

doi: 10.7690/bgzd.2025.06.018

武器装备软件测试用例复用技术

邹诗苑, 张 龙, 丁小琴, 王丽娜, 刘 佳

(上海机电工程研究所, 上海 201109)

摘要: 为提高测试的效率和质量, 对武器装备软件测试用例的复用进行了探讨。提出测试项、测试用例的概念及用例库的基本构架, 采用 Z 规格说明语言对测试用例进行统一描述, 描述方法运用在武器装备软件的测试用例复用实例中, 能保证软件测试的有效性、测试用例的通用性、独立性、规范化和易修改性。通过测试用例的复用, 结果表明: 该研究能加快测试用例的设计进度, 减少测试人员的负担, 提高武器装备软件测试的效率和质量。

关键词: 测试用例; 复用; 武器装备软件; Z 语言

中图分类号: TJ06; TP311.55 **文献标志码:** A

Test Case Reuse Technology for Weapon Equipment Software

Zou Shiyuan, Zhang Long, Ding Xiaoqin, Wang Li'na, Liu Jia

(Shanghai Electromechanical Engineering Institute, Shanghai 201109, China)

Abstract: In order to improve the efficiency and quality of test, the reuse of test cases for weapon equipment software is discussed. The concepts of test items and test cases and the basic framework of the use case library are proposed. The Z specification language is used to describe the test cases in a unified way. The description method is used in the test case reuse examples of weapon equipment software, which can ensure the effectiveness of software testing and the universality, independence, standardization and easy modification of test cases. Through the reuse of test cases, the results show that the research can speed up the design progress of test cases, reduce the burden of testers, and improve the efficiency and quality of weapon equipment software testing.

Keywords: test case; reuse; weapon equipment software; Z language

0 引言

随着装备系统数字化和智能化程度越来越高, 软件产品就愈发庞大, 软件的质量要求不断提高, 对软件测试工作的效率、质量提出了更高的要求。武器装备软件测试工作量非常大, 通常需要设计大量的测试用例, 才能覆盖所有的软件需求, 测试用例的设计成为整个测试过程中耗时最长、难度最大的节点。如何在短时间内高质量地完成测试用例设计是武器装备软件测试急需解决的问题。

测试用例复用就是将一个软件已经执行过的测试用例, 在不同时间、不同程度地应用到同一软件新的测试或同类软件的测试中。优秀的测试用例记录有大量的测试数据、结果, 同时也体现软件测试的方法和技巧。充分利用前期测试过程中积累的经验、素材, 建立测试用例库, 通过测试用例的复用, 不仅能加快测试用例的设计进度, 还能减少测试人员的负担, 提高武器装备软件测试的效率和质量^[1-3]。笔者对武器装备软件测试用例特征进行研究, 提出可复用的测试用例设计方法, 旨在提高武器装备软件测试用例的质量与软件测试的效率。

1 测试用例复用的质量特性

可复用性是用例的主要特性之一。用例的可复用性是根据用例本身的描述或特征, 判断在未来应用中用例潜在的加以利用的程度。文献[4-5]描述了可复用测试用例的特征。对大量测试用例及其复用情况进行研究与分析后, 笔者认为可以把用例的可复用性分解成以下 4 个要素:

1) 通用性。用例的通用性指的是用例测试的应用领域或应用测试场景覆盖面的广度。通用性越广的用例越易于复用。通用性的度量可用来度量用例测试的应用领域或测试场景的覆盖度。

2) 独立性。通常用例的执行不是一个独立的过程, 一些用例运行的初始状态取决于另外一些用例的终止状态; 因此, 复用一個用例, 该用例必须要有相对的独立性, 即用例之间具有无关联性。用例的独立性是指在要求的运行环境下, 用例不依赖其他用例而能独立工作的程度。

3) 易修改性。用例的易修改性是指用例可以被修改的容易程度。用例的复用方式分为直接复用和间接复用。间接复用是指用例不能满足当前软件测

收稿日期: 2024-08-23; 修回日期: 2024-09-21

第一作者: 邹诗苑(1985—), 女, 贵州人, 硕士。

试需求, 需经过测试工程师的修改才能使用, 是采用最多的复用方式。通常一个用例被复用后, 都要经过适当的修改才能满足测试工程师的要求。因此, 用例的易修改程度对该用例被复用的可能性有极大地影响。

4) 规范化。一个好的测试用例不仅要体现软件的测试思想、技巧, 同时还要包含大量的测试数据、结果以及测试过程。用一定的规范将测试用例组织成一个统一的标准形式, 这样的测试用例才是可以被理解的、可复用的。

