

# 测试用例复用研究与实践<sup>①</sup>

万琳, 廖飞雄

(装甲兵工程学院 信息工程系, 北京 100072)

**摘要:** 测试用例设计在军用软件功能测试中需要占用大量的时间和人力。有效复用已有的测试用例能够缩短测试用例设计的时间, 减少测试用例设计的工作量, 从而提高测试的效率。针对大型军用软件功能测试用例的复用进行了探讨, 提出了一种军用软件的测试用例复用方法, 并在工程中进行了实践应用。

**关键词:** 软件质量保证; 功能测试; 用例复用; 军用软件

## Research and Practice on Test Case Reuse

WAN Lin, LIAO Fei-Xiong

(Department of Information Engineering, Academy of Armored force Engineering, Beijing 100072, China)

**Abstract:** The designing of test case costs a lot of time and manpower in military software functional testing. Effective reuse existing test case can shorten the time and reduce the workload of test case designing, leading to high testing efficiency. This paper researches on the technology of the reuse of functional test case for large scale military software. A reuse method in military software functiona testing is proposed, and it has been practiced in project.

**Key words:** software quality assurance; functional testing; test case reuse; military software

## 1 引言

随着信息技术的发展和在世界新军事变革的推动下, 制信息权成为夺取未来战场主动权的关键因素, 直接决定未来战争的胜负, 信息化装备成为各国军队的方向发展, 我军也不能例外。由于装备信息化水平的不断提高, 军用软件的“核心地位”和“神经中枢”作用也日益突出, 军用软件的质量已成为影响装备质量的关键因素<sup>[1]</sup>。

软件测试是保证软件质量的重要方法, 而功能测试又是对军用软件测试时最常用的一种测试方法。在对军用软件进行功能测试时的工作量非常大, 而其测试又要求很高的效率, 要在尽量短的时间内完成测试任务。如何通过一些有效的手段和方法, 在最短的时间内, 用最小的费用完成测试是军用软件测试急需解决的一个问题。

测试用例复用是指测试人员在执行一项新的测试工作时, 通过直接调用或修改现有的、适合此项测试的测试用例, 并将它们运用在测试中的过程<sup>[2]</sup>。有效

复用现有的测试用例能加快测试用例的设计进度, 减少测试人员的负担<sup>[3]</sup>, 提高测试的效率。本文在长期实践的基础上, 针对军用软件功能测试提出了一种测试用例复用方法, 并在对某军用软件的测试中验证了该方法的有效性。

## 2 一种测试用例复用方法的实现

### 2.1 复用的框架原理

可复用的测试用例的存在是测试用例复用的重要条件<sup>[4]</sup>。因此可以通过对测试用例进行一定规则的处理和组织构建可复用测试用例库。本文在基于对界面构件操作的基础上构建可复用测试用例库, 从而实现测试用例的复用。测试用例的复用及可复用测试用例库的构建过程如图1所示。

图1描述了可复用测试用例的生成及测试用例复用过程。在设计测试用例时, 可以分为以下三个阶段:

#### ① 复用测试用例的匹配

首先对被测软件的界面构件进行分析, 确定待测

① 收稿时间:2010-11-04;收到修改稿时间:2010-12-04

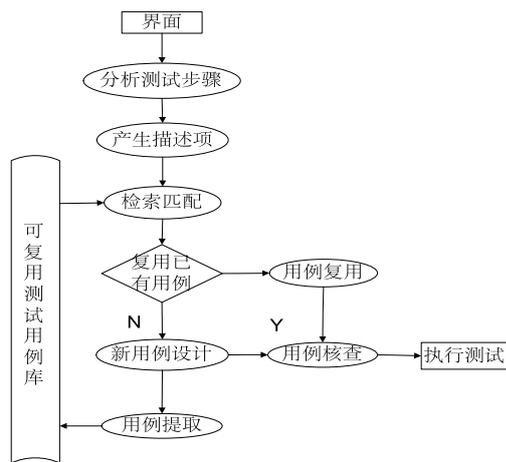


图 1 测试用例复用的框架

构件的测试步骤和对控件的相应操作，然后抽取其类型及功能特性产生描述项，与可复用测试用例库中已有项进行检索匹配；

### ②测试用例复用

如果在可复用测试用例库中可检索到与待测构件相同的描述项或相近的描述项，则复用库中已有的测试用例；

### ③生成可复用测试用例并入库

如果在可复用测试用例库中没有检索到与待测构件相同或相近的描述项，则设计新的测试用例，并且按一定的规则对测试用例进行提取，通过关联描述项对其进行标记后存入数据库中，为以后的测试用例复用做准备。

