Embedded Technoloy and Intelligent Systems 嵌入式技术与智能系统, 2025, 2(5), 330-336
Published Online October 2025 in Hans. https://www.hanspub.org/journal/etis
https://doi.org/10.12677/etis.2025.25032

Hans汉斯

基于提示工程的软件测试用例生成方法研究

王　媛，韩　启，张　絮，杨丰辉

航空工业第一飞机设计研究院，陕西　西安

收稿日期：2025年12月24日；录用日期：2026年1月16日；发布日期：2026年1月29日

摘　要

软件测试用例的质量直接关系被测软件的质量。在当前大语言模型飞速发展的阶段，人们运用提示工程等关键技术来生成软件黑盒测试用例。然而，普遍存在生成的测试用例质量参差不齐，尤其异常场景测试用例无法完全契合实际测试需求的问题。因此，本文针对座舱显示控制软件进行了结构化提示词设计方法的研究。通过专家打分法以及结合软件代码缺陷检测率，对比不同结构化提示词因子组合下的测试用例生成效果，从而得出最佳结构化提示词，有效提升了软件测试用例的生成质量与效率。

关键词

大语言模型，提示工程，黑盒测试，座舱显示控制软件，测试用例生成

Research on Software Test Case Generation Method Based on Prompt Engineering

Yuan Wang, Qi Han, Xu Zhang, Fenghui Yang

AVIC The First Aircraft Institute, Xi'an Shaanxi

Received: December 24, 2025; accepted: January 16, 2026; published: January 29, 2026

Abstract

The quality of software test cases directly impacts the quality of the software under test. In the current era of rapid development of large language models, prompt engineering and other key technologies are being applied to generate black-box test cases. However, a common issue is the inconsistent quality of generated test cases, particularly in abnormal scenarios, which often fail to fully meet actual testing requirements. To address this, this study focuses on the research of structured prompt design methods for cockpit display control software. By comparing the effectiveness of different structured prompt factor combinations in test case generation through expert scoring and software code defect detection rates, the optimal structured prompt is derived, effectively improving the quality and efficiency of software test case generation.

文章引用: 王媛, 韩启, 张絮, 杨丰辉. 基于提示工程的软件测试用例生成方法研究[J]. 嵌入式技术与智能系统, 2025, 2(5): 330-336. DOI: 10.12677/etis.2025.25032

Keywords

Large Language Model, Prompt Engineering, Black-Box Testing, Cockpit Display Control Software, Test Case Generation

Copyright © 2025 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

软件测试用例编写是测试工作的核心环节，测试用例的质量直接影响软件缺陷的检测率，对保障软件质量起着关键作用。软件测试技术可分为白盒测试技术和黑盒测试技术。白盒测试基于源代码结构，随着人工智能的发展，相关测试用例生成方法的研究不断涌现且成效显著，例如基于遗传算法、蚁群算法以及粒子群优化算法等生成高覆盖率的测试用例[1]。

而在黑盒测试方面，通常采用 UML 等基于模型的测试用例生成技术。对于需求繁杂且软件规模较大的座舱显示控制软件而言，通过人工绘制 UML 图来生成测试用例的方法，与传统手工编写用例相比，其工作效率并未显著提升[2]。因此，大部分座舱显示控制软件仍然采用手工编写用例的方式。然而，随着软件高频次迭代升级的需求增加，测试用例的生成效率变得低下；而且测试用例的设计严重依赖于测试人员的经验积累，导致生成测试用例的质量参差不齐。

近年来，随着大语言模型(Large Language Model, LLM)的快速发展，其在文本理解、文本生成等方面表现突出，在测试领域也表现出极大的潜能。如何运用模型微调、提示工程(Prompt Engineering)和检索增强生成等 LLM 关键技术，使大语言模型更好地应用于测试领域，也是当前研究的热点方向。目前，在实际工程应用中，人们尝试编写提示词以开展基于 LLM 的黑盒测试用例生成工作。然而，大语言模型的性能极易受提示词的格式、精准度、约束等因素影响。因此，本文针对座舱显示控制软件进行结构化提示词设计方法研究，以提升 LLM 生成黑盒测试用例的效率和质量。

