文档编号：SY00011_ISTQB_CT_GenAI_EN1.0_CN1.1

# 生成式人工智能测试
# 认证测试工程师
# (CT-GenAI)
# 大纲

版本：EN1.0_CN1.1

发布日期：2026 年 04 月 09 日

国际软件测试认证委员会

中文版的翻译、编辑和出版统一由 ISTQB®授权的 CSTQB®负责

若您对此文档有任何问题，欢迎您扫码添加【官方微信号】反馈。

# 版权声明

# 修订历史

| 版本 | 日期 | 备注 |
| --- | --- | --- |
| v1.0 | 2025/07/25 | CT-GenAI v1.0 版本发布。 |
| EN1.0_CN1.0 | 2026/04/07 | CT-GenAI v1.0 中文版常规发布版本。 |
| EN1.0_CN1.1 | 2026/04/09 | 部分修订与术语表对齐。 |

# 目录

# 致谢

本文档由 ISTQB®大会于 2025 年 7 月 25 日正式发布。

本文档由以下国际软件测试认证委员会成员组成的团队完成：Abbas Ahmad（产品负责人），Gualtiero Bazzana，Alessandro Collino，Olivier Denoo，以及 Bruno Legeard（技术经理）。

团队感谢 Anne Kramer，Jedrzej Kwapinski，Samuel Ouko 和 Ina Schieferdecker 的技术评审，并感谢评审团队及各成员委员会提出的建议与意见。

以下人员参与了本大纲的评审、评论与投票：

Albert Laura，Aneta Derkova，Anne Kramer，Arda Ender Torçuk，Baris Sarialioglu，Claire Van Der Meulen，Daniel van der Zwan，Derek Young，Dietmar Gehring，Francisca Cano Ortiz，Gary Mogyorodi，Gergely Ágnecz，Horst Pohlmann，Ina Schieferdecker，Ingvar Nordström，Jan Sabak，Jaroslaw Hryszko，Jedrzej Kwapinski，Joanna Kazun，Karol Frühauf，Katalin Balla，Koray Yitmen，Laura Albert，Linda Vreeswijk，Lucjan Stapp，Lukáš Piška，Mario Winter，Marton Siska，Mattijs Kemmink，Matthias Hamburg，Meile Posthuma，Michael Stahl，Márton Siska，Nele Van Asch，Nils Röttger，Nishan Portoyan，Piet de Roo，Piotr Wicherski，Péter Földházi，Péter Sótér，Radoslaw Smilgin，Ralf Pichler，Renzo Cerquozzi，Rik Marselis，Samuel Ouko，Stephanie Ulrich，Stuart Reid，Tal Pe'er，Tamás Gergely，Thomas Letzkus，Wim Decoutere，Zsolt Hargitai，Mark Rutz，Patrick Quilter，Earl Burba，Taz Daughtrey，Judy McKay，Randall Rice，Thomas Adams，Tom Van Ongeval，Sander Mol，Miroslav Renda，Geng Chen，Chai Afeng，Xinghan Li，Klaudia Dussa-Zieger，Arnd Pehl，Florian Fieber，Ray Gillespie，József Kreisz，Dénes Medzihradszky，Ferenc Hamori，Giorgio Pisani，Giancarlo Tomasig，Young jae Choi，Arnika Hryszko，Andrei Brovko，Ilia kulakov，Praveen，Kostas Pashalidis，Ferdinand Gramsamer，A. Berfin Öztaş，Abdullah Gök，Abdurrahman AKIN，Aleyna Zuhal IŞIK，Anıl Şahin，Atakan Erdemgil，Aysel Bilici，Azmi YÜKSEL，Bilal Gelik，Bilge Yazıcı，Burak Gel，Burcu ÖZEL，Büşra İlayda Çevik Köken，Can Polat，Canan Ayten Dörtkol (Polat)，Cansu Mercan Daldaban，Denizcan Orhun Karaca，Didem Çiçek Bay，Duygu Yalçınkaya，Efe Can Yemez，ELİF CERAV，Emine Tekiner，Emre Aman，Emre Can Akgül，Esra Kücük，Gençay GENÇ，Gül Çalışır Açan，Gül Nihal SİNGİL，Güler GÖK，Gulhanim Anulur，Hakan GÜVEZ，Haktan Bilgehan Dilber，Halil Ibrahim Tasdemir，Hasan

Küçükayar, Hatice Erdoğan, Hatice Kübra Daşdoğan, Hüseyin Sevki ARI, Hyulya Gyuler, İLKNUR NEŞE TUNCAL, Kaan Eminğlu, Kamil Isik, Koray Danışman, Melisa Canbaz, Merve Guleroglu, Müjde Ceylan, Mustafa Furkan CEYLAN, Nergiz Gençaslan, Nuh Soner Bozkurt, Omer Fatih Poyraz, Onur Ersoy, Özlem Körpe, Özgür Özdemir, Sedat YOLTAY, Selahattin Aliyazıcıoğlu, Sevan Lalikoğlu, Sebastian Malyska, Sevim Öykü Demirel, Tatsiana Beliai, Tayg.

本大纲 1.0 版本的中文版本由 ISTQB®成员国委员会 CSTQB®负责并于 2026 年 4 月 7 日正式发布。

本大纲 1.0 版本的中文版本由 CSTQB®本地化工作组团队制作完成，参加翻译和评审工作的成员有（按姓氏拼音排序）：白宇、陈飞、贺炘（组长、QA 评审）、李健、刘海英、刘晓更、叶岚、张喆、周杨虹。

# 引言

## 0.1 大纲目的

本大纲是国际软件测试认证委员会-生成式 AI 测试（CT-GenAI）认证考试的基础。ISTQB®提供此大纲的用途如下：

1. 提供给成员委员会，将其翻译成本地语言，并对培训机构进行认证。成员委员会可根据其特定的语言需求调整大纲内容，并可修改参考文献以适配本地的出版物。
2. 提供给认证机构，以便依据本教学大纲的学习目标，编制当地语言的考试题目。
3. 提供给培训机构，以便制作课件并确定合适的教学方法。
4. 提供给认证考生，以便其准备认证考试（可作为培训课程的一部分，也可自主备考）。
5. 提供给国际软件与系统工程界，以推进软件和系统测试行业的发展，并作为书籍和文章创作的基础。

## 0.2 利用生成式 AI 进行软件测试

生成式 AI 测试认证，面向所有运用生成式 AI（GenAI）进行软件测试的相关人员，涵盖测试员、测试分析师、测试自动化工程师、测试经理、用户验收测试人员以及软件开发人员等岗位。此外，对于希望初步了解如何将生成式 AI 应用于软件测试的人员，如项目经理、质量经理、软件开发经理、业务分析师、IT 总监和管理顾问，该认证同样适用。

## 0.3 测试人员职业发展路径

ISTQB®认证体系为测试专业人员在其职业生涯的各个阶段提供支持。获得 ISTQB® 生成式 AI 测试认证的个人也可能对核心域高级（测试分析师、技术测试分析师、测试管理和测试自动化工程师）以及之后的专家级（测试管理或改进测试过程）感兴趣。请访问 www.istqb.org 获取 ISTQB 认证测试体系的最新信息。

## 0.4 商业价值

本节列出了对获得生成式 AI 测试认证的考生有望带来的商业价值。

获得生成式 AI 测试认证的测试人员可以：

| | |
|---|---|
| GenAI-BO1 | 理解生成式 AI 的基本概念、能力及局限。 |
| GenAI-BO2 | 切实掌握针对软件测试场景，向大语言模型输入有效提示的实用技能。 |
| GenAI-BO3 | 深入了解在软件测试中使用生成式 AI 所面临的风险，以及相应的应对策略。 |
| GenAI-BO4 | 深入了解生成式 AI 解决方案在软件测试领域的具体应用。 |
| GenAI-BO5 | 切实助力组织内部软件测试生成式 AI 战略及路线图的制定与实施。 |

## 0.5 考核的学习目标、实践目标及知识认知水平

学习目标和实践目标有助于达成商业成果，用于编制生成式 AI 测试的认证考试题目。

通常，除引言、实践目标及附录外，本大纲所有内容均在考核范围内，考核认知水平分为 K1、K2 和 K3 三个级别。考试题目将考查 K1 级别的关键词知识（见下文）或各 K 级别的对应学习目标。

每章开头会给出具体的学习目标级别，并分类如下：

- K1:牢记
- K2:理解
- K3:应用

学习目标的更多详细信息及示例见附录 A。

章节标题正下方列出的所有关键词均需牢记，即便学习目标中未明确提及

每章开头都会列出具体的实践目标(HO)。为了能够通过实践来强化学习效果，每个实践目标都与一个 K2 或 K3 级别的学习目标(LO)相关联，HO 级别分类见下：

- H0：对练习进行现场演示，或者播放视频录像。由于并非由学员自己操作，因此严格定义上不算练习。
- H1：引导式练习。学员跟随讲师执行一系列操作步骤。
- H2：附带提示的练习。为学员布置练习及相关提示，以便在给定时间内完成练习。