另外，由于个别界面构件，尽管其描述项相同或相近，但在细节上可能仍存在一些差别，可能会导致其测试用例上的差别，故在执行测试前，必须对测试用例进行核查<sup>[5]</sup>，避免发生错误。

## 2.2 生成可复用测试用例

在复用测试用例时，描述项不仅是可复用测试用例的标记，更是可复用测试用例库检索及用例匹配的依据。可复用测试用例的提取就是抽取相对通用的测试操作，对其产生描述项，并在此基础上将用例与描述项关联，以便为后期类似的界面构件测试时复用。

### 2.2.1 抽取描述项

本文从软件 GUI 自动化测试的原理和方法，利用构件的两个属性<sup>[6]</sup>抽取描述项：

①标题：标题是界面构件的一个确定的名称，也可以说是界面构件的显示文本，同时也是程序准确识

别一个界面构件的最重要的属性之一。

②构件类型：总体上讲，可以将构件分为两大类：

\* 行为构件——用户交互可以触发某个事件，如界面中的选择菜单、按钮等构件。

\* 信息构件——接受用户输入数据或根据控制条件输出数据，如文本框、列表框等。

在此两大类构件中又可以根据其功能性质划分为更小的类。例如：如果某文本框中只能输入数字，就可以将其定义为“数字文本框类型”；如果其用来显示执行结果，可将其定义为“输出文本框”。通常同类型的构件具有相同或相近的输入和输出。

基于上述两个属性，描述项格式定义为：标题（或功能特性）—类型，如描述项“姓名—中文文本框”，其中“姓名”是对其功能特性的抽取，“中文文本框”是该构件的类型，表示该构件用来输入姓名的文本框。

### 2.2.2 可复用测试用例的生成规则

通常设计出的测试用例与模块环境的相关性较大，其具体的实现包含的细节较多不利于复用，因此需要经过一定规则的处理之后才能方便复用：

#### ①提取共性特征

从测试用例中发掘“共性”特征可以加大复用的机会。本方法通过对构件的输入域进行等价类划分的方式，使得复用的匹配过程可以基于等价类进行，从而可以在复用时排除个别细微差异带来的影响，提高匹配成功率。

定义 1. 对功能构件  $P$  进行测试时的输入集合  $IN = \{in_1, in_2 \dots in_n\}$ ， $P$  构件的输入域

$Q = \{IN_1, IN_2 \dots IN_n\}$ ，其中  $\forall i, j \in (1, n)$  且  $i \neq j$  使得  $IN_i \cap IN_j = \Phi$ 。

规则 1 如果对于已测试的构件  $P$ ，和待测构件  $P_1, P_2 \dots P_m$ ， $\exists Q' = Q \cap Q_1 \cap Q_2 \dots Q_n = \{IN_{i_1}, IN_{i_2} \dots IN_{i_k}\}$ ，其中  $1 \leq i_1 \leq i_2 \dots \leq i_k \leq n$  则  $IN' = \{in'_1, in'_2 \dots in'_k\}$ ，其中  $in'_j \in IN_{i_j}, 1 \leq j \leq k$ ，可提取入可复用测试用例库。

定义 1 对构件的输入域划分等价类，而规则 1 则是基于等价类提取共性特征。

#### ②抽象期望结果

对测试用例的期望结果进行抽象可以扩大复用的范围。本方法通过在多个测试用例间挖掘其内在的联系，进而封装成一个独立的实体。当复用库中一个实体时，实质上是同时复用了多个测试用例，从而提高了效率。

定义 2. 在测试中对各控件  $PA = \{P_i | i = 1, 2, \dots\}$  的操作次序可以形成一条路径  $w = (P_1 P_2 \dots P_n), n = 1, 2, \dots$ ,

规则 2. 如果对某一功能点测试时  $\exists W = \{w_1, w_2 \dots w_m\}$  和  $w = (P_1 P_2 \dots P_k)$ ,  $\forall w_i = \{P_{i_1} P_{i_2} \dots P_{i_n}\}$  使  $i_n > k, w = (P_{i_1} P_{i_2} \dots P_{i_n})$  则  $W' = \{w'_i = w_i - w | i = 1, 2, \dots, m\}$  可抽象为  $w$  的期望结果, 提取入库。