此外对测试用例库的管理也要有一套规范的流程。例如: 依照测试类型对测试用例的划分规则, 测试用例基准库的建立标准, 测试用例变更管理流程, 测试用例的版本控制等。

2 可复用的测试用例库设计

统一规范的描述和定义可以为测试用例库提供良好的复用机制。根据装备软件的测试经验及步骤, 笔者按照需求规格说明, 可复用的测试用例库包括: 测试项与测试用例, 同时也提供测试用例库的基本架构。

2.1 测试项

测试项是根据需求规格说明进行分解的各项功能点的集合, 例如, 系统加电初始化是一个系统功能, 它便是一个测试项, 每个测试项下包含多个测试用例。测试项定义如下:

定义 1: 测试项 (test item) 是一个四元组: $TI=(TID, ID, Req, Ts)$, 即 {测试项名称, 测试项标识, 关联的软件需求, 测试级别}。

测试项名称 (test item definition, TID) 一般是待测功能的简要体现。测试项标识 (identification, ID) 是标识一个唯一的测试项。关联的软件需求 (requirement, Req) 是该测试对应的需求规格说明里的功能项。测试级别 (test stage, Ts) 包含单元测试、组装测试及配置项测试。

2.2 测试用例

测试用例是若干个有先后顺序的测试步骤的集合, 是可独立执行的测试过程描述。在可复用的测试用例库中, 测试用例是可复用的最小单位^[6]。测试用例定义如下:

定义 2: 测试用例 (test case) 是一个八元组: $TC=(TCD, ID, Ty, Pre, Ter, St, Er, V)$, 即 {测试用例名称, 测试用例标识, 测试类型, 先决条件, 终止

条件, 测试步骤, 期望输出, 复用次数}。

测试用例名称 (testcase definition, TCD) 一般体现该测试用例的功能侧重点。测试用例标识 (identification, ID) 是标识一个唯一的测试用例。测试类型 (type, Ty) 描述了测试用例的分类: 主要功能测试、性能测试、接口测试、边界测试、强度测试等。先决条件 (prerequisite, Pre) 是测试用例执行的必要条件, 表示测试用例执行前需要让测试环境达到指定的状态, 或设置为指定的状态。终止条件 (termination, Ter) 一般为用例正常执行完成或遇到故障停止。测试步骤 (step, St) 表示执行该测试用例的每个操作过程, 是若干个有序的序列。期望输出 (expected result, Er) 描述当执行操作或输入数据后, 被测对象应该做出的预期反应。复用次数 (version, V) 即测试用例的版本字段, 所有测试用例的版本字段 Version 初值 1.0, 可复用测试用例本身经过修改或重新设计的, Version 值依次为 2.0、3.0 等。

2.3 测试用例库管理系统的架构

测试用例库是有经验的、优秀的用例的集合, 建立是能有效管理和快速复用历史的测试用例, 从而提高软件测试工作效率和质量^[7]。测试用例库一般包括用例管理模块、查询模块、交互模块和系统管理模块, 测试用例库的架构如图 1 所示。

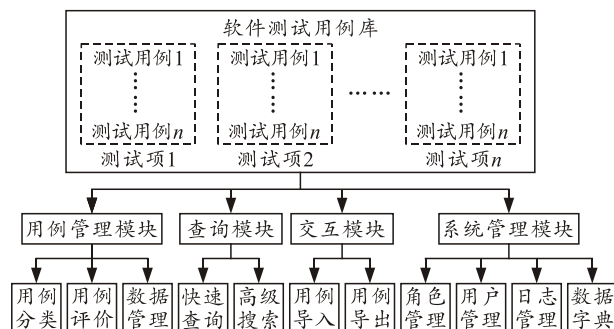


图 1 测试用例库的架构

用例管理模块中包含用例分类、用例评价、数据管理 3 大功能, 用例分类目的是便于系统管理, 给用户的分类查询提供帮助。用例评价可给后续的用例选择、复用时提供参考标准。数据管理可实现用例基本信息、执行信息、复用信息的添加、删除及修改操作。

查询模块中包括快速查询和高级搜索功能。可根据用户输入的查询字段内容, 从众多的测试用例中快速查询出相应的测试用例, 便于用户对测试用例的复用操作。

交互模块中包括用例导入和用例导出的功能。一般支持 Excel 格式的文件导入, 便于对大量用例进行操作; 而选择需要的字段, 就可以将所选用例导出, 便于用户对用例进行借鉴、复用。

