

# 基于软件、测试、缺陷“三域”铰链的软件测试新架构研究与应用

顾滨兵, 李军锋, 朱佳良

(中国人民解放军 91404 部队, 河北 秦皇岛 066000)

**摘要:** 为推动软件测试更加精准深入高效, 促进软件测试从“遍历式”向“启发式”转变, 提出了一种新型的基于软件、缺陷、测试“三域”铰链的关联知识启发式的新型软件测试架构; 通过信息表征模型化、缺陷数据知识化、测试投向精准化等, 突破了软件测试数据聚类重组技术和软件典型缺陷模式提取技术, 设计开发了相应的软件测试信息管理系统, 实现软件精准化测试; 在提高软件测试效益的同时, 也为软件研制单位和和使用单位提高研发质量、快速诊断排故等提供更具针对性的数据服务。

**关键词:** 软件测试架构; 三域铰链; 信息表征; 启发式; 知识化; 精准测试

## Research and Application of New Software Testing Architecture Based on Three Domains Hinges of Software, Test and Defect

GU Binbing, LI Junfeng, ZHU Jialiang

(No. 91404 of PLA, Qinhuangdao 066000, China)

**Abstract:** To improve the accuracy, depth and efficiency of software testing, and promote the software testing from ergodic to heuristic, a new software testing architecture based on the associated knowledge heuristic of three domains hinges of software, test, and defect is proposed. Based on the technical characteristics of information representation modeling, defect data knowledge and accurate test orientation, it breaks through the technologies of software testing data cluster recombination and software typical defect pattern extraction, a corresponding software test information management system is designed and developed. The accurate software testing is realized, it not only improves the efficiency of software testing, but also enhances the research and development quality for software development companies and users, which provides more accurate data services for rapid diagnosis and troubleshooting.

**Keywords:** software test architecture; three domains hinge; information characteristics; heuristic method; knowledgeable; precise testing

## 0 引言

软件测试是贯穿整个软件开发生命周期, 对软件产品进行验证和确认的活动过程, 也是对软件产品质量持续的评估过程, 其目的是尽快尽早地发现在软件产品中所存在的各种问题, 尽最大可能揭示软件开发过程中所存在的产品质量风险<sup>[1]</sup>。软件工程实践证明, 软件测试是检验软件质量, 把好软件质量关的关键手段。通过开展软件测试, 测试人员在软件中发现并解决了大量软件缺陷, 软件测试对软件质量提升作用十分明显。

随着软件规模越来越大、结构越来越复杂, 软件测试从伴随软件开发、发现软件错误的一项活动, 向相对独立、成熟方法、标准手段的专业方向发展。软件测试已成为一个专业, 形成了专门的方法手段, 遵循规范的流程规程。软件测试通常按照被测软件的不同类型特点, 分别采用静态、动态、白盒、黑盒等测试方法, 单元测试、集成测试、

配置项测试、系统测试等测试级别<sup>[2]</sup>, 并根据不同软件制定针对性策略, 按照不同次序对软件开展测试。

软件测试过程中, 特别是对于大型复杂软件, 必须设置恰当的测试策略, 选择合适的测试工具和方法手段, 合理分解和设计测试用例, 有效搭建测试环境, 才能优质高效地发现软件问题, 客观评价质量, 完成测试任务, 而这些工作需要建立相应标准统一的测试框架, 这就是软件测试架构。

但目前, 从单元测试到系统测试、从功能测试到非功能测试的传统线性架构, 不能体现软件敏捷开发、快速迭代和柔性扩展、高度复用等特点, 也不能体现软件测试从层层递进向分级分类迭代式验证的发展新趋势。为有效应对软件比重的快速增长以及井喷式软件测试需求, 根据软件敏捷开发和迭代升级特点, 设计了一种基于软件、测试、缺陷“三域”铰链的软件测试新型架构, 通过测试数据跨项目共享复用, 能够有效实施全过程敏捷测试, 极大提升

收稿日期: 2023-04-01; 修回日期: 2023-05-23。

作者简介: 顾滨兵(1978-), 男, 硕士研究生, 正高级工程师。

引用格式: 顾滨兵, 李军锋, 朱佳良. 基于软件、测试、缺陷“三域”铰链的软件测试新架构研究与应用[J]. 计算机测量与控制, 2023, 31(10): 97-102, 121.