2. 提示工程技术研究现状

提示工程因具有无需对模型参数进行调整，便能增强大语言模型理解与推理能力的优势，广泛应用于多个业务领域。其核心在于设计离散或连续的提示模板，将任务目标转化为自然语言指令序列并输入大语言模型，有效缓解了大语言模型存在的幻觉、知识迁移等问题。

主流的提示工程技术包括无训练提示、逐步思考提示、自动提示和检索增强提示等。零样本提示和少样本提示作为无训练提示的核心技术，通过设定角色、明确任务、限制风格等优化手段，为特定领域的复杂任务提供高效简便的解决方案。思维链提示、自一致性提示和思维树提示方法，通过将复杂问题拆解为若干子问题，引导大语言模型逐步思考得到最终答案，提高了大语言模型的回答准确率。自动提示工程中的元提示利用大语言模型生成提示词，而无梯度指令提示搜索方法在轮次迭代中通过评分函数获取最优提示。检索增强提示方法通过在外挂知识库中检索相关信息，丰富输入大语言模型的上下文信息，在复杂多变的任务中同样表现出色[3]。

由于黑盒测试用例的构成元素较为复杂，包括输入数据、预期结果和操作步骤等，难以对其进行量化评估。以及结合黑盒测试用例生成的任务特点，本文基于零样本、少样本、思维链和元提示等，进行基于提示工程的座舱显示控制软件测试用例生成方法研究。

3. 本文研究内容

3.1. 研究方法

提示工程中的提示词即输入大语言模型的指令，简单无结构的指令如“请根据输入需求，帮我生成测试用例”，往往无法得到预期输出，甚至在连续两次输入同样的指令时会得到完全不同的输出。因此，设计好的提示词，应具备结构合理、指令明确、输出可控等特性。

通用的提示词模板，如“角色－任务－行动”等，在特定任务中的表现并非最佳。例如“你是一位经验丰富的测试用例设计专家，请依据输入需求，使用异常场景测试方法……生成测试用例”，实际模型输出的异常场景测试用例中，将参数显示的状态值当作连续数值进行异常场景设计，与实际测试需求不符。因此，设计提示词时需结合被测软件特点，对大模型回答进行约束或补充更详细的测试规则等。

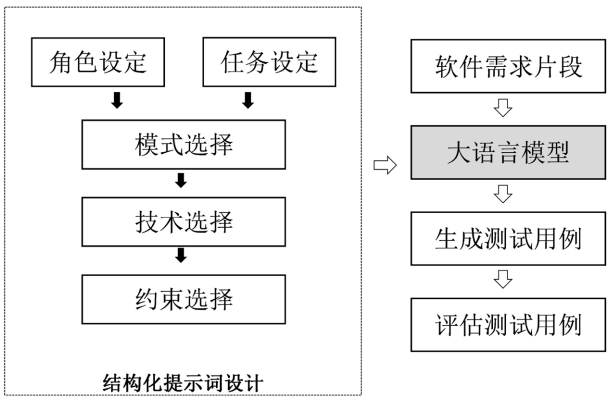


Figure 1. Research process for test case generation methods
图 1. 测试用例生成方法研究流程

座舱显示控制软件常见于汽车领域和航空领域，软件主要模块包括数据交互模块、画面逻辑管理模块[4] [5]。数据交互模块，用于完成外围设备数据与软件图形界面之间的数据传输，以及用户与软件图形界面之间的数据传输，确保软件的实时响应功能。画面逻辑管理模块，用于完成图形界面内部逻辑控制，实现画面配置切换时图形界面在不同显示分区上的显示。座舱显示控制软件本质为 GUI 软件，由各类控件组成图形界面显示元素，控件的交互状态可分为状态显示和状态下发。

