根据表 1（缺陷汇总表）建立检查单，并通过三级质量管控机制，实现测试用例典型设计缺陷的修正：

1）一级管控，测试人员自查修正。

2）二级管控，项目审核修正。

3）三级管控，同行评审修正。

消除上述覆盖率分析无法修正的用例设计缺陷之后，再根据用例执行后的覆盖率统计数据，进一步修正因输入不足而导致的用例设计缺陷，通过两种手段的互补，确保测试充分性。

4 结束语

测试用例设计缺陷，尤其是通过覆盖率分析手段无法发现和修正的测试用例设计缺陷，不容忽视。目前对此所开展的专题研究较少，后续还需要进一步加强研究，完善缺陷库。为提高测试效率，后续应开展相关工具研究，通过工具自动化手段实现测试用例设计缺陷的自动化检查与消除。其中，构建由知识库和推理机组成的专家系统是一个未来可探索方向。

参考文献：

[1] 丁永康. 基于遗传算法的基本路径测试用例自动生成算法研究 [D]. 武汉：华中科技大学，2016.

[2] 苗亮亮. 基于遗传算法软件测试用例自动生成分析与研究 [D]. 兰州：兰州交通大学，2013.

[3] 宋 倩. 基于遗传算法的测试用例生成技术 [J]. 计算机系统应用，2014，23 (11)：264－267.

[4] 王小银，王曙燕，孙家泽. 基于蚁群算法的三三组合测试用例集的生成 [J]. 计算机应用研究，2015，33 (11)：3328－3331.

[5] 崔应霞. 组合测试技术的研究与应用 [D]. 合肥：安徽大学，2011.

[6] 汪 勇. 基于符号执行的软件测试技术研究与设计 [D]. 成都：电子科技大学，2017.

[7] 高共革. 基于专家系统的测试用例自动生成方法 [D]. 贵阳：贵州大学，2009.

[8] 潘丽丽. 软件测试用例集简化及其构建方法研究 [D]. 长沙：湖南大学，2008.

[9] 华 丽. 基于蚁群算法的测试用例集约简技术研究 [D]. 重庆：西南大学，2009.

[10] 闫字华，李 谊，黄 宁，等. 军用软件测评实验室测评过程和技术能力要求 [S]. 北京：总装电子信息部，2005.

(上接第 4 页)

[3] 李 冲，张 安，毕文豪. 基于 PHM 的作战飞机可用度和任务可靠度计算 [J]. 兵工自动化，2013，32 (9)：37－41.

[4] 艾剑良，王 鹏，高 明. 飞行控制系统的重构技术研究 [J]. 火力与指挥控制，2006，31 (1)：1－3.

[5] Wolfig R，Jakovljevic M. Distributed IMA and DO－297：Architectural，communication and certification attributes [J]. IEEE，2008：978－1－4244－2208－1/08.

[6] 景 博，徐光跃，黄以锋，等. 军用飞机 PHM 技术进展分析及问题研究 [N]. 电子测量与仪器学报，2017，2 (31)：161－169.

[7] 郭阳明，蔡小斌，张宝珍，等. 故障预测与健康状态管理技术综述 [J]. 计算机测量与控制，2008，16 (9)：1213－1219.

[8] 张宝珍，曾天翔. 先进的故障预测与状态管理技术 [J]. 测控技术，2003，22 (11)：4－6.

[9] Liu Z Y，Rong L Q. Study on the multi－agent model for PHM system [J]. IEEE，2011，1044－1048.

[10] 许晋瑞，李 峭，赵露茜，等. DIMA 系统实时通信流量的时延分析方法 [J]. 计算机工程与设计，2015 (4)：879－88.

[11] Gu Q F，Wang G Q，He Y Z. Dynamic reconfiguration mechanism for distributed integrated modular avionics system [R]. AIAA－2015－3285，2015.

[12] 杨德才. 综合模块化航空电子系统的故障预测与健康管理技术 [J]. 现代电子技术，2015，38 (5)：125－128.

[13] 赵宁社，王国庆，王 恒. 一种面向需求的综合化系统健康度量方法 [J]. 计算机工程，2011，37 (21)：267－272.

[14] 李运喜，何 翔. 符合 ARINC653 的多核操作系统任务调度研究 [J]. 航空计算技术，2017，47 (5)：108－115.

[15] Tao X，Zhu Y X，Mao Y S. Designing ARINC653 partition constrained scheduling for secure real time embedded avionics [J]. IEEE，2015：978－1－4673－9300－3/15.

[16] Ye Z A，Wang S H，Zhao T D. IMA dynamic reconfiguration modeling and resource criticality analysis based on Petri net [J]. IEEE，2017：978－1－5386－0918－7.

[17] 阎 芳，邢培培，赵长啸，等. 基于联合 k/n (G) 模型的 DIMA 系统可靠性建模与分析 [J]. 航空学报，2018，39 (6)：321971.

[18] 詹志娟，周 庆，洪 蓉. DIMA 动态重构管理方法研究 [A]. 2016 航空试验测试技术学术交流会论文集 [C]. 测控技术，2016：190－192.

[19] 刘 冬，邓 健，王承惠. 一种基于 ARINC 653 标准机载电子设备健康监控体系：中国，103544092 [P]. 2014－01－29.