软件测试质效。

## 1 软件测试架构的分析与研究

在大型软件的测试中,软件测试架构对提高测试团队的技术能力,有效开展软件测试,起着越来越关键的作用。虽然软件测试架构是一个新兴概念,但像微软、阿里巴巴集团等软件公司早就设置了软件测试架构师的职位,由软件测试架构师对软件测试进行规划指导、整体设计、并统筹测试资源、安排各项测试活动等。

### 1.1 软件测试架构的发展及现状

最初,软件测试作为辅助软件编码、提高软件质量的重要手段,其测试架构高度依赖于软件开发架构。软件测试为了与研发活动相衔接,在软件开发模型的基础上,建立配套的软件测试模型。目前常见的测试模型有 V 模型、W 模型(见图 1)、H 模型、X 模型<sup>[3]</sup>、“跑道”模型<sup>[4]</sup>等。

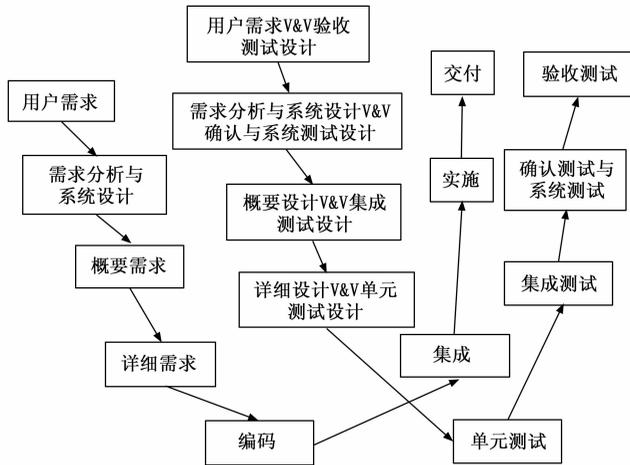


图 1 W 模型测试架构

V 模型强调了在整个软件项目开发中需要经历的若干测试级别,并与开发级别对应。W 模型强调了测试计划工作的先行和对系统需求,以及系统设计的测试。在 H 模型中,软件测试活动完全独立,贯穿于整个软件周期,与其他流程并发进行。此外还有 X 模型,多根并行线代表着变更可以在各个部分发生,提高了迭代效率。虽然从 H 模型开始,软件测试活动独立,但从测试模型构建角度上讲,还是与开发模型相互绑定的。

随着软件测试的进一步发展,软件测试相对于软件研发进一步独立,2018 年起,国内首次提出软件测试模型—“川模型”测试架构<sup>[5]</sup>,开始建立独立的测试体系。“川模型”由 3 条相对独立的测试实施流程组成,包括验收测试实施流程、需求级测试实施流程和业务级测试实施流程。“川模型”突出体现了测试活动对质量把控的重要性,站在测试者的角度,以测试来保证最终用户的质量认可,体现测试先行的意义。

以上这些软件测试模型主要考虑同一测试对象不同级别、不同类型测试的前后衔接与继承关系,但未考虑与同类测试对象间的横向关联。实际上,同类软件测试项目特

别是同一领域的测试项目具有较强的关联性和相似度,其数据具有很高的借鉴意义,必须将相关项目的测试先验数据纳入本项目测试设计,构建启发式的软件测试架构模式,最大程度上吸收已有测试经验,复用已有测试成果,才能满足软件敏捷开发和快速交付要求下,快速增长的测试需求。对于软件测评机构,虽然测试任务存在一定的不确定性和非延续性,但目前看来,随着软件测试领域不断发展成熟,越专业的机构,主体业务越是相对固化,形成了对该专业领域的核心竞争力,竞争力来自经验技术的不断积累。因此,优质的软件测评机构往往高度重视专业领域测评技术总结沉淀,通过测试平台建设,建立测试数据共享机制,使已发现同类项目的典型软件缺陷数据和测试数据等能用于指导后续测试,有效复用测试用例及相关测试资源,规避问题风险,同时利用工程化方法预测软件缺陷<sup>[6-7]</sup>,可以大大提高测试精准性。因此,建立更为高效适用的软件测试架构,必要具备“测试知识”的积累应用能力和数据共享复用能力。