Table 1. Structured prompt design
表 1. 结构化提示词设计

元素		因子/关键词
角色		*r*：你是一位拥有 30 年经验的软件测试专家
任务		*g*：为用户给定的功能需求，生成测试用例集
技术		*s*1：零样本提示
		*s*2：少样本提示
		*s*3：思维链提示
		*s*4：元提示
约束	任务边界	*cg*：运用功能分解法、等价类划分法、场景法、错误推测法和状态迁移法等方法设计测试用例
	方法边界	*cs*1：人机交互界面测试补充规则

续表

	*cs*2：接口测试补充规则
	*cs*3：边界值测试补充规则
	*cs*4：异常场景测试补充规则
	*cs*5：功能测试补充规则
质量约束	*cq*：测试用例应满足100%需求覆盖率，应步骤清晰可执行，预期结果明确，功能划分独立、不冗余
格式约束	*cf*：用例标识、名称、说明、输入数据、操作步骤、预期结果等

基于座舱显示控制软件的上述组成特点，在结构化提示词中设计以下约束内容，如表1所示。

(1) 人机界面测试补充规则：a. 外观测试，测试元素的字体/形状、大小、颜色、；b. 布局测试，测试元素的布局、位置、对齐；c. 画面配置测试，测试画面配置切换时元素的外观、布局显示。

(2) 接口测试时补充规则：a. 显示数据流测试：元素数据来源设备的各数据状态显示测试；b. 控制数据流测试：用户操作元素为不同状态时的状态下发测试。

(3) 边界值测试时补充规则：a. 合法值测试，包括最大值、最小值、最大值 − 步长、最小值 + 步长等；b. 非法值测试，包括最大值 + 步长、最小值 − 步长、非法数据类型、非法小数位数、非法负值、以及空值等。

(4) 异常场景测试时补充规则：a. 数据异常测试，测试数据的状态为故障、无效、未定义时的显示；b. 外围设备下电测试，测试数据来源设备为下电状态时、异常下电并重启后的显示。

(5) 功能测试时补充规则：a. 测试计时功能时，测试用例应至少包括以下场景：1) 画面显示计时直至计时结束；2) 计时过程中切换其他类型画面，计时结束后再切回原画面显示；3) 计时过程中切换其他类型画面，计时结束前切回原画面显示；4) 计时过程中画面切换至其他位置直至计时结束；5) 计时过程中画面切换至其他位置，计时结束前切回原位置显示。b. 测试叠加功能时，测试用例应至少包括以下场景：1) 画面叠加元素显示时取消叠加；2) 画面叠加元素显示时，切换为其他类型画面后再切回原画面；3) 画面叠加元素显示时切换至其他位置显示。c. 其他特殊需求测试补充规则。

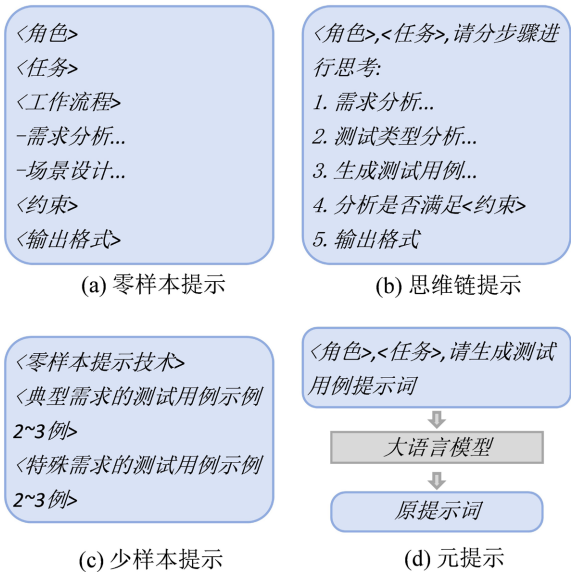


Figure 2. Explanation of structured prompting techniques
图 2. 结构化提示词技术说明

结合上述内容，本文在结构化提示词中添加以下要素：角色、任务、提示工程技术、以及针对座舱显示控制软件的任务边界、方法边界、质量约束和格式约束等。本文研究过程如图 1 所示，依据各要素选择不同的因子进行结构化提示词构建，然后对大语言模型生成的测试用例进行评估，最终得到最佳结构化提示词。构建的结构化提示词中使用的关键提示工程技术如图 2 所示。