## 0.6 生成式 AI 测试的认证考试

生成式 AI 测试认证考试将以本大纲为依据。回答考试题目可能需要运用本大纲中多个章节的内容。除引言、实践目标及附录外，大纲中所有章节均为考试范围。引用的标准、书籍和文章仅作参考，其内容不作为考试要求，以大纲本身的总结为准。

详情请参阅《CT-GenAI-培训评估认证指南 v1.0》文档。

报考说明：考生须先取得 ISTQB®基础级证书，方可参加 ISTQB®生成式 AI 测试认证考试。

## 0.7 认证

ISTQB®成员委员会可对培训材料遵循本教学大纲的培训机构进行认证。培训机构应从负责认证的成员委员会或机构获取认证指南。经认证的课程被视为符合本大纲，并允许将 ISTQB®考试作为课程的一部分。

本大纲的认证指南在《CT-GenAI-培训评估认证指南 v1.0》文档中予以明确。

## 0.8 标准的使用

存在一些与质量特性及软件测试相关的标准，例如基础级大纲中引用的 IEEE 和 ISO 等标准。引用这些标准旨在提供一个框架，或者在读者有需要时提供额外信息来源。请注意，本大纲仅将标准文档作为参考资料，标准文档内容不列入考试范围。如需了解更多关于标准的信息，请参阅第 6 章。

## 0.9 详细程度

本大纲在细节设定上，确保了国际范围内相关课程与考试的一致性。为达成这一目标，教学大纲涵盖以下内容：

- 以通用教学目标描述 ISTQB®生成式 AI 测试认证的总体意图。
- 明确学生必须能够回忆的术语。
- 针对每个知识领域制定学习目标，说明需要达成的认知学习成果。
- 对关键概念展开描述，并引用公认文献或标准等参考资料来源。
- 针对每个实践操作目标，给出推荐的实践内容，以辅助学习。

本大纲的内容，并未涵盖生成式 AI 测试的全部知识领域，它仅体现了 ISTQB®生成式 AI 测试认证培训课程需要覆盖的详细程度。大纲聚焦于在使用生成式 AI 进行测试时，可适用于所有软件项目的测试概念与技术。

本教学大纲采用 ISTQB®术语表中定义的软件测试和质量保证相关术语的表述（即名称与含义）。

## 0.10 本大纲的结构安排

考试范围总共有五个章节。每章的大标题注明了该章的建议学时；章节中的内容未再单独设定时间。对于经认证的培训课程，本大纲要求授课时长至少为 13.6 小时，各章节分配如下：

- 第 1 章：生成式 AI 在软件测试中的应用简介（100 分钟）
  - 测试人员将学习大语言模型的基础知识，包括分词处理和多模态能力。
  - 测试人员将探索生成式 AI（GenAI）在软件测试中的应用，区分 AI 聊天机器人与基于大语言模型的测试工具，并进行分词、上下文窗口及多模态提示的实验。
- 第 2 章： 面向高效软件测试场景的提示词工程（365 分钟）
  - 测试人员将学习如何为软件测试中的生成式 AI 构建有效且结构化的提示。
  - 测试人员将通过实践，获取用于软件测试任务的提示词工程技术经验并加以应用。
- 第 3 章： 生成式 AI 在软件测试中的风险管理（160 分钟）
  - 测试人员将学习如何识别并降低在使用生成式 AI 进行测试时出现的幻觉、推理错误和偏见。
  - 测试人员将学习如何处理软件测试中生成式 AI 涉及的数据隐私和信息安全问题。
  - 测试人员将了解软件测试中生成式 AI 的能源消耗和环境影响。
  - 测试人员将学习人工智能相关法规、标准以及如何在软件测试中合乎伦理道德、透明且安全地使用生成式 AI 的最佳实践。
- 第 4 章： 基于大语言模型（LLM）驱动的软件测试基础架构（110 分钟）

- 测试人员将探索检索增强生成等生成式 AI 架构以及生成式 AI 智能体。
- 测试人员将学习针对软件测试任务对大语言模型进行微调的过程。
- 测试人员将学习大语言模型运维（LLMOps）概念，以便在软件测试中部署和管理大语言模型。

- 第 5 章：在测试组织开展生成式 AI 的部署与集成工作（80 分钟）
  - 测试人员将学习一套用于将生成式 AI 集成到测试过程中的结构化路线图。
  - 测试人员将学习为实现生成式 AI 在测试过程中的集成，组织需做出哪些转型。

# 1. 生成式 AI 在软件测试中的应用简介 - 100 分钟

**关键词**

无

**生成式 AI 专用关键词**

AI 聊天机器人（AI chatbot），上下文窗口（context window），深度学习（deep learning），嵌入（embedding），特征（feature），基础大语言模型（foundation LLM），生成式 AI（generative AI），生成式预训练转换器（generative pre-trained transformer），指令微调大语言模型（instruction-tuned LLM），大语言模型（large language model），机器学习（machine learning），多模态模型（multimodal model），大语言推理模型（reasoning LLM），符号 AI（symbolic AI），词元化（tokenization），模型/转换器（transformer /transformer）

**第一章的学习目标：**

### 1.1 生成式 AI 基础与关键概念

| | | |
|---|---|---|
| GenAI-1.1.1 | (K1) | 回顾机器学习的不同类型，包括传统机器学习、深度学习以及生成式。 |
| GenAI-1.1.2 | (K2) | 阐述生成式 AI（GenAI）与大语言模型（LLM）的基础原理。 |
| HO-1.1.2 | (H1) | 在使用大语言模型（LLM）开展软件测试任务时，练习实施词元化(tokenization)以及 词元（token）数量评估。 |
| GenAI-1.1.3 | (K2) | 准确区分基础大语言模型、指令微调大语言模型以及大语言推理模型（LLM）。 |
| GenAI-1.1.4 | (K2) | 归纳多模态大语言模型（LLM）与视觉 - 语言模型的基本原理。 |
| HO-1.1.4 | (H1) | 针对软件测试任务，编写并执行适用于多模态大语言模型（LLM）的提示词，同时运用文本与图像作为输入内容。 |

### 1.2 软件测试中生成式 AI 的应用：核心准则

| | | |
|---|---|---|
| GenAI-1.2.1 | (K2) | 列举将大语言模型（LLM）用于测试任务时所应展现的关键能力的示例 |
| GenAI-1.2.2 | (K2) | 对比在软件测试工作中采用生成式 AI（GenAI）时的交互模式 |

# 1.1 生成式 AI 基础与关键概念

生成式 AI 是人工智能的一个分支，它运用大规模的预训练模型，生成诸如文本、图像或代码等类人化输出，大语言模型属于生成式 AI 模型，经海量文本数据集预训练，能够识别上下文信息，并根据用户的提示给出相关的回复。

关键概念包括：词元化（tokenization 即将文本分解为便于高效处理的单元）、上下文窗口（限制单次考量的信息量，以确保相关性）以及多模态模型（可处理文本、图像、音频等多种数据类型，实现丰富的交互）。

在软件测试中，这些大语言模型能够在整个测试过程中，为诸多任务提供支持，比如评审与完善验收准则、生成测试用例或测试脚本、识别潜在缺陷、分析缺陷规律、生成合成测试数据以及辅助生成文档等。

## 1.1.1 AI 谱系：符号 AI、经典机器学习、深度学习与生成式 AI

人工智能（AI）是一个广泛领域，涵盖了不同类型的技术，每种技术都有其独特的解决问题的方式，例如符号 AI、经典机器学习、深度学习和生成式 AI 等（本大纲范围之外的其他技术在此不作讨论）：

- 符号 AI 能借助基于规则的系统来模拟人类的决策过程。本质上，它运用符号与逻辑规则来表征知识。
- 经典机器学习是一种数据驱动的方法，需要进行数据准备、选择特征和模型训练环节，可用于缺陷分类、软件问题预测等任务。
- 深度学习运用一种名为神经网络的机器学习架构，能够自动从数据中学习特征。深度学习模型无需用户手动定义特征，就能在极为庞大且复杂的数据集（如图像、视频、音频或文本）中找出模式。但在实践中，像数据标注、模型调校和结果确认等工作，可能还是需要人工参与。
- 生成式 AI 运用深度学习技术，通过学习和模仿训练数据中的模式，创造新的内容（如文本、图像、代码）。如大语言模型这类模型能够在其训练范围内生成文本、编写代码，还能模拟推理或问题解决的过程。

总之，人工智能领域已在多个方向上发展演进，每个方向都有其独特优势与局限。在软件测试中应用生成式 AI 的关键优势在于，其采用的预训练模型能够直接用于测试任务，无需额外的训练环节。不过，这确实也伴随着一些风险（参见第 3.1 节）。