### 1.2 软件测试架构的特性与发展方向

软件测试架构的构建由两方面因素决定:一是被测软件本身的技术特点,二是软件测试资源投向。

从被测软件来看,软件本身的功能、性能、系统架构等决定了软件如何被优质高效地测试,必须针对性制定测试计划,进行测试安排,例如软件功能架构情况决定了软件功能测试项和测试用例的划分、分解和设计,从而确定合理的测试用例集。

从测试资源看,人工设计用例方式受人为因素的影响较大且日益不能满足现有需求<sup>[8]</sup>,必须根据软件测试需求和要求,统筹考虑测试环境、测试工具、测试方法,并引入测试管理系统来提高测试自动化水平。

在线形模型架构基础上,近年来国内也提出以测试过程为中心,软件质量、测试团队、测试技术 3 要素铰链构建的软件测试架构<sup>[9]</sup>,特别是针对专业领域所研究的人员素质、工作流程与模型、测试技术和工具三个因素的公安交通管理信息系统软件测试总体架构<sup>[10]</sup>,以及以高效可信、灵活赋能为目标的雷达产品软件测试框架<sup>[11]</sup>等,均主要从测试管理角度构建,主要为发挥测试团队质量优势,本文提出的软件测试新架构偏重于为发现软件缺陷提高测试质效而构建软件测试技术框架,但这些架构的提出也有一定的借鉴意义。

当前被测软件系统,不但规模上越来越复杂,还在交付周期上不断压缩。其为了迅速应对使用需求和快速部署要求,往往采用分阶段目标设定、分阶段设计实现,分阶段能力生成的研发方式,确保软件能够快速交付,并按计划目标不断迭代优化。在这种情况下,软件测试架构发展呈现以下的特点趋向:

1) 确保测试先行,尽早发现问题,及时了解项目的测试风险,及早制订应对方案,并使软件测试贯穿软件全生命周期;

2) 必须适应软件快速开发要求, 在敏捷测试理念下明确测试重点, 设置合理的测试策略, 缩短软件测试周期, 提高测试效率;

3) 打破软件测试从单元测试到验收测试的传统线性关系, 支持在全寿命周期内根据软件变更进行测试迭代, 以应对软件需求变更调整和自身优化升级;

4) 充分利用历史经验, 基于前序测试知识, 尽可能进行测试用例复用, 从而科学有效地实现分级分类迭代测试。

设计的基于软件、测试、缺陷的软件测试新架构主要着眼于已有软件测试数据的积累应用, 在软件测试已形成的“知识”的启发指导下, 准确应对软件测试需求, 精准投入测试资源, 达到提高软件测试质效的目的。

## 2 基于“三域”铰链的软件测试新架构

当前, 传统的软件测试架构虽然也试图解决迭代测试的问题, 但线性为主的架构方式, 以及不同软件项目间测试信息关联度低, 未能充分挖掘数据信息内在规律等现实情况, 不可避免地导致大量软件测试工作简单重复、无法跨项目复用历史用例等。而之所以出现软件测试数据无序累积, 无法共享应用, 不能实现优化迭代测试等问题, 一个重要的原因就是缺少测试信息共性基础标准, 未形成以共享复用为基础的启发式测试体系架构。

本文提出的软件测试新架构, 在重新梳理软件测试信息及其模型构建的基础上分析认为, 要在传统的软件测试架构基础上, 重点针对由于软件测试数据采编标准不统一、存储管理无序化、自动处理落地难, 以及缺乏专业信息系统对软件测试数据进行常态化积累与常态化应用等问题, 着力推动软件测试数据共享复用、凝结测试经验知识, 支撑迭代性测试。以软件测试数据的采存管理、异构融合、分析挖掘、交换共享等为基点, 提出了基于软件、测试、缺陷“三域”铰链的“知识化”软件测试新型架构, 通过基于测试的软件缺陷参数化表述模型, 形成了软件测试/试验数据关联关系图谱, 突破了软件试验数据聚类重组技术和软件典型缺陷模式提取技术, 实现了软件测试从随机发散型向缺陷特征规律启发型转变, 推动软件测试更加精准深入高效。