3.2. 评估方法

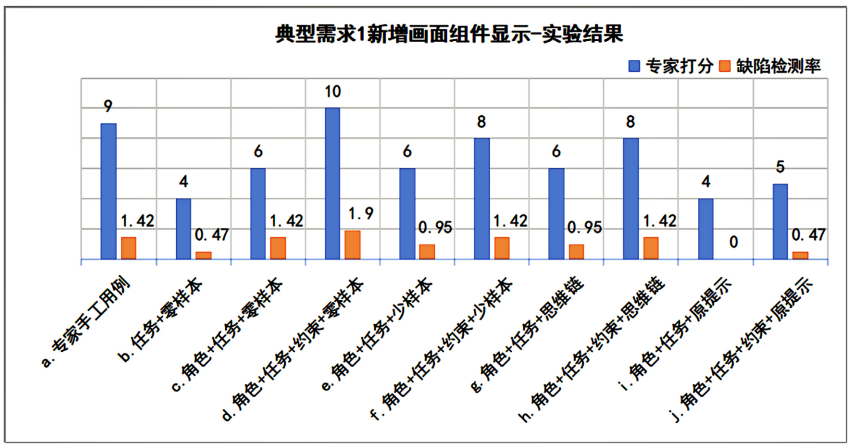
依据军用软件测试指南[6]，高质量的测试用例应满足 100%需求覆盖率，且具备完整性，即包含用例名称、用例标识、用例说明、前置条件、测试输入、预期结果和操作步骤等要素。

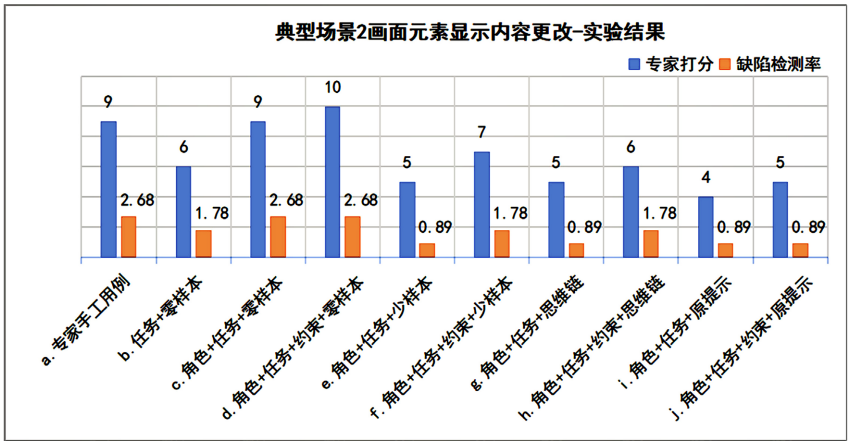
邀请 5 位在座舱显示控制软件测试领域经验丰富的测试专家，结合上述内容，将测试用例的评价标准划分为以下维度：a. 需求覆盖率。b. 用例完整性。c. 用例准确性。d. 异常场景覆盖率，对测试用例进行打分评价。同时，执行大语言模型生成的测试用例，对比实测结果与预期结果，结合代码更改规模得出 e. 软件代码缺陷检测率，计算公式为：软件代码缺陷检测率 = 测试问题个数/千行代码 × %。