系统管理模块中包括角色管理、用户管理、日志管理、数据字典等。主要是负责整个测试用例库的角色分配、系统维护和后台管理等操作。

表 1 3 种测试用例描述方法性能分析

描述方法	基本步骤	描述粒度	通用性	易修改性	规范性
自然语言	广大测试员熟知的自然语言进行描述	粒度完全取决于用例设计者	易于通用	易修改	规范性不足
结构化表格	指定表格中, 将测试内容分栏目填写	粒度取决于用例设计者	通用性不足	不易修改	规范
形式化方法	基于数学方法来描述软件系统性质的方法	粒度细	易于通用	易修改	规范

通过上述 3 种测试用例描述方法性能分析可以看出: 形式化描述方法能克服自然语言描述时可能带来的模糊性和歧义性, 通过形式化、规范化的数学方法, 描述测试用例的前置条件, 测试步骤和操作结果等^[9-12]。选择形式化描述方法来对测试用例进行描述, 正好符合可复用的测试用例在通用性、易修改性、规范性上的要求。其中 Z 规格说明语言的测试用例描述方法被用于基于需求规格说明的测试用例的自动生成和验证中, 适用于本文中武器装备软件的测试用例描述中。

4 基于 Z 规格说明语言的可复用测试用例

4.1 Z 语言描述测试用例框架

为进行测试有关的模式定义, 下面给出可复用测试用例库中一些 Z 语言描述的类型声明。测试级别、测试类型、先决条件、终止条件这 4 项信息的内容在同一领域相对较为固定, 可以将其枚举列出: $[Test\ Stage]=\{Unit, Integration, System\}$

$[Test\ type]=\{Function, Interface, Performance, Security, Reliability, Installation, Workload\}$

$[Prerequisite]=System\ is\ powered\ \wedge (Environment=true)$

$[Termination]=(Step1\ \wedge\ Step2\ \wedge\ \cdots\ \cdots\ Stepn)\ \vee (\exists\ C?:(Environment\ or\ Stepm).Stepm\ \vee\ Step(m+1))\ \vee\ \cdots\ \cdots\ Stepn=false)$

测试用例的 ID, 测试类型, 测试阶段, 复用次数等会在描述一组同类型的测试用例的操作步骤之前给出一个如下的初始化描述:

—————TestInit—————

ID: SerialNumber

TID: $Ty \in TestType$

$Ts \in TestStage$

3 测试用例表述方法选择

在文献[8]中描述测试用例一般的描述语言, 鉴于武器装备软件的测试是基于设计说明与需求说明来开展的, 不能采用脚本语言、测试工具专用语言等, 笔者选择自然语言、结构化表格、形式化 3 种描述方法进行比较, 表 1 对 3 种测试用例描述方法进行性能分析。

$V=1.0+0.x$

4.2 Z 语言描述测试用例实例

为验证 Z 语言描述可复用测试用例的可行性, 这里给出一个装备软件中导弹发射功能测试的可复用测试用例的 Z 语言描述。导弹发射在满足允射条件时, 软件响应导弹发射令, 否则不响应。

首先给出所需属性的集合:

$[Target\ Attribute]=\{Our, Enemy, Unknown\}$ //目标属性

$[F-Vehicle\ Condition]=\{Normal, Abnormal\}$ //发射车状态

$[Launching\ Zone]=\{In, Out\}$ //目标是否进入发射区

$[Missile\ Status]=\{Power\ On, Power\ Off\}$ //导弹状态

$[Launch\ Order]=\{On, Off\}$ //发射令

按照测试项的元素, 其 Z 语言描述如下:

—————Test Item—————

TID=Missile launching

ID=DDFS

Req=X.XX.XX//一般可用需求规格说明章节序号

$Ts=System$

按照测试用例的元素, 其 Z 语言描述如下:

—————Test Case 01—————

TCD=Missile launching (normal)

ID=CSCIT_DDFS_001

$Ty=Function \wedge Interface$

$Pre \in Prerequisite$

$Ter \in Termination$

$St01: (Target\ Attribute? = Enemy) \wedge (F-Vehicle\ Condition? = Normal) \wedge (Launching\ Zone? = In)$

$St02: Missile\ Status?:PowerOn$

Er: Launch Order! = On

V: 1.0

—————Test Case 02—————

TCD=Missile launching (abnormal)