### 1.1.2 生成式 AI 与大语言模型基础

大语言模型基于生成式预训练的 Transformer 深度学习模型，通过海量数据集（涵盖书籍、文章及各类网站内容等）进行训练。相较于大语言模型，小语言模型（SLM）属于紧凑型模型，参数较少，旨在提供轻量级且专注特定领域的生成式 AI 解决方案。

大语言模型既能处理语言中的细微差别，也能生成连贯的内容。“词元化”和“嵌入”是助力大语言模型处理和生成内容的两个关键概念。他们将语言转换为模型可高效处理的数值形式。

- 语言模型中的“词元化(tokenization)”是指将文本拆解成名为 “词元”（token）的更小单位，词元小至单个字符，大到子词或单词。当大语言模型处理句子时，首先会对输入进行“词元化”，这样每个词元能被单独理解，同时整体语境得以保留。

- “嵌入”是词元的数值表示形式，它以一种适合生成式 AI 模型处理的格式，对词元的语义、句法以及上下文关系进行编码。每个词元被转换为高维空间中的一个向量，以此捕捉关于其含义和用法的微妙信息，含义相近或在上下文中作用相似的词元，它们的 “嵌入” 向量在这个空间中的位置会彼此靠近。这种位置上的接近，让大语言模型能够理解词汇间的关系，保留上下文信息，并生成连贯且贴合语境的回应。

大语言模型采用一种名为“Transformer 模型”（也叫“转换器”）的神经网络架构。Transformer 模型在处理长文本序列上下文以及学习词元间相互关系方面表现卓越，因此在语言任务中优势显著。在推理过程中，大语言模型会依据这些习得的关系，预测序列中的下一个词元，从而生成连贯且贴合上下文的文本。Transformer 模型能够基于训练数据和输入提示词，生成在统计上合理的新文本，但看似合理并不一定就正确。

大语言模型之所以呈现非确定性行为，主要源于推理机制和超参数设置的概率特性，这种内在的随机性会导致即使多次提供相同的输入，输出结果仍可能有所不同。

在大语言模型领域，上下文窗口指的是模型在生成回复时能够参考的前文文本量（以词元/token 为单位衡量），例如分析海量测试日志的场景时，更大的上下文窗口能让模型在处理较长文本段落时

保持内容的连贯性；然而，增大上下文窗口中的词元数量，会提升模型有效运行所需的计算复杂度，进而延长处理时间。

**实践目标 HO-1.1.2 (H1)：练习实施词元化**(tokenization)**以及 词元**（token）**数量评估**

这项实践性活动旨在帮助学员深入理解“词元化”及其在使用大语言模型时所带来的影响。该练习分为两个关键部分：

- **词元化**(tokenization)：使用分词器将一段示例文本拆分为若干个独立的**词元**。检查输出结果，观察单词、标点符号和短语的呈现方式，并识别其中的分词模式或细微差别。
- **词元**（token）数量评估：统计不同输入文本所生成的**词元**/token 数量。特别是结合模型上下文窗口限制以及效率方面的因素来进行分析。

完成练习后，学员将能更准确地预估不同文本结构与输入长度，会如何影响与大语言模型的交互。

## 1.1.3 基础、指令微调以及推理大语言模型

大语言模型通过逐步专业化的训练阶段来开发，从而提升其在各种任务上的表现。这些训练阶段也衍生出三大主要类别：基础大语言模型、指令微调大语言模型以及推理大语言模型。

- 基础大语言模型：这些是通用型模型，基于包含文本、代码、图像及其他形式的海量多样数据集进行训练。广泛的预训练使它们能够支持诸如自然语言处理、计算机视觉、语音识别等不同领域的各类任务。虽然基础模型功能强大且灵活，但一般来说，仍需进一步调整以满足特定任务需求。
- 指令微调大语言模型：由基础模型衍生而来，利用将提示与预期响应配对的数据集进行微调。该阶段能提升模型与人类指令的一致性，增强其在实际应用场景中的易用性。微调过程着重于优化任务契合度、对指令遵循程度和响应连贯性进行优化，进而有效提升模型对用户意图的理解与执行能力。
- 大语言推理模型：是在指令微调大语言模型基础上的拓展，它着重强化结构化的认知能力，例如逻辑推理、多步骤问题解决以及思维链推理等能力。这类模型会在精心筛选的任务集上进一步训练或微调，此类任务要求模型具备上下文理解能力、执行中间推理步骤以及处理综合复杂信息的能力。因此，他们更适用于处理高认知负荷的任务，包括各技术领域的相关任务。

在面向软件测试的生成式 AI 场景中，会同时用到指令微调大语言模型（有时也称为非推理型）和大语言推理模型，具体选用哪种模型，取决于当前特定测试任务的复杂度与推理需求。

### 1.1.4 多模态大语言模型与视觉-语言模型

多模态大语言模型拓展了传统的 Transformer 模型，使其能够处理文本、图像、音频、视频等多种数据模态。这类模型在海量且多元的数据集上进行训练，从而学习不同类型数据之间的关系。为处理各类模态，会针对每种数据类型调整词元化方式，例如，在 Transformer 模型处理图像前，会先利用视觉 - 语言模型将图像转换为嵌入向量。

视觉 - 语言模型作为多模态大语言模型的一个子集，专门融合视觉与文本信息，可完成图像描述、视觉问答以及分析文本和视觉输入间一致性等任务。

在软件测试领域，多模态大语言模型（尤其是结合了视觉 - 语言的大语言模型）带来了重要应用价值。它们能够同时分析应用程序的视觉元素（如屏幕截图和 GUI 图形用户界面）以及相关的文本描述，比如缺陷报告、用户故事。这一能力使测试人员能够识别出预期结果与屏幕截图中的实际视觉元素之间的差异。此外，结合了视觉 - 语言的大语言模型还能结合文本数据和视觉线索，生成内容丰富且逼真的测试用例，从而提高整体的测试覆盖率。

**实践目标 HO-1.1.4 (H1)： 根据给定的提示词，使用多模态大语言模型完成一项测试任务的评审与执行工作**

本次练习包含两个步骤，你需要评审并执行一个给定的提示词，利用文本和图像输入，驱动多模态大语言模型来完成测试任务：

- 评审输入：评审提示词以及输入数据（文本和图像）。
- 执行提示词并验证结果：使用多模态大语言模型同时输入图像与文本，检查模型的反馈结果。

此练习展示了如何利用多模态大语言模型来完成既包含文本又包含图像输入的软件测试应用场景，同时涵盖了其带来的优势及潜在挑战。

## 1.2 软件测试中生成式 AI 的应用：核心准则

生成式 AI 在各类测试活动中具备变革性的能力。大语言模型擅长处理自然语言和代码，能够生成连贯的文本和代码、回答问题、归纳信息、进行语言翻译，以及在多模态环境下分析图像。

不同岗位的测试专业人员可通过两种相辅相成的方式利用生成式AI：其一，借助生成式AI聊天机器人，它能针对各类询问即刻给出回应；其二，运用整合到测试工具中的，由大语言模型驱动的应用程序。

### 1.2.1 大语言模型用于测试任务时所应展现的关键能力

大语言模型能够解读需求、规格说明、截图、代码、测试用例及缺陷报告，这使得它们能够成为在整个测试过程中理解并梳理所需信息，并生成测试件要素的有力工具。以下是与软件测试相关的部分大语言模型的能力：

- 需求分析与改进：大语言模型能够帮助分析需求以及测试依据中的其他要素，通过识别其中模糊不清、前后矛盾或信息的缺失之处，还能提出有价值的问题，以便在与利益相关方沟通时，进一步明确需求。
- 支持测试用例生成：大语言模型能够基于系统需求、用户故事或任何其他测试依据为要素，辅助生成测试用例，并给出相应的测试目标建议。
- 测试结果参照物生成：大语言模型够辅助生成预期结果。
- 测试数据生成：大语言模型能够生成数据集、设定边界值，并创建不同组合的测试数据。
- 测试自动化支持：大语言模型能够根据测试用例描述生成测试脚本，并通过提出修改建议和识别合适的测试设计技术，来优化现有的测试脚本。
- 测试结果分析：大语言模型能够生成摘要并根据严重程度和优先级对异常情况进行分类，来帮助分析测试结果。
- 测试件创建：大语言模型能够辅助生成各类文档，例如测试计划、测试报告和缺陷报告，并随着项目的进展更新这些文档。