### 2.1 软件、测试、缺陷“三域”

按照 CCSE2004 发布的报告, 软件工程学科由 10 个知识域构成: 软件需求、软件设计、软件构造、软件测试、软件维护、软件配置管理、软件工程管理、软件工程过程、软件工程工具和方法、软件质量。对软件测试信息的有效收集、表示和存储, 不仅能为重复测试提供方便<sup>[12]</sup>, 但其根本目的是通过软件测试方法改进, 精准发现软件缺陷<sup>[13]</sup>, 快速提高软件质量。在传统软件工程知识域的基础上, 以建立软件测试新型架构为目标, 测试信息划分为 3 个独立的域: 被测软件域、测试域、软件缺陷域<sup>[14]</sup>, 在“三域”铰链下对软件测试信息进行一系列“知识化”处理, 形成软件测试领域知识模型<sup>[15]</sup>, 最终制导构建软件一体化测评框架<sup>[16]</sup>。国内提出基于软件缺陷知识的测试框架<sup>[17]</sup>, 而这

里提出的三域铰链“知识化”下的软件测试新架构更利于表述复用共享高效的测试解决方案。

### 2.2 “三域”铰链下的软件测试架构

软件、测试、缺陷“三域”中, 被测软件域和测试域之间, 在分析软件测试信息时, 以历史软件测试项目的数据作为基础, 并参考软件测试项目数据, 对测试信息进行自顶向下、层层细化的详细分析, 并通过测试人员及开发人员不同的视角进行, 从而得出各种分散的测试信息知识<sup>[18]</sup>, 构建跨项目测试框架<sup>[19]</sup>; 其次, 测试域与软件缺陷域之间, 测试需求、测试用例和测试发现的故障之间的映射关系具有复杂的多对多对应关系<sup>[20]</sup>。在系统建立的测试数据库的基础上, 进行合理的统计分析和数据挖掘, 获得需要的数据<sup>[21]</sup>。把知识库的结构引入到软件缺陷管理中, 把缺陷分析从数据管理层面提升为知识管理层面, 提取规则知识, 并生成能指导软件测试工作的策略<sup>[21]</sup>; 再次, 被测软件域与软件缺陷域之间, 开展软件故障模式分析, 针对特定软件的常见故障, 归纳故障特征(包括故障类型、原因、现象等), 形成软件故障模式库<sup>[22]</sup>。目的在于根据故障模式生成可能触发软件故障的异常测试数据, 并施加于被测软件, 从而对其进行针对性的测试。

软件测试新架构构建软件、测试、缺陷三域铰链的测试信息模型, 如果没有一致的信息模型定义, 数据的交互和重用难度很大。测试信息模型要求通过创建一致的信息模型和信息接口, 把与测试相关数据进行信息的模型化, 消除不同域内数据和信息的表达差异, 实现装备软件、测试资源和测试过程和测试结果各个域内所有测试数据的共享和信息的有效融合, 从而切实提高装备软件的测试与诊断能力。

## 3 关联知识启发式的新型测试架构详细设计

通过总结“三域铰链”的软件测试信息规律, 根据当前软件测试业务从随机性向“知识化、启发式”的发展需求, 提出了基于软件、测试、缺陷“三域”铰链的软件测试技术架构(见图 2)。

该软件测试新架构的主要内容是:

### 3.1 信息表征模型化

通过建立统一软件测试技术框架和测试描述语言规范, 实现软件典型缺陷与测试样例的描述标准化、结构参数化(见图 3), 消除不同域内软件测试数据的表达和组织差异, 为软件测试数据“采存管用”规范化、标准化、常态化奠定技术基础, 提供由研发、测试等各阶段、各方共享, 用于测试的信息框架, 实现测试信息的有效融合和复用。