3.3. 实验分析与讨论

以某军用座舱显示控制软件为被测对象，在其需求规格说明文档中抽取 3 类典型需求：新增画面组件显示、画面元素显示内容更改、画面元素显示逻辑更改；以及抽取 2 类特殊需求：画面新增计时显示、画面新增叠加显示。对每类需求整理 6 条需求，共 30 条需求。

由于 DeepSeek 是国内领先的大语言模型之一，其中 R1 系列专注于复杂任务的推理工作。因此，在本地部署 DeepSeek R1 32B 模型进行本文相关研究工作。





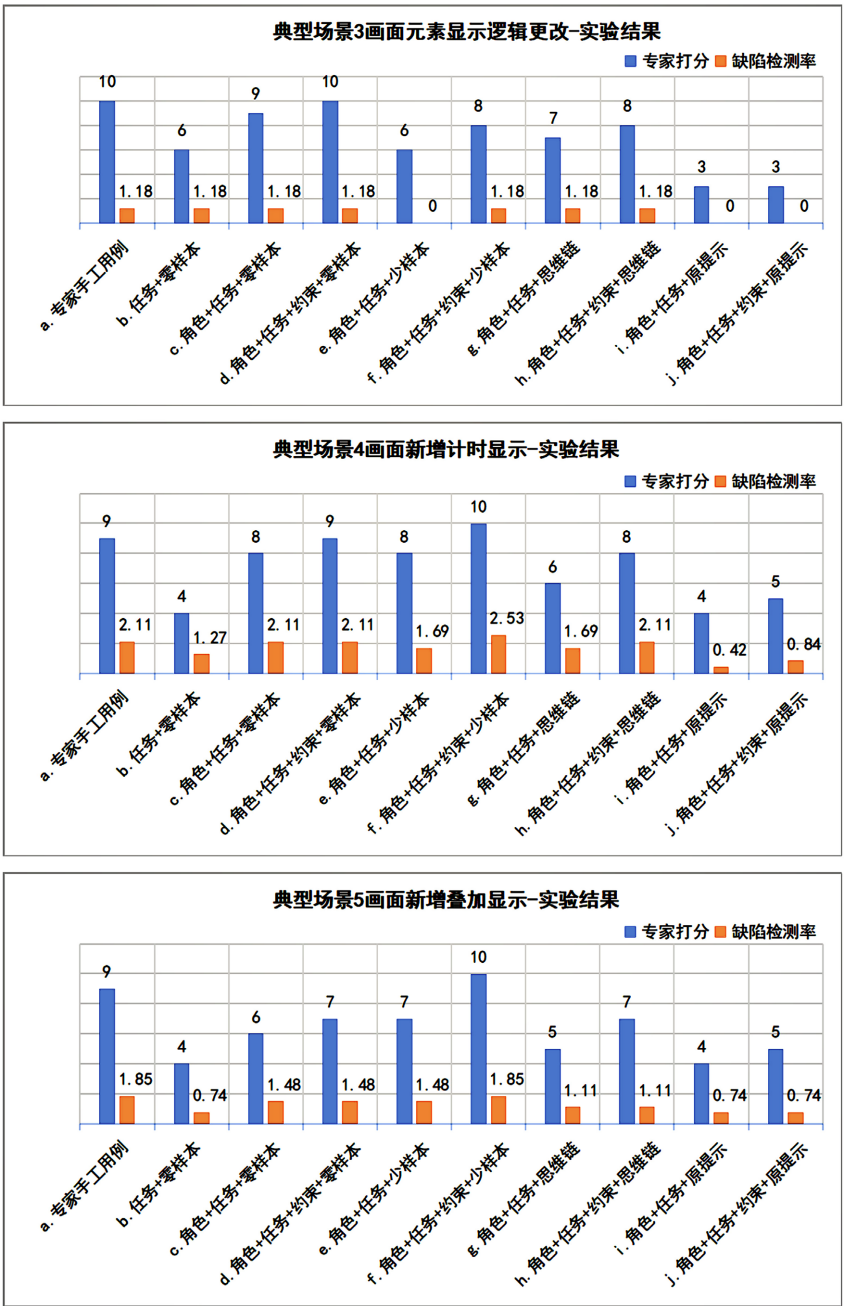


Figure 3. Comparison of test case generation results
图 3. 测试用例生成结果对比

将 30 条软件需求和结构化提示词组合因子作为大语言模型的输入，生成测试用例，专家对其进行打分评价。并且依据 30 条需求对应的软件代码更改规模 9153 行，以及执行用例得到的问题数，根据公式计算软件代码缺陷检测率，实验结果如图 3 所示，左列数据表示专家平均打分，右列数据表示软件代码缺陷检测率。

分析上述实验结果，在典型需求的测试用例生成中，“d. 角色 + 任务 + 约束 + 零样本”结构化提示词表现最佳，专家评分和软件缺陷检测率均为最高，在测试执行中发现了更多的软件问题。而在特殊需求的测试用例生成中“f. 角色 + 任务 + 约束 + 少样本”结构化提示词表现最佳。

专家手工编写用例的在所有需求场景下的评分均略低于最佳结构化提示词生成用例的评分。实验发现即使是测试专家在手工编写用例时，也会不可避免地出现笔误和格式问题，导致在用例执行时需要对用例进行再次修正，增加了人力成本和时间成本。

零样本技术在典型需求的测试用例生成中表现优秀，结合约束因子能覆盖正常、边界测试场景以及绝大部分的异常场景。少样本技术因提示词中的示例具有强针对性，在特殊需求的测试用例生成中表现良好。思维链提示技术是针对复杂逻辑推理提出的方法，因此在测试用例生成领域的表现并未明显优于其他方法，在典型需求和特殊需求中表现较为均衡。而原提示技术由于提示词由大语言模型生成，其测试类型明显少于零样本等方法中设置的测试类型，存在冗余且低关联需求的用例，因而在所有需求场景中表现较差。