上述这些能力展示了大语言模型如何在整个测试过程中，对软件测试的各个方面产生影响。

### 1.2.2 AI 聊天机器人和大语言模型驱动的测试应用程序

AI 聊天机器人与大语言模型驱动的测试应用程序，都能协助测试人员开展工作，不过它们在功能、灵活性以及集成方式上存在差异。

AI 聊天机器人提供了一个用户友好的对话式界面，使测试人员能够直接与大语言模型进行交流。这种自然语言交互方式，允许测试人员输入问题、指令或提示，并立即获得贴合上下文的回应，通过诸如提示词链接等技术，测试人员可以反复优化输出结果，这使得 AI 聊天机器人特别适用于常规任务、探索性测试，甚至通过为新测试人员提供快速获取测试知识和实践的途径来进行入职培训。

在需要快速反馈、阐述测试概念，或动态探索需求及潜在测试用例的场景中，这些 AI 聊天机器人尤为实用。其界面直观，即便是非技术背景的相关人员也能轻松上手，这扩大了潜在用户群体，促使更多人使用。

相比之下，由大语言模型驱动的测试应用程序，是通过 API 整合大语言模型的能力，来执行定义清晰且往往可自动化的测试任务。这类应用程序具备更高的定制化能力与可扩展性，使企业和工具供应商能够将生成式 AI 集成到现有的测试框架中。这就实现了重复性或复杂任务的自动化处理，例如测试用例生成、缺陷分析、测试数据合成等。在更先进的实施方案里，企业可以创建专门用于承担特定测试职责的 AI 智能体（详见第 4 章）。

无论测试人员是通过 AI 聊天机器人，还是通过集成了大语言模型驱动的测试应用程序与大语言模型进行交互，要在测试中成功应用生成式 AI，都离不开强大的提示词工程（详见第 2 章）。精心设计的提示词、清晰明确的指令至关重要，只有这样才能确保大语言模型生成的输出准确、贴合测试场景，且与测试目标保持一致。这种方法有助于充分发挥生成式 AI 的价值，并为各类测试活动提供持续、可靠的支持。

# 2. 面向高效软件测试场景的提示词工程 - 365 分钟

**关键词**

验收准则（acceptance criteria），测试脚本（test script），测试用例（test case），测试条件（test condition），测试数据（test data），测试设计（test design），测试报告（test report）

**生成式 AI 专用关键词**

少样本提示（few-shot prompting），元提示（meta prompting），自然语言处理（natural language processing），单样本提示（one-shot prompting），提示词（prompt），提示词链（prompt chaining），提示词工程（prompt engineering），系统提示词（system prompt），用户提示词（user prompt），零样本提示（zero-shot prompting）

**第二章 学习目标：**

**2.1 高效提示词开发**

| | | |
|---|---|---|
| GenAI-2.1.1 | (K2) | 列举在软件测试生成式 AI 应用中所采用的各类提示词结构示例。 |
| HO-2.1.1 | (H0) | 观察针对软件测试任务给出的多个提示词，逐一识别其中的角色、上下文、指令、输入数据、约束条件及输出格式等要素。 |
| GenAI-2.1.2 | (K2) | 区分用于软件测试的核心提示技术。 |
| HO-2.1.2a | (H0) | 观察提示词链、少样本提示以及元提示在软件测试任务中的应用演示。 |
| HO-2.1.2b | (H1) | 判定在给定示例中正在运用的提示词工程技术。 |
| GenAI-2.1.3 | (K2) | 区分系统提示词和用户提示词。 |

**2.2 将提示词工程技术应用于软件测试任务**

| | | |
|---|---|---|
| GenAI-2.2.1 | (K3) | 将生成式 AI 应用于测试分析任务中。 |
| HO-2.2.1a | (H2) | 练习通过多模态提示，依据图形用户界面（GUI）线框图，为用户故事生成验收准则。 |
| HO-2.2.1b | (H2) | 通过练习提示词链与人工验证的方法，循序渐进地分析给定用户故事，并对验收准则进行细化。 |
| GenAI-2.2.2 | (K3) | 将生成式 AI 应用到测试设计与测试实施任务中。 |
| HO-2.2.2a | (H2) | 练习运用提示词链、结构化提示和元提示，借助人工智能，用户故事生成功能测试用例。 |
| HO-2.2.2b | (H2) | 运用少样本提示技术，依据用户故事生成 Gherkin 风格的测试场景。 |

| | | |
|---|---|---|
| HO-2.2.2c | (H2) | 基于给定的测试套件以及风险/依赖关系数据，运用提示词链来为测试用例进行优先级排序。 |
| GenAI-2.2.3 | (K3) | 将生成式 AI 应用于自动化回归测试任务中。 |
| HO-2.2.3a | (H2) | 通过练习少样本提示法，来创建并管理关键字驱动的测试脚本。 |
| HO-2.2.3b | (H2) | 练习运用结构化提示，在回归测试的环境下开展测试报告分析工作。 |
| GenAI-2.2.4 | (K3) | 将生成式 AI 应用于测试控制和监测任务中。 |
| HO-2.2.4 | (HO) | 查看 AI 依据原始数据整理而形成的测试监测度量。 |
| GenAI-2.2.5 | (K3) | 基于给定的情境和测试任务，挑选适合的提示技术并加以应用。 |
| HO-2.2.5 | (H1) | 基于给定的测试任务，挑选并采用契合任务上下文的提示技术。 |

### 2.3 评估生成式 AI 结果并优化软件测试任务提示词

| | | |
|---|---|---|
| GenAI-2.3.1 | (K2) | 理解用于评估生成式 AI 在测试任务结果上的度量。 |
| HO-2.3.1 | (HO) | 观察如何运用评估度量，对生成式 AI 在测试任务中的结果进行评估。 |
| GenAI-2.3.2 | (K2) | 给出评估和迭代优化提示词的方法示例。 |
| HO-2.3.2 | (H1) | 针对特定的测试任务，对提示词进行评估与优化。 |

# 2.1 高效提示词开发

有效的提示词设计,能确保生成式 AI 工具精准且高效地执行软件测试任务，让测试人员从大语言模型中获取有价值的结果。一个结构化的提示词包含不同的组成部分（见第 2.1.1 节）。这些部分各自发挥作用，提升提示词的清晰度和精准度，从而将需求和预期有效地传达给大语言模型。

多样化的提示词工程技术，能够增强提示词在软件测试中的效用。诸如提示词链、少样本提示和元提示等技术，有助于应对复杂的测试难题（见第 2.1.2 节）。

将结构化提示词（见第 2.1.1 节）与核心提示技术相结合，目的在于针对软件测试任务向大语言模型发起查询时，能够取得理想的结果（见第 2.1.3 节）。

## 2.1.1 结构化提示词在生成式 AI 软件测试中的应用

一个结构完善的软件测试提示词通常包含以下六个组成部分：

- 角色：角色用于明确生成式 AI 模型在生成响应时应扮演的视角或身份。指定角色有助于大语言模型明确其职责，并采用恰当的语气或方法，例如以测试人员、测试经理或测试自动化工程师的身份开展工作。
- 上下文：上下文提供了生成式 AI 模型确定测试条件所需的背景信息。包括测试对象的详细信息、待测试的具体功能以及任何相关的背景情况。
- 指令：指令是向生成式 AI 发出的指示，用于概述需要执行的具体任务。清晰、明确且简洁的指令应当包含任务描述以及该任务的各项相关要求。
- 输入数据：输入数据包括执行任务所需的各类信息，例如用户故事、验收准则、屏幕截图、代码、现有测试用例或输出示例。提供详细且结构化的输入数据有助于大语言模型生成更准确且具备上下文感知能力的结果。
- 约束：约束用于明确大语言模型应遵守的各项限制或特殊注意事项。约束条件有助于说明如何将指令内容应用于输入数据。
- 输出格式：输出规范用于指明响应的预期格式、结构或特征。这些规范有助于塑造大语言模型的最终输出形态。

上述各部分共同构成了提示词的基本结构。在实际应用中，这一结构需要与提示工程技术的实施（详见第 2.1.2 节）相结合，具体方案应依据待执行的任务类型和所使用的大语言模型特性进行相应调整。

**实践目标 HO-2.1.1 (HO)：观察和分析提示词组成部分**

在一次演示中，在 AI 聊天机器人上试验多个结构化提示词，每个提示词都针对特定的软件测试任务。这些提示词遵循由六个关键部分组成的结构化格式：角色、上下文、指令、输入数据、约束和输出格式。该演示旨在促进对这些结构化提示词的观察和分析，强调各组成部分如何为用于软件测试任务的大语言模型助力，提供准确、相关且可行的见解做出贡献。

## 2.1.2 软件测试的核心提示技术

近年来，针对不同的生成式 AI 应用场景，提出了许多大语言模型提示技术（Schulhoff 2024）。其中，三种核心提示技术通常与上述六部分提示词结构（见第 2.1.1 节）结合，用于生成式 AI 的测试任务：提示词链、少样本提示和元提示。