建立基于测试视角的参数化软件测试信息表述模型(见图 4), 以测试视角为数据处理主要原则, 将软件、软件测试项目、软件缺陷等信息进行参数化描述, 建立信息模型, 实现软件测试数据、缺陷数据的聚类重组。

### 3.2 缺陷数据知识化

对于某一装备领域比如说软件密集的电子信息装备等, 软件缺陷呈现重复性、规律性。软件典型缺陷与测试样例

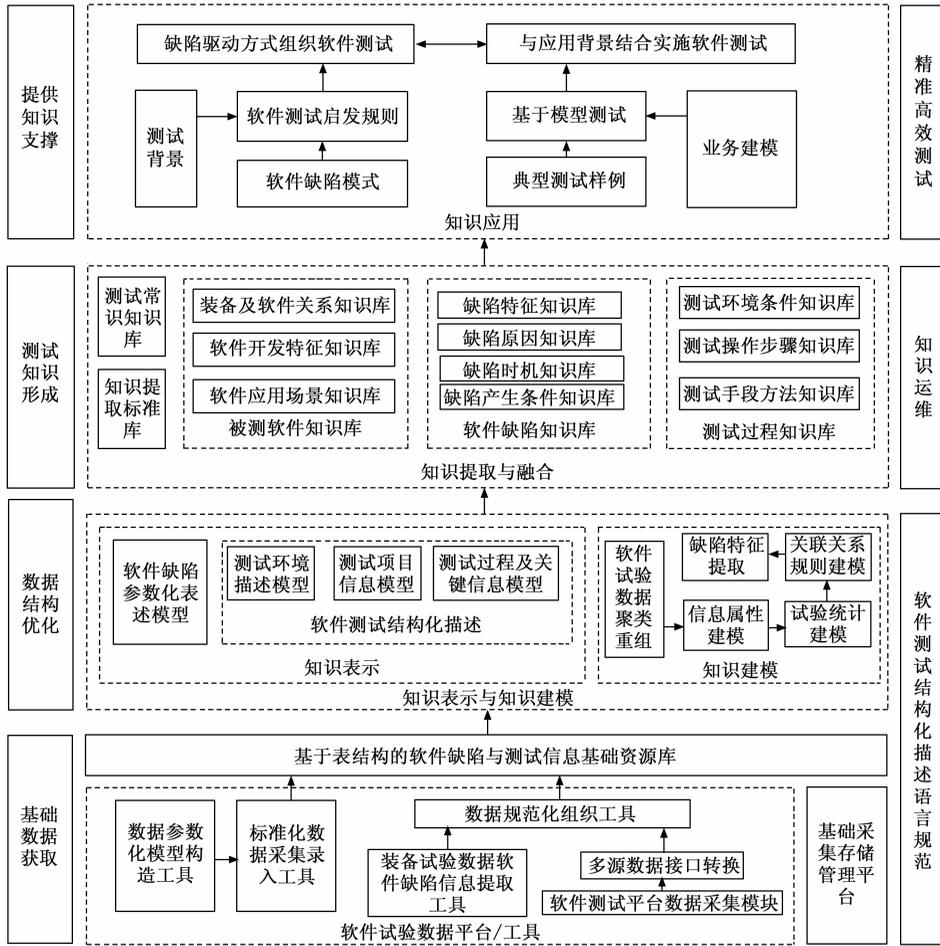


图 2 基于软件、测试、缺陷“三域”铰链的软件测试技术架构

库通过对软件缺陷数据按“层次”进行聚类重组，按软件典型缺陷模式提取软件缺陷知识，显式表述软件缺陷及测试过程中呈现的规律，用于指导装备软件的研发、测试等工作，避免软件历史缺陷及类似缺陷的重复发生。