定义 2 定义了基于控件操作次序的测试路径, 规则 2 是基于测试路径的可复用测试用例提取方法。

③筛选求精

库中用例的数量太多, 会导致匹配的效率降低, 因此必须有选择的入库, 力求提取的用例具有典型性, 代表性。应用时还要定期对用例库进行维护, 去掉库中的冗余用例。

定义 3. 可复用测试用例库中的测试用例用  $tc$  表示, 多个用例的集合为  $TC = \{tc_1, tc_2 \dots tc_n\}, n = 1, 2, \dots$

规则 3. 如果库中  $\exists TC = \{tc_1, tc_2 \dots tc_n\}$  和  $tc$ ,  $\forall tc_i \in TC, i = 1, 2, \dots, n$ , 使  $tc$  与  $tc_i$  的差别较小

则用例库中可以保留  $tc$ , 去掉  $TC$ 。

2.3 复用方法

2.3.1 用例的查找算法

在进行复用时, 要从可复用测试用例库中查询所需要的测试用例。可复用测试用例的查找是通过对描述项的匹配算法实现的。

设需要进行测试的构件  $P$ , 进行检索的测试用例库为  $C = \{C_1, C_2 \dots C_m\}, C_i = \{D, \{c_1, c_2 \dots c_n\}\}, D = \{h, t\}$ , 其中  $D$  为描述项, 由  $h$ (标题)和  $t$ (类型)组成,  $\{c_1, c_2 \dots c_n\}$  为与描述项关联的用例集, 匹配算法具体如下:

/\*算法按描述项的匹配度依次输出  $RA, RH, RT$  \*/

①为构件  $F$  产生描述项  $headline$ (标题),  $type$ (类型),

②选取  $C_i \subset C, 1 \leq i \leq m$  则  $D_i \subset C_i, t_i \in D, h_i \in D$ ,

③判断  $h_i = headline$ , 为真则进行下一步, 否则转至⑦,

④判断  $t_i = type$ , 为真则进行下一步, 否则转至⑥,

⑤  $RA = RA \cup C_i$ , 转至⑨,

⑥  $RH = RH \cup C_i$ , 转至⑨,

⑦判断  $t_i = type$ , 为真则进行下一步, 否则转至⑨,

⑧  $RT = RT \cup C_i$ , 进行下一步,

⑨用户终止查找则转至⑩, 否则进行下一步,

⑩如果  $i \leq m$  转至②, 否则进行下一步,

⑪输出  $RA, RH, RT$ , 算法结束。

通过查找得出的最终结果分别为:

$$RA = \{C_i | (1 \leq i \leq m) \wedge (C_i \subset C) \wedge (D \subset C_i) \wedge (h_i, t_i \in D_i) \wedge (h_i = caption) \wedge (t_i = type)\}$$

$$RA = \{C_i | (1 \leq i \leq m) \wedge (C_i \subset C) \wedge (D \subset C_i) \wedge (h_i, t_i \in D_i) \wedge (h_i = caption) \wedge (t_i \neq type)\}$$

$$RA = \{C_i | (1 \leq i \leq m) \wedge (C_i \subset C) \wedge (D \subset C_i) \wedge (h_i, t_i \in D_i) \wedge (h_i \neq caption) \wedge (t_i \neq type)\}$$

通过匹配算法将查找出多个用例集, 测试人员必须根据描述项的匹配程度和待测构件的实际情况选择用例进行复用。

2.3.2 复用的实现

测试用例复用就是在看似相互关联不大的不同构件之间抓住其本质的联系<sup>[7]</sup>。笔者提出的复用方式主要有两种:

①参考引用。其实质就是舍弃测试用例中非本质的个别的特性带来的影响, 引用其本质。在复用测试用例时, 以复用测试用例库中的构件为基础, 根据待测构件的特点对库中用例做一定的变动或组合等。例如:

表 1 中  $V_1, V_2$  是可复用测试用例库中构件,  $V_3, V_4, V_5$  为待测构件, 由于对  $V_3$  的操作, 可以引出对  $V_4, V_5$  的操作, 如图 2 所示, 因此, 可通过复用  $V_1, V_2$  的用例来产生新测试用例。

表 1 相关构件及描述项

构件	描述项	输入	期望结果
$V_1$	确认—按钮	单击	确认操作
$V_2$	取消—按钮	单击	放弃操作
$V_3$	删除—按钮	单击	弹出提示界面
$V_4$	确认—按钮	单击	确认操作
$V_5$	取消—按钮	单击	放弃操作