- 提示词链（Prompt chaining）是将单一任务分解为一系列中间步骤（多轮提示）的技术方法。每一步的输出结果需经人工或自动核验与优化后，方可进入下一流程。该方法通过前序响应为后续提示提供导向，可显著提升结果准确性。提示词链在测试过程中具有重要应用价值，尤其适用于任务复杂度高、需拆解为子任务并对大语言模型中间输出进行系统性校验的场景；同时，该技术支持测试过程中的动态交互机制。

- 少样本提示（Few-shot prompting）是指在提示内容中向大语言模型供示例的技术方法。零样本提示（无示例）依赖模型的固有知识生成响应；而单样本提示（One-shot prompting）则通过提供一个示例，以明确特定输入对应的预期输出。少样本提示包含多个示例（即 “少量示例”），用以进一步规范模型的目标响应行为。

- 这项技术通过提供清晰的参考示例并确保结果连贯且符合预期，来引导模型。少样本提示在那些可以通过示例说明所需行为的任务中特别有效，它能让模型有效地进行归纳，进而得出可靠的结果。

- 元提示（Meta prompting）是指利用人工智能自主生成或优化自身提示的技术方法。在迭代循环中，大语言模型可生成提示，供测试人员进行评估与优化。该方法通过借助 大语言模型 对

优化提示的相关知识，实现提示质量的提升。当效率与提示优化至关重要时，元提示技术尤为适用，因其可减少设计有效提示所需的人工成本。元提示的另一优势在于，若测试人员不确定如何构建有效的提示，可与大语言模型协作共创。这体现了一种与生成式 AI 工具的协同工作模式，即测试人员与 AI 通过交互式协作达成共同目标。这一协同理念揭示了与 AI 工具协作的新范式，不仅能提升提示词工程的效率与学习效果，也适用于结对编程与结对测试等场景。

这些提示技术可以有效地结合使用，以提升大语言模型的输出结果（见第 2.2.5 节）。

**实践目标 HO-2.1.2a (HO)：在软件测试任务中观察和讨论提示词链、少样本提示和元提示**

参与者将在 AI 聊天机器人上体验提示词链、少样本提示和元提示，并分别将其用于特定的软件测试任务。本演示旨在结合软件测试场景，探索和研讨上述提示技术，重点阐明各技术对提升大语言模型输出准确性与完整性中的作用机制。

**实践目标 HO-2.1.2b (H1)：在给定示例中识别提示词工程技术**

参与者将阅读一组与软件测试相关的提示示例，以识别其中所应用的核心提示技术。本次活动的重点在于辨识提示词链、少样本提示及元提示等技术，并着重阐明其独特特征与实际应用场景。

本活动旨在深化参与者对不同提示技术如何助力生成式 AI 在软件测试领域实现高效应用的理解。

## 2.1.3 系统提示词与用户提示词

系统提示词和用户提示词在与大语言模型的交互中服务于不同的目的，在引导对话方面各自发挥着独特作用。系统提示词通常由开发人员或测试人员设定，用于指引大语言模型的整体行为，在大多数界面中，使用聊天机器人的用户无法查看或修改该提示词。

系统提示词相当于一组预定义的指令，用于界定大语言模型的行为模式、特性和运行参数。这些运行参数决定了大语言模型的响应方式。例如，采用正式口吻、保持回答简洁明了、遵守特定领域规则或避免某些行为。系统提示词为整个对话奠定了规则基础。它可能涵盖结构化提示词的部分要素，例如角色、上下文及约束条件等。

系统提示词在整个交互会话中保持不变，为大语言模型的响应方式奠定了基本框架。例如，一条系统提示词可能这样表述：“你是一名专业的软件测试助手。始终以清晰、正式的语言回应，聚焦与

国际软件测试工程师认证（ISTQB）标准一致的实践方法。避免主观推测，相关内容需引用测试原则作为依据。”

另一方面，用户提示词代表用户在聊天机器人中的实际输入或问题。内容可包含用户希望大语言模型处理的具体指令、问题或任务。与系统提示词不同，用户提示词是直接可见的，并构成每次响应的直接上下文。

例如，一条用户提示词可能是：“列举黑盒测试与白盒测试的主要区别，并举例说明。”

典型用法是在对话开始时设置一次系统提示词，随后在每次交互中发送连续的用户提示词。大语言模型通过综合考虑固定不变的系统提示词与当前的用户提示词，生成相应响应。为了实现有效应用，系统提示词应清晰地明确大语言模型的角色和可能的存在的约束条件。它也可能包含上下文和一般指令，例如关于预期输出的说明。

用户提示词必须重点突出且结构合理，应包括明确的指令、额外的相关上下文以及输出格式说明。

## 2.2 将提示词工程技术应用于软件测试任务

将提示词工程技术应用于软件测试，使生成式 AI 能够支持测试分析、测试设计、测试自动化、测试用例优先级设置、缺陷检测、覆盖率分析以及测试监测与控制等任务。通过运用并结合提示词链、少样本提示和元提示等技术，团队可针对具体测试目标定制 AI 提示词，从而使输出结果更精准、更相关且更有效。高质量的输入对获得有意义的 AI 结果至关重要。

### 2.2.1 基于生成式 AI 的测试分析

生成式 AI 可通过生成并确定测试条件的优先级、识别测试依据缺陷以及提供覆盖率分析来支持测试分析任务。输入数据包括需求、用户故事、技术规格、图形用户界面（GUI）线框图及其他相关信息。输出由典型的测试分析工作产品组成，例如按照优先级排序的测试条件（例如验收准则）。

以下是一些可由生成式 AI 支持的典型测试分析任务：

- **识别测试依据中的潜在缺陷：**生成式人工智能可协助分析测试依据中可能导致缺陷的不一致性、模糊性或信息缺失。通过对比相似需求模式或运用历史缺陷报告的知识，大语言模型能标记出潜在异常并提出改进建议。

- **基于测试依据生成测试条件，**例如基于需求/用户故事：大语言模型可分析需求和用户故事，生成测试条件。通过自然语言处理技术，它们能够解读需求含义并将其分解为可量化、可测试的表述，从而将需求转化为具体测试条件。

- **基于风险等级优先级对测试条件排序：**依据各测试条件的风险发生概率与失效影响程度，大语言模型可协助对测试工作进行优先级排序。通过考量合规要求、用户可见特征（例如登录功能或支付处理）及历史缺陷数据，大语言模型能提出优先级建议。

- **支持覆盖率分析：**通过将需求和用户故事映射到测试条件，大语言模型可执行覆盖率分析，以确定测试依据是否覆盖所有方面。这对于需求复杂的项目尤为有用，因为覆盖率缺口可能导致遗漏缺陷。

- **建议测试技术：**生成式 AI 可根据待测需求或用户故事的类型，推荐相应的测试技术（例如边界值分析、等价类划分）。这有助于测试人员针对特定测试条件应用最有效的测试方法。

向大语言模型提供的输入内容质量及相关性，会直接影响模型生成的输出结果的准确性和精确性。

**实践目标 2.2.1a (H2)：练习创建结构化多模态提示词，以依据图形用户界面（GUI）线框图，为用户故事生成验收准则**

这是一项练习，旨在通过多模态输入（文本与图像）来实践结构化提示词的编写。目标是从用户故事和 GUI 线框图中生成高质量（即结构合理、清晰完整）的验收准则。可添加其他文本元素提供背景信息，例如输入字段的约束条件或数据处理需遵循的业务规则。

通过对比大语言模型生成的结果，评估不同设计方案的结构化提示词（角色设定、上下文描述、操作指令、文本/图像输入数据、约束条件及输出格式）对测试分析任务的影响。

本练习旨在提供实践经验，让参与者认识到提示词结构构建的重要性、精确指令的贡献价值，以及文本和图像上下文数据在从大语言模型中获取准确相关结果时所发挥的关键作用。

**实践目标 2.2.1b(H2)：通过练习提示词链与人工验证的方法，循序渐进地分析给定用户故事，并对验收准则进行细化**

这是一项练习，旨在通过实践提示词链分析给定用户故事并完善验收准则：首先识别模糊点，其次评估可测试性，最后评估完整性。本练习倡导循序渐进的方法，在每个步骤中不断优化分析，确保

验收准则结构合理且切实可行，从而实现测试目标。在每个步骤中，由大语言模型生成的结果需人工核验，必要时通过调整输出内容或与大语言模型进行提示词链交互进行修正。如此，后续阶段便能基于前阶段的准确结果，聚焦于完善验收准则的其他方面。

这项练习提供了将复杂任务分解为子任务，并通过人工验证每个阶段结果来体会其好处的实际经验。

### 2.2.2 将生成式 AI 应用到测试设计与测试实施任务

如[ISTQB_CTFL_SYL]所述，测试设计涉及测试条件的详细制定与优化，这些条件随后被转化为测试用例及其他测试件。测试实施则包含创建或获取执行测试所需的必要测试件。