约束因子是结构化提示词中的关键因子。当结构化提示词中未添加约束时，所生成用例的评分和软件缺陷检测率，均明显低于有约束的结构化提示词生成效果。甚至在表现不佳的原提示技术中，添加约束时也能明显提高测试用例的生成质量。因此，设计符合软件特点的约束因子，是提高大语言模型在黑盒测试用例生成质量的关键手段。

4. 总结与展望

本文结合座舱显示控制软件的功能特点，设计该软件的结构化提示词，并在提示词中添加约束等因子。探究如何结合结构化提示词中的元素和因子信息以提高生成测试用例的覆盖率、准确率和合格率，最终得到座舱显示控制软件的最优结构化提示词。

人工对比和评估各结构化提示词的测试用例生成效果仍需耗费一定的人力，以及通过人工执行用例来评估软件缺陷检测率的方式也效率低下。在后续工作中，可与自动化测试方法相结合，自动评估生成测试用例的质量，利用反馈机制等，动态选择及生成结构化提示词，得到被测软件的最佳结构化提示词。

参考文献

[1] 王延永, 黄松. 测试用例自动生成技术综述[J]. 电子技术与软件工程, 2021(18): 51-53.

[2] 杨波, 吴际等. 一种软件测试需求建模及测试用例生成方法[J]. 计算机学报, 2014, 37(3): 522-538.

[3] 黄峻, 林飞等. 生成式 AI 的大模型提示工程: 方法、现状与展望[J]. 智能科学与技术学报, 2014, 6(2): 115-133.

[4] 刘硕, 林荣超. 综合座舱显示控制系统的设计与实现[J]. 现代电子技术, 2010, 33(15): 160-162.

[5] 王华荣, 宁伟. 综合显示控制系统的设计与实现[J]. 科技资讯, 2011(11): 36.

[6] 中国人民解放军总装备部司令部. 军用软件测试指南: TE-BTCG-00302021[S]. 北京: 总装备部军标出版发行部, 2021.