将以往的软件缺陷数据按照测试视角以及应用分层的思想进行聚类重组，提取模式化的软件典型缺陷（见表 1）。

在面向测试视角的软件缺陷信息模型基础上，统一规范软件缺陷数据和测试信息，消除不同域内数据信息的表达差异，并易于分析软件缺陷特征规律，实现软件精准测试。

1) 测试需求分析精准化：主要依托被测软件知识库实现，利用软件开发特征知识库知识分析被测软件技术特点；利用软件应用场景知识库知识，分析测试背景，确定测试重点内容；利用装备及软件关系知识库，抽取装备软件间继承依赖关系，分析历史信息 and 测试轨迹，优化调整测试策略。

2) 测试策划精准化：主要依托测试过程知识库，尤其是其中的测试环境条件知识库，根据被测软件实际情况，既可利用依赖项目抽取相关信息建立初始环境工具数据集，也可利用跨项目比对，对已有测试环境搭建需求提报和测

试工具选择进行精准化的调整。

表 1 跨越凌晨(0:00)的软件典型缺陷模式示例

序号	按编码错误归类的软件缺陷模式
1	时间精度处理中，考虑了时间变化，未考虑日期变化
2	时间数据处理中，24:00 与 0:00 未进行转换导致错误
3	以天为单位报表统计处理中，起始时间跨越凌晨的统计算法错误
...	...
序号	按结果归类的软件缺陷模式
1	软件跨越凌晨连续运行，业务流程中数据覆盖前一天数据
2	软件跨越凌晨连续运行，软件当前处理数据丢失，流程中断
3	软件跨越凌晨运行，统计报表中的数据计算结果错误
...	...
序号	按现象归类的软件缺陷模式
1	软件连续运行至 0:00 后，正常运行过程中突然死机
2	软件连续运行至 0:00 后，软件运行界面反复切换，无法操作
3	软件连续运行至 0:00 后，无法进入主界面
...	...