在生成式 AI 的支持下，手动测试与自动化测试脚本均可在测试执行计划中完成创建、优先级排序及安排。生成式 AI 能通过协助创建和评估各类测试件，包括测试用例、测试数据、测试脚本及测试环境，为这一庞大的测试活动提供显著支持。

以下是生成式 AI 可支持的典型测试设计与实施任务：

- **测试用例生成：**通过自然语言处理技术，生成式 AI 能基于功能性与非功能性需求创建测试用例草稿。当接收到合适的提示时，大语言模型可自动建议测试先决条件、输入值、预期结果及覆盖率标准，从而生成满足不同测试目标的测试用例——从基础功能验证到复杂端到端测试均可覆盖

- **测试数据合成：**生成式 AI 能创建具有代表性且保护数据隐私的合成测试数据，类似生产数据，同时覆盖极端情况和多样化测试条件。此类合成数据可用于功能性与非功能性测试，能根据应用需求定制，在模拟真实场景时可防止泄露敏感信息。

- **自动化测试脚本生成：**生成式 AI 能从结构化测试用例中生成手动测试规程和自动化测试脚本，解析测试步骤并将其转化为兼容各类测试自动化框架的代码。这些测试脚本可根据新需求进行更新或扩展

- **测试执行调度与优先级设置：**生成式 AI 可以分析测试用例及其相互依赖关系，基于诸如优先级、关联风险、资源可用性及测试目标等因素优化测试执行计划。

**实践目标 2.2.2a (H2):练习运用提示词链、结构化提示和元提示，借助人工智能，从用户故事生成功能测试用例**

本练习聚焦于运用生成式 AI 从用户故事中开发功能测试用例，借助提示词

链、结构化提示及元提示技术确保覆盖全面。第一步是创建提示词，指导 AI 根据给定的验收准则按特定输出格式生成功能测试用例。第二步是验证生成的测试用例完整性。在此处，通过提示 AI 生成一份覆盖率汇总表，以确保每个验收准则均能被涵盖。最后，第三步创建元提示以辅助生成端到端测试规程。该元提示有助于优化生成全面端到端测试的指令，通过迭代改进，实现效果的最大化。

该实践深化了对运用大语言模型进行测试用例生成、覆盖率确认及端到端测试的理解。

**实践目标 2.2.2b (H2)：运用少样本提示技术，依据用户故事生成 Gherkin 风格的测试用例**

本练习旨在运用少样本提示生成技术，根据给定用户故事生成 Gherkin 风格测试用例。首先回顾预定义示例及 Gherkin 语法，步骤 1 需选取 n 个包含提示词的实例，每个实例包含用户故事、测试条件及预期“假定-当-那么”风格测试用例以建模目标输出。随后将该提示词应用于新的用户故事，生成反映原始测试条件的 Gherkin 场景。若结果不准确，需优化提示词或实例。

本练习有助于积累将少样本提示技术应用于实际测试设计与实施任务的经验。

**实践目标 2.2.2c (H2)：基于给定的测试套件以及以及其中测试用例的风险 / 依赖关系数据，运用提示词链来为测试用例进行优先级排序**

本练习聚焦于运用生成式 AI 优化特定测试套件中的测试用例优先级设置，同时综合考虑风险分析及测试用例间的依赖关系。课程首先简要概述风险驱动、覆盖率驱动和需求驱动等不同测试方法，并审视给定的测试套件。随后参与者将通过创建提示词，为各类测试优先级设置策略生成可操作的优先级设置方案。基于提示词和给定输入数据生成的大语言模型结果需人工验证，以检测其推理过程中的错误。

本练习旨在探索生成式 AI 在需要多维度推理能力的测试任务中的应用（此处指测试用例优先级设置需考虑的各类风险与依赖关系）。

### 2.2.3 将生成式 AI 应用于自动化回归测试任务中

随着每次新迭代或版本的完成，需运行的回归测试用例数量往往会增加，这使其成为自动化的理想之选，特别是在测试执行频率高的持续集成 / 持续交付（CI/CD）流程中尤为适用。生成式 AI 可以通过辅助自动化回归测试套件的创建、维护和优化来简化该流程。通过动态适配代码库的变更并开展影响分析，生成式 AI 可以识别软件中哪些区域最有可能受近期改动的影响，从而将回归测试的力量集中在最需要的地方。

以下是一些可借助生成式 AI 提示词支持的典型自动化回归测试和测试报告活动：

- **采用关键词驱动自动化的方式开展自动化测试脚本实施工作：**大语言模型能够基于关键字驱动的测试自动化框架来实现测试脚本，其中预定义的关键字代表常见的测试步骤。生成式 AI 能够将这些关键词映射到特定的测试用例，生成测试脚本，为测试人员和测试自动化工程师的工作提供协助

- **影响分析和测试优化：**生成式 AI 可用于分析代码更改以识别高风险区域，从而在最需要的地方实现有针对性的回归测试。

- **自愈与自适应测试：**生成式 AI 可应用于自动调整测试脚本，以应对用户界面（UI）或应用程序编程接口（API）的细微变更，避免因微小修改而引发的不必要的失败，保障测试套件在较长时期内维持稳定状态。

- **自动化测试报告与洞察：**生成式 AI 能够生成详细且即时可用的测试报告，其中包含成功度量、失败情况及关键分析结果。其还能为利益相关方提供可视化面板，该面板不仅可以突出展示测试趋势，还能针对潜在故障点提供预测性见解。

- **增强的缺陷报告和根本原因分析：**生成式 AI 可以支持自动汇总生成包含测试日志、截图和测试环境数据的综合缺陷报告。

这些活动可以应用于各类回归测试，包括功能和非功能的回归测试。然而，测试员必须意识到，生成式 AI 可能会出错。因此，必须根据相关风险仔细检查生成的输出内容（参见第 3 章）。

此外，生成式 AI 可以协助进行端到端的基于图形用户界面（GUI）和 API 的自动化回归测试，每种测试都有其独特的挑战和解决方案。GUI 测试经常因用户界面的经常性更改而变得不稳定；生成式 AI 可以自动调整测试脚本以处理诸如动态定位符和修改后的交互等更改，减少对手动干预的需求。

API 回归测试面临请求/响应格式、端点和认证更改等挑战。生成式 AI 可以自动使测试脚本适应不断演进中的 API 规格说明，并生成多样化的测试数据，从而保持全面的覆盖并减少手动更新的需求。

**实践目标 2.2.3a (H2)：通过练习少样本提示法，来创建并管理关键字驱动的测试脚本**

本练习重点在于借助 GUI 测试自动化框架，为特定 Web 应用程序开发并实现测试脚本自动化。。练习主要分为两个部分:测试自动化和测试脚本调试。第一部分：指导如何为关键字库创建文档、生成初始测试脚本、利用 AI 对这些测试脚本进行验证，并通过添加额外的测试脚本扩大覆盖范围。第二部分：强调调试支持，即利用系统提示创建一个可以检查和纠正测试脚本的 AI 助手.

本练习将传统的测试自动化与 AI 辅助验证相结合,展示了如何利用少样本提示，完成关键字驱动测试脚本的创建、维护与调试工作。

**实践目标 2.2.3b (H2)：练习运用结构化提示，在回归测试的环境下开展测试报告分析工作**

本练习展示了一种借助结构化提示词，分析回归测试报告的系统化方法。过程始于分析提供的测试结果，并与测试规格说明进行比较。随后逐步对相似缺陷进行聚类、维护已知异常列表以及对发现的结果进行交叉核对。在单次大语言模型对话中，每一步都与下一步骤相关联。

这种循序渐进的方法演示了如何使用结构化提示词将回归测试结果和测试日志转化为可执行的见解，从而支持回归测试背景下的有效测试报告分析。

## 2.2.4 将生成式 AI 应用于测试监测和测试控制任务

测试监测任务需要检索大量（有时为非结构化）数据，这些数据往往已存在于测试管理工具中，生成式 AI 可助力对其进行分析与综合处理。

生成式 AI 有助于开展多项测试监测与测试控制任务，包括：

- **测试监测和度量分析：**生成式 AI 能够推动测试监测的自动化进程，同时对数据趋势进行分析，以此预测潜在风险，并在出现任何与计划偏差时向团队发出警报。这使团队能够随时掌握情况，进而采取行动，以维持质量标准。

- **测试控制：**生成式 AI 可辅助测试控制工作，通过提供专业见解，助力重新确定测试优先级、调整测试进度表，以及按需重新分配资源。如此一来，能够确保测试工作保持灵活性，并聚焦于高优先级领域。

- **测试完成情况分析与持续学习：**生成式人工智能能够生成详尽的测试完成报告，精准提炼成功经验与汲取的教训，为团队提供有力支持。借助这些信息，团队可以进一步优化测试策略，完善后续测试过程，实现持续改进。