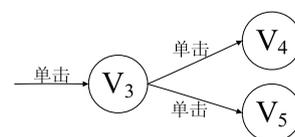


图 2 某构件的操作时序

具体为: 对于  $V_4, V_5$  依次引用  $V_1, V_2$  对应描述项的测试用例, 再与  $V_3$  用例结合形成新测试用例。

②扩展引用。即在直接引用其“共性”的基础上进行扩展。进行用例设计时, 将产生的描述项在复用

测试用例库中检索, 往往很难存在完全相同的描述项。在复用相近的描述项时, 要充分利用它们的共性, 暂时隐藏它们之间的细节差异, 最大限度利用“共性”部份。在“共性”的基础上, 再找出它们之间的差异, 根据新描述项的“个性”特征进行扩展。

例如: 表 2 中是两个不同界面上的构件, 构件  $V_1$  是可复用测试用例库中构件, 构件  $V_2$  是待测构件,  $V_2$  测试操作如图 3 所示。

表 2 两个构件及描述

构件	标题	类型	描述项	输入域	期望结果
$V_1$	添加器材	界面按钮	器材添加一界面按钮	单击	进入器材添加操作
$V_2$	添加人员	界面按钮	人员添加一界面按钮	单击	进入人员添加操作

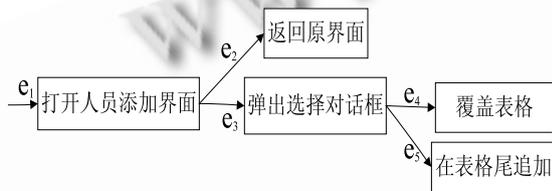


图 3 某软件的添加人员操作示意图

构件  $V_1$  和  $V_2$  的共性特征是: 两个构件均为添加操作; 而个性特征是:  $V_1$  和  $V_2$  的入口操作不同, 一个是“人员功能”, 一个是“器材功能”。故  $V_2$  在复用  $V_1$  时, 直接引用  $V_1$  描述项所关联的用例中的代表“共性”部份的用例实体, 再扩展成与人员相关的操作, 就可以得到新的测试用例集。

### 3 实践应用

为验证本文所提出的测试用例复用方法的有效性, 笔者在对某军用软件的功能测试中对该方法进行了应用。待测试软件为大型信息系统, 菜单层次平均为 4 级, 最低为 2 级, 最深可达 6 级。共计 8 个一级菜单, 底层菜单为 279 个, 每个底层菜单对应一个功能模块。在此次测试用例设计过程中, 测试组历时 3 个月, 完成测试用例共计 5525 个, 其中未通过复用产生的用例 2672 个, 复用产生用例 2853 个。在测试中, 我们对每个月完成的测试用例数进行了统计, 如图 4:

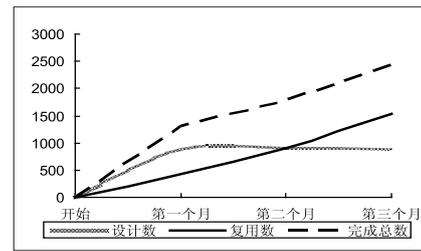


图 4 测试用例复用统计图

从图中可以看出, 完成的测试用例总数和复用的测试用例数在开始时均较少, 但随着测试的进行, 以后逐渐均有所增加。这是因为在测试初期, 可复用测试用例库中的测试用例数较少, 复用的范围较小。随着测试过程的深入, 库中可复用测试用例的数量增加, 则复用产生的测试用例也随之增加, 从而使相同时间段内产生的测试用例总数明显增加, 提高了测试的效率。

### 4 结语

本文针对军用软件的功能测试用例复用方法进行了探讨, 提出了一种测试用例复用方法, 并在实际工作中加以实践。实践数据证明: 利用测试用例复用技术可以大大提高测试用例设计的效率, 从而缩短了测试的时间, 减少了测试的费用。在下步的研究中, 我们将进一步完善相应的支持工具, 提高自动化程度。

### 参考文献

- 刘根. 军用软件质量保证问题研究. 质量与可靠性, 2010, 147(3):48-50.
- 卜国峰, 孙志刚, 丁小良. 软件测试用例的复用研究. 四川兵工学报, 2009, 30(5):124-126.
- 张玉彬, 谢康林. 测试用例的设计和复用技术. 计算机应用与软件, 2008, 25(1):100-101.
- Prieto-Diaz R. Status report: Software Reusability. IEEE Software, 1993, 10(3):61-66.
- Lee NY. An empirical study of software reuse with special attention to ADA. IEEE Transaction, 1997, 23(9):537-544.
- 许赞泉. 针对图形用户界面的软件测试研究[硕士学位论文]. 南京: 南京航空航天大学, 2003.
- Myers GL. The Art of Software Testing. Wiley- Interscience, 1979.