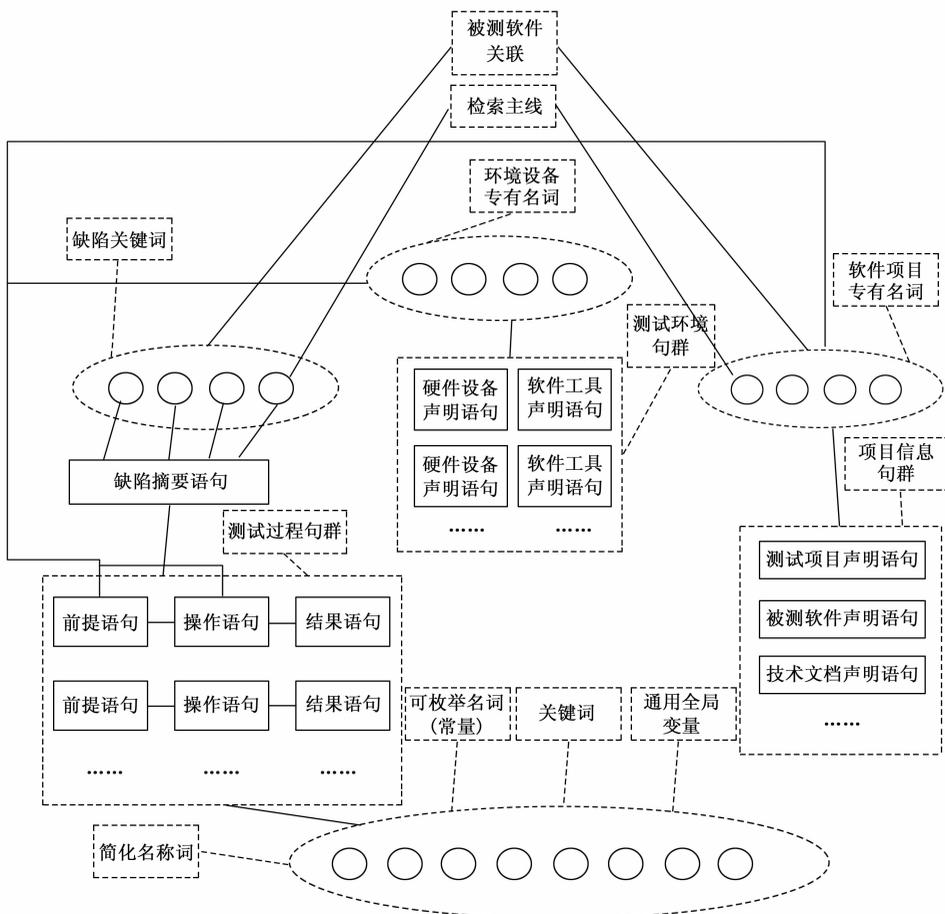


图 3 标准化的软件测试信息

3) 测试设计精准化: 主要依托软件缺陷知识库和测试过程知识库, 根据软件质量理论, 功能类似、同一领域软件的缺陷经常呈现重复性、聚集性等规律特征, 在航空航天、电子信息装备等特定专业领域中的软件进行测试时, 可以通过知识化的缺陷数据和缺陷规律, 通过吸收测试经验, 复用测试用例, 借鉴测试方法和测试手段应用, 指导软件测试过程, 充分发挥数据驱动作用和倍增效应, 实现测试投向的精准化。

4) 测试执行精准化: 主要依托被测软件知识库、软件缺陷知识库与测试过程知识库的关联关系, 吸收以往测试执行经验, 以测试数据采集、处理、分析为中心, 利用知识反馈实现测试迭代, 实现更有效的测试过程管理。

5) 测试总结精准化: 主要依托软件缺陷知识库, 评价遗留问题危害和后果, 同时依托被测软件知识库, 进行同类装备软件质量的横向和纵向比较, 真实反映当前软件的质量情况。

#### 4 软件测试新架构的应用

要应用软件测试新架构, 离不开软件测试信息系统和数据共享平台的支撑, 必须实现软件测试信息系统与测试新架构的有效融合。通过建立统一的技术框架和信息表述

规范, 并在此基础上设计软件测试信息管理系统, 实现软件测试数据、缺陷数据的完整准确留存和优质高效共享, 为加强软件测试设计、优化组织实施提供有效数据保障。

首先通过基于测试的软件缺陷参数化表述模型和软件测试结构化描述语言规范, 按照因素结构分析法, 区分固定变量、枚举型变量、自由变量等要素设计数据采集界面, 以及多源数据的融合整编, 构建了软件典型缺陷与测试样例共享平台。

其次实现从软件缺陷基本特征到各要素分类表述的数据组织, 对软件测试/试验数据进行聚类重组。新的重组方式要更易按照软件缺陷进行检索和统计, 形成测试知识, 实现数据标准化采编和共享复用, 为软件测试任务提供数据支撑。

再次是软件测试信息管理系统的设计与实现。根据“被测软件”、“测试”、“软件缺陷”三域主题, 确定主题应该包含的数据。虽然各软件测评机构使用的测评实验室管理系统保存着大量测试数据, 但这些数据不能等同于围绕软件缺陷主题的数据。“软件监测人”、“测试的具体日期”等对软件缺陷没有直接的影响, 可在系统中进行字段切除。要将关于某一主题的、散落在其他软件测试实验室管理系统中的信息组织进来。例如“软件缺陷”还存在于试验、