- **强化的测试度量可视化与报告:**生成式 AI 能够辅助打造动态化的仪表盘，并生成自然语言形式的汇总报告。这确保了所有利益相关方都能便捷获取关键度量数据。这种支持为快速决策提供了坚实的数据基础，同时让测试进展清晰直观地呈现出来，便于各方及时掌握测试动态。

**实践目标 2.2.4 (HO)：查看 AI 依据原始数据整理而形成的测试监测度量**

本演示旨在说明生成式 AI 如何协助测试团队，将测试数据转化为可用于实际决策的测试监测度量，助力团队做出明智的决策。首先从测试工具中提取测试数据，然后由大语言模型对其进行处理，生成诸如测试进度、缺陷趋势或覆盖率等关键度量，并突出其中潜在风险。之后，这些由 AI 生成的度量会展示在仪表盘上，同时以自然语言形式进行总结，以便所有利益相关方能够轻松理解。

本演示清晰展示了生成式 AI 如何将测试数据转化为实用的分析结果，帮助测试团队有效监测测试进度、管理质量，并迅速适应各种变化。

## 2.2.5 为软件测试选择提示技术

下表呈现了第 2.1.2 节中提到的三种提示技术，基于测试任务特点的适用性情况。

| 提示技术 | 推荐应用场景 | 关键特性与应用 |
| --- | --- | --- |
| 提示词链/Prompt Chaining | 适用于需要高精度且每一步都需人工验证的复杂任务。 | 该技术将任务拆解为多个较小步骤，在测试分析、测试设计以及测试自动化场景中颇具效用，因为在此类场景下，每个测试步骤的准确性都需加以检查。 |

| | | |
|---|---|---|
| 少样本提示/Few-shot Prompting | 适用于重复性任务，或要求特定 / 受限输出格式的任务。 | 此方法通过向生成式 AI 提供示例，使其能够按照特定模式进行重复性生成。例如在 Gherkin 风格的测试用例（如基于场景的测试用例）、关键字驱动的测试，以及具有特定输出格式的测试报告生成中，都能发挥作用。 |
| 元提示/Meta Prompting | 适用于灵活、动态的任务，对为新任务构思提示词很有帮助。 | 通过对目标和要执行任务的一般性描述，来引导大语言模型生成提示词。在诸如测试报告分析和异常检测等各类复杂任务中都十分有用。 |

甚至对于单个用例，我们可以组合运用多种技术。例如，先通过元提示生成初始提示词。这个生成的提示词中可能包含一些示例，这些示例需要根据实际情况调整并进一步完善，此时就可以采用少样本提示技术。最后，将任务拆解为多个更小的子任务，以便对中间步骤进行确认，这一步则要用到提示词链技术。

**实践目标 2.2.5 (H1)：为给定测试任务选择贴合上下文的提示技术**

本练习着重于让参与者针对不同测试任务，挑选合适的提示技术。参与者会拿到几个具有不同挑战的测试任务。对于每个测试任务，参与者需要评估任务的特点，判断它是需要精确性，还是具有重复性结构，进而推荐最符合任务情境、满足任务特定需求的提示技术。之后，小组将对这些选择进行讨论。

本练习旨在深化参与者对不同提示技术如何在实际测试工作中有效应用的理解。

## 2.3 评估生成式 AI 结果并优化软件测试任务提示词

评估生成式 AI 在软件测试中的性能，需要一套清晰的度量来评判生成结果的质量、相关性及有效性（Li，2024）。这些度量，无论是通用的还是针对特定任务的，都有助于优化大语言模型的提示词。

### 2.3.1 评估生成式 AI 测试任务结果的度量

可采用以下度量来评估生成式 AI 在测试任务中的结果质量与效率：

| 度量 | 描述 | 示例 |
| --- | --- | --- |
| 精度/Accuracy | 对照专家编写的测试用例、需求规格说明及其他规范性文件，衡量生成式 AI 输出结果的整体正确性。 | 生成的测试用例对所有指定需求的覆盖程度。 |
| 准确率/Precision | 评估生成的输出在特定目标方面的正确程度。 | 生成的测试用例正确识别异常的程度。 |
| 召回率/Recall | 衡量模型识别数据集中所有相关实例的能力。 | 生成的测试用例对数据类中有效和无效等价分区的覆盖程度。 |
| 相关性与周境契合度/Relevance and Contextual Fit | 判定所生成的输出是否适用于特定周境。 | 所生成的测试用例与测试基准的相符程度，以及整合特定领域需求的程度。 |
| 多样性/Diversity | 确保覆盖广泛的输入和场景，避免重复。 | 生成的测试用例对各种用户行为的覆盖程度，以及对边缘情况的探究程度。 |
| 执行成功率/Execution Success Rate | 衡量生成的测试用例或测试脚本能够成功执行的比例。 | 确定在正常运转的测试环境下，所生成的测试脚本有多少可在无语法错误或输出格式问题的情况下得以执行。 |
| 时间效率/Time Efficiency | 评估相较于人工测试所节省的时间。 | 对比 AI 生成测试用例所需的时间与人工创建同等测试用例所需的时间。 |

除了这些通用度量指标外，还可以定制特定于任务的度量指标，以评估生成式 AI 对特定测试活动的支持程度。

为了有效地评估这些度量指标，测试人员可以进行人工评审，也可以将评估自动化，比如将大语言模型的输出与预定义的参考内容进行比较。鉴于生成式 AI 的不确定性，这些度量指标必须基于具有统计意义的数据。

**实践目标 2.3.1 (HO)：观察如何使用度量评估生成式 AI 在测试任务中的结果**

在针对一项特定测试任务的演示过程中，会展示适配该任务的、用于评估生成式 AI 成果的各项度量指标，以及这些度量指标如何具体应用于大语言模型，并在该测试任务中所取得的成果。

本次演示旨在说明，评估指标对于增强人们对生成式 AI 在软件测试领域成果的信心具有重要意义。

## 2.3.2 评估与迭代优化提示词的技巧

基于上述度量指标，可运用特定的提示词评估与优化技巧，来提升生成式 AI 的输出效果：

- **提示词的迭代式调整：**从一个基础提示词开始，依据观察到的结果对其进行迭代式调整。逐步增添更多上下文信息或调整表述方式（比如专业术语的使用），从而改进提示词的精准性与相关性。
- **提示词的 A/B 测试：**创建多个版本的提示词，依据预先设定的指标，评估哪个版本能得出更优的结果。该方法有助于确定哪种提示词的表述方式或提示结构，能够产出最准确且相关性最高的结果。
- **输出分析:**仔细审查 AI 生成的输出，查看是否存在与测试依据等方面相关的不准确或不一致的情况。了解错误和不一致的类型，有助于优化提示词，从而在后续迭代中避免类似问题。
- **整合用户反馈:**收集测试人员对生成输出的实用性和清晰度（如生成测试的详细程度）的反馈。分析他们的反馈，并利用这些反馈优化提示词，以更好地满足实际测试需求。
- **调整提示词长度与精准度:**尝试不同长度和详细程度的提示词。有时，增加更多上下文可提高响应的质量；但在某些情况下，较短的提示词可能更具通用性。

借助这些技巧，测试团队能够对提示词组织评估与优化工作，确保对生成式 AI 提示词的持续改进。通过在测试团队或测试组织内部共享实践经验，这不仅有助于规范提示词使用方法、保持质量稳定，还能营造学习与迭代改进的文化氛围。这种协作方式可以推动生成式 AI 测试方法的不断发展，测试团队借此依托集体智慧进步，避免重复犯错，并通过共享提示词库等方式，在长期过程中更有效地优化对生成式 AI 工具的运用。

**实践目标 2.3.2 (H1)：针对特定的测试任务，对提示词进行评估与优化**

本次练习聚焦于将提示词优化技术应用到特定测试任务中。参与者将从初始提示词开始，通过迭代优化，提升生成式 AI 生成结果的质量。他们会采用 A/B 测试、人工验证等方法，评估并改进提示词质量。目标是让参与者切实体验到，迭代优化是如何实现生成更有效且贴合周境的测试用例。

练习结束时，参与者将完成多轮提示词优化迭代，并利用之前讨论的指标评估每次迭代，从而提高生成式 AI 的输出质量。

# 3. 生成式 AI 在软件测试中的风险管理 - 160 分钟

**关键词**

安全性（security），漏洞（vulnerability），数据隐私（data privacy）

**生成式 AI 专用关键词**

幻觉（hallucination），温度系数（temperature），推理错误（reasoning error），偏差（bias）

**第三章 学习目标：**

### 3.1 幻觉、推理错误及偏差

| | | |
|---|---|---|
| GenAI-3.1.1 | (K1) | 回顾生成式 AI 系统中幻觉、推理错误及偏差的定义。 |
| GenAI-3.1.2 | (K3) | 识别大语言模型输出内容中的幻觉生成、推理错误及偏差问题。 |
| HO-3.1.2a | (H1) | 针对生成式 AI 测试中的幻觉现象进行实验。 |
| HO-3.1.2b | (H1) | 针对生成式 AI 测试中的推理错误现象进行实验。 |
| GenAI-3.1.3 | (K2) | 总结软件测试任务中应对生成式 AI 的幻觉、推理错误及偏差的缓解技巧。 |
| GenAI-3.1.4 | (K1) | 回顾针对大语言模型非确定性行为的缓解技术。 |