ID=CSCIT_DDFS_002

Ty=Function $\wedge$ Interface

Pre $\in$ Prerequisite

Ter $\in$ Termination

St01: (Target Attribute? = Our) $\wedge$ (F-Vehicle
Condition? = Normal) $\wedge$ (Launching Zone? = In)

St02: Missile Status? : PowerOn

Er: Launch Order!= Off

V: 1.0

可见,通过 Z 语言描述的测试用例元素简单清晰,测试用例的输入、输出描述准确,并无二义性,从而保证了可复用测试用例的通用性、独立性、规范化和易修改性。

5 测试用例的复用

在测试用例复用中,为对测试资源进行共享,测试用例库的建立是关键。为提供测试用例新建、编辑、复用等功能,建立适合武器装备软件的测试管理平台是十分必要的,旨在为测试人员提供软件测试项目监控、管理,测试过程和知识共享、复用于一体的软件测试服务。测试用例复用步骤如图 2 所示。

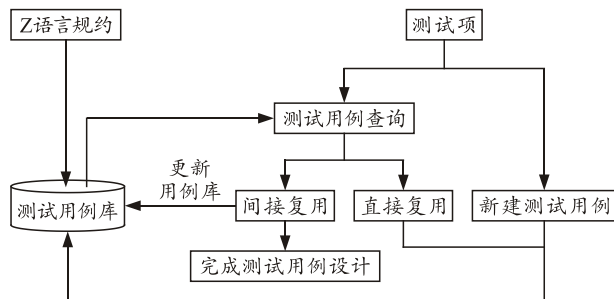


图 2 测试用例复用步骤

用例的复用一般采用以下步骤:

- 1) 通过需求中功能点分解,建立测试项;
 - 2) 在测试用例库中进行查询,选择与测试项关联的测试用例;
 - 3) 选择复用策略,包括直接复用或间接复用;
 - 4) 编辑、完善测试用例;
 - 5) 完成该测试项的用例设计。
- 为提高复用的效果,测试用例是可编辑的,即

间接复用的方式,间接复用的测试用例仍可添加至测试用例库中,用于更新用例库。应注意的是,新建的测试用例也可以添加到用例库中,为下一次用例的复用提供资源。

6 结束语

笔者对可复用测试用例的质量特性进行了研究,提出测试项、测试用例、测试用例库概念,并使用 Z 规格说明语言对测试项及测试用例进行描述,最后将这种描述方法用在装备软件中导弹发射功能的测试项及测试用例的描述中,通过测试用例的复用步骤说明了该方法的有效性。该方法不仅逻辑清晰、测试用例属性全面,而且表达上更加严谨:规范,也更加有利于武器装备软件的测试用例复用,从而缩短测试用例设计时间,提高武器装备软件测试的效率和质量。

参考文献:

- [1] 李春秋. 软件测试用例复用策略研究及实现[J]. 电脑知识与技术, 2020, 16(22): 218-220.
- [2] 姜蓉, 崔仕颖. 基于模型的航空机载软件测试用例复用方法研究[J]. 工业控制计算机, 2021, 34(6): 15-17.
- [3] 姚佳瑜. 软件测试中的测试用例及复用研究[J]. 数字技术与应用, 2018: 58-59.
- [4] 王珊珊. 软件测试中可复用测试用例研究[J]. 信息技术与信息化, 2015(3): 119-121.
- [5] 袁松, 杨根兴, 张娟. 基于层次分析法的测试用例可复用性度量研究[J]. 计算机应用与软件, 2001, 28(9): 60-62.
- [6] 董成基, 齐杏林, 吕静, 等. 飞行控制软件测试用例生成技术[J]. 兵工自动化, 2012, 31(9): 93-96.
- [7] 叶素梅, 肖寒. 基于测试用例库的复用方法研究[J]. 电脑知识与技术, 2018, 14(32): 272-273.
- [8] 张娟. 软件测试中测试用例复用的研究[D]. 上海: 上海大学, 2012: 16-19.
- [9] 王明兰, 叶东升. 测试用例描述语言研究[J]. 计算机工程与设计, 2006, 27(22): 4281-4284.
- [10] 李莹, 吴江琴. 软件工程形式化方法与语言[M]. 杭州: 浙江大学出版社, 2009: 2-5.
- [11] 缪准扣, 陈怡海. 软件形式规格说明语言——Z[M]. 北京: 清华大学出版社, 2012: 3-6.
- [12] 龙腾, 刘震宇, 史人赫, 等. 基于神经网络的防空武器目标智能分配方法[J]. 空天防御, 2021, 4(1): 1-7.