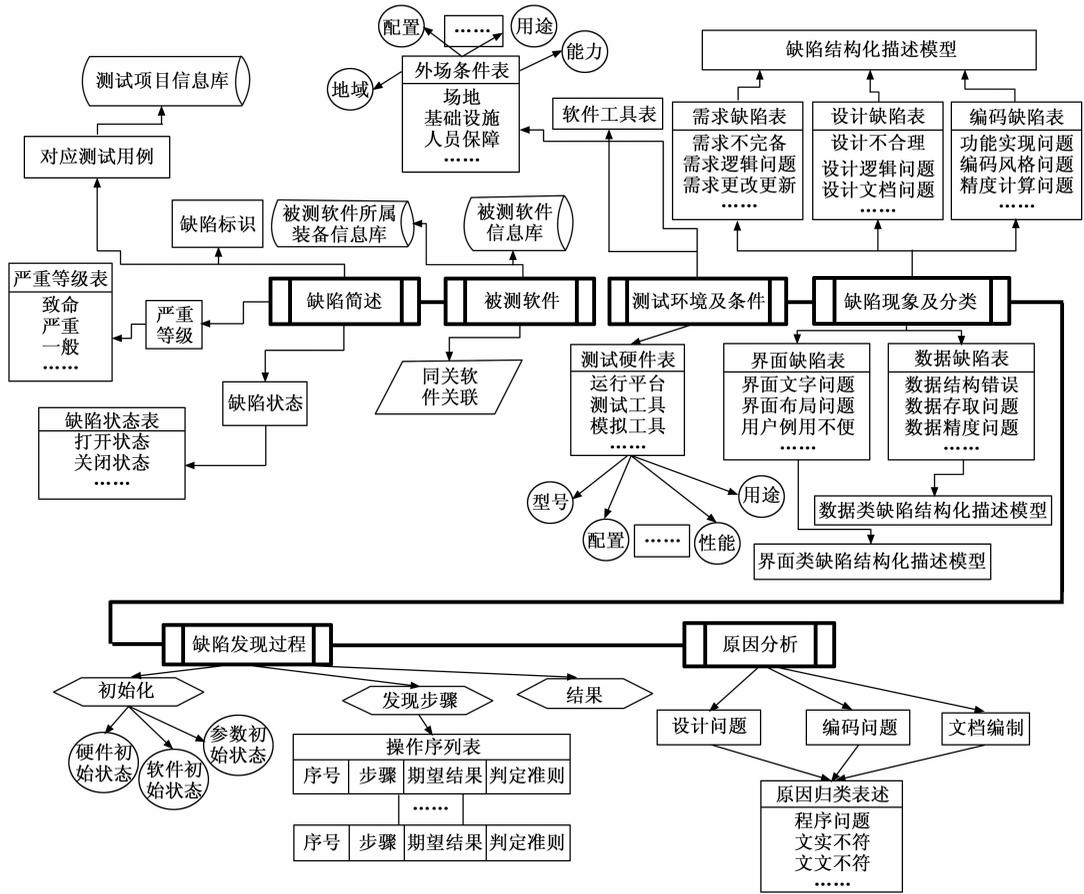


图 4 基于测试视角的参数化软件测试信息表述模型

研制的相关数据系统中，在进行软件缺陷的分析和统计时，需要了解这一信息，由于这些信息包含了更多的装备实际外场环境和使用背景，利用价值很高，就必须将其组织进来。不同的主题之间可能会出现相互重叠的信息，例如“软件缺陷”与“测试项目”主题在具体的测试结果和测试过程信息方面有很多相互重叠的信息，这种重叠信息往往来源于两个主题之间的联系，例如，“软件缺陷”主题与“测试项目”主题在测试项目编号和测试过程方面的相互重叠来源于软件测评的实验室管理系统及相关的数据存储处理系统。因此需要围绕“软件缺陷”这一主题重新进行数据的组织。在数据组织基础上采用软件仓库的相关工具对软件测试信息管理系统（见图 5）进行了设计和开发。

最后，应用软件测试信息管理系统和软件测试新架构，进行测试需求辅助分析，通过跨项目经验借鉴优化调整测试策略，快速完成软件测试设计，利用先验缺陷知识和测试知识指导用例设计，通过跨项目关联明确测试用例复用和筛选策略，在测试执行中记录重要和关键现象，与预期结果进行比对，在缺陷知识支持下进行测试信息分析和缺陷匹配，确定软件缺陷，分析定位原因，同时以质量数据和质量模型为基础评估软件质量。

在实际工作中，建立了电子信息装备软件典型缺陷与



图 5 软件测试信息汇集管理界面

测试样例平台和测试信息系统，收录了十多年的试验/测试数据，已在多项测试任务和研制项目中得到应用。在测试信息系统基础上的软件测试新架构，应用于各项测试任务中，为改进传统软件测试方式，精准投放测试资源，提高测试质效方面发挥作用明显，在某型侦察系统中将软件测评周期从 6 个月缩减至 3 个月。特别在某型装备中根据已服役类似装备的相关缺陷知识，发现多年来一直未被发现的信号分选问题。并根据信号侦收不稳定相关的测试数据分

（下转第 121 页）