### 3.2 生成式 AI 在软件测试中的数据隐私与安全风险

| | | |
|---|---|---|
| GenAI-3.2.1 | (K2) | 解释软件测试中运用生成式 AI 所引发的关键数据的隐私与安全风险。 |
| GenAI-3.2.2 | (K2) | 举例说明在软件测试中使用生成式 AI 时的数据隐私与漏洞问题。 |
| GenAI-3.2.3 | (K2) | 总结生成式 AI 应用于软件测试时，保障数据隐私与强化安全性的缓解策略。 |
| HO-3.2.3 | (H0) | 在特定的生成式 AI 测试案例研究中，识别数据隐私与安全风险。 |

### 3.3 生成式 AI 用于软件测试时的能耗与环境影响

| | | |
|---|---|---|
| GenAI-3.3.1 | (K2) | 解释任务特性和模型使用方式，对软件测试中使用生成式 AI 能耗的影响。 |
| HO-3.3.1 | (H1) | 使用模拟器，计算生成式 AI 执行给定测试任务时的能耗及二氧化碳排放量。 |

### 3.4 AI 法规、标准与最佳实践框架

| | | |
|---|---|---|
| GenAI-3.4.1 | (K1) | 回顾软件测试中与生成式 AI 相关的 AI 法规、标准以及最佳实践框架示例。 |

# 3.1 幻觉、推理错误及偏差

生成式 AI 系统，特别是大语言模型，容易出现某些特定缺陷，包括幻觉生成、推理错误以及偏差。这些缺陷会降低生成式 AI 在测试任务中的输出质量，导致生成的测试工件无法达到测试人员的预期。测试人员需要识别大语言模型输出中的这些幻觉生成、推理错误以及偏差情况，并采取措施降低这些风险。

大语言模型的非确定性行为（见 1.1.2 节）使得这类缺陷难以修复；它们可能在大语言模型的某一次输出中看似已得到修复，但在与同一大语言模型的另一次对话中又会再度出现。

## 3.1.1 生成式 AI 中的幻觉生成、推理错误及偏差

当大语言模型生成的内容存在事实性错误，或者与给定任务不相关时，就会出现“幻觉”现象。在软件测试领域，“幻觉”可能体现为大语言模型编造虚构或无关的测试用例、生成错误或无法运行的测试脚本，亦或是推荐用于验证并不存在的验收准则的测试用例。这可能误导测试人员，降低测试输出的可信度。

当大语言模型错误解读逻辑结构，如因果关系、条件逻辑或循序渐进的问题解决过程，并由此得出错误结论时，就会出现推理错误。与人类不同，大语言模型缺乏真正的逻辑推理能力，而是依赖模式匹配。这就导致在执行诸如数学推理之类的任务时，可能出现逻辑错误（Mirzadeh 2024）。测试规划和测试用例优先级排序就是需要逻辑推理的测试任务示例，大语言模型在这些任务中可能会犯推理错误。

大语言模型的偏差（Gallegos 2024）源于模型所训练的数据。这些偏差可能导致输出结果偏向某些特定类型的信息、方法或假设。例如，主要基于英语数据训练的大语言模型，可能对非英语视角的呈现不足。在软件测试中，当生成测试数据或完善测试用例的验收准则时，偏差可能会影响大语言模型的响应。

生成式 AI 输出中出现的幻觉、推理错误和偏差，是由其训练数据的特性以及 Transformer 模型的内在局限性所致（详见第 1 章）。识别并应对这些挑战，能够提高测试过程中生成式 AI 输出结果的质量。

### 3.1.2 识别大语言模型输出中的幻觉、推理错误与偏差

要将生成式 AI 系统有效地融入软件测试，需要具备检测大语言模型输出中幻觉、推理错误及偏差的能力。根据问题类型的不同，可以采用不同的检测方法。以下是通过评审，或评审与自动验证相结合的方式来实施的常见方法：

幻觉检测：

- 交叉验证：将 AI 生成的输出与现有文档、需求及已知系统行为进行比对。自动化工具可协助将输出与既定数据源进行交叉核对，以标记差异。
- 咨询领域专家：邀请领域专家对生成内容的准确性进行核验。他们的专业见解对于捕捉自动化系统可能遗漏的细微差别至关重要。
- 一致性检验：核实生成的输出内容相互之间以及与已知信息是否一致。自动化系统能够辅助识别规律，并标记出不一致的地方。

推理错误检测：

- 逻辑确认：通过多轮评审，评估人工智能生成内容的逻辑流（例如生成文本中的一致性、连贯性和结构化推理），以确保其连贯性与正确性。自动化工具能提供协助，但复杂情况可能仍需人工判断。
- 输出测试：例如，将生成的测试用例或测试脚本运行于测试对象上以验证测试结果。根据生成的测试类型，此过程可以部分或完全自动化。

偏差监测：

- 评审所生成的测试工件（如合成测试数据），看其在多大程度上依据测试策略进行了公正、准确的呈现。
- 评估与测试类型相关的偏差，例如大语言模型生成的输出中对非功能测试的体现不充分。

这些检测方法具体如何实施，取决于运用生成式 AI 执行测试任务时，所预估的幻觉、推理错误或偏差的风险程度。

**实践目标　3.1.2a (H1)：针对生成式 AI 测试中的幻觉现象进行实验**

本练习着重围绕软件测试知识体系，开展针对生成式 AI 幻觉示例的实验。参与者需促使至少两个大语言模型处于特定情境，在此情境下，这些模型会编造出不相关的元素，例如增添给定上下文数据中并不存在的无关标准。研究人员将对提示词的变体进行测试，以探究提示词对幻觉现象产生的影响。

本练习有助于加深对在软件测试中识别生成式 AI 幻觉的理解。

**实践目标 3.1.2b (H1)：针对生成式 AI 测试中的推理错误现象进行实验**

本练习聚焦于展示一个生成式 AI 推理错误的实例。以测试规划领域中一个待解决的问题为例，比如测试工作量的估算以及测试用例的优先级排序（见 [ISTQB_CTFL] - 第 5 章）。练习所设计的输入数据具有一定复杂性，需要具备解决问题的能力，以此凸显大语言模型在这方面的局限性。大语言模型给出的结果将与应达成的准确结果进行比对。我们会尝试三种不同类型的大语言模型（大语言模型、小语言模型和推理模型），并通过变换提示词来努力优化结果。

本练习有助于提升对如何在需要运用逻辑推理能力来解决软件测试问题的任务中，识别生成式 AI 推理错误的理解。

## 3.1.3　生成式 AI 在软件测试任务中幻觉、推理错误及偏差的缓解技巧

为了最大程度减少生成式人 AI 在软件测试中产生的不良结果，可以采用多种策略来降低幻觉、推理错误及偏差出现的概率。当提示词设计不合理（详见第 2 章），或者给定的测试任务缺乏相关的上下文输入数据时，这些问题更容易发生。缓解与人工智能幻觉、推理错误和偏差相关风险的关键技术包括：

- 提供完整上下文：确保提示词包含所有相关信息（详见第 2.1.1 节），提供全面的背景信息，以引导 AI 输出准确结果。
- 将提示词拆分为易处理的片段：运用提示词链技术（详见第 2.1.2 节），将复杂提示词拆解为一个个较小步骤，每前进一步前，都对输出内容进行系统验证。这种逐步推进的方式，有助于在生成过程的早期阶段就发现推理错误。

- 采用清晰且易解读的数据格式：避免使用那些可能模糊不清、生成式 AI 难以理解的数据格式。结构规整、简单直白的数据格式，能帮助模型聚焦于任务的核心要点。
- 为任务选择合适的生成式 AI 模型：选用专门为当前任务训练的大语言模型（见 5.1.3 节）。
- 对比不同模型的结果：在适当情况下，用多个大语言模型对提示词进行评估，并对比输出结果，这样有助于发现输出错误，进而选出最可靠的结果。

第 4 章介绍了两种用于改进大语言模型输出结果的技术，分别为检索增强生成和微调技术。

### 3.1.4 大语言模型非确定性行为的缓解技术

大语言模型内在的非确定性行为（Shuyin 2023）意味着，即便输入相同，其输出也可能有所不同。这源于推理过程中所采用的概率采样方法。因此，在使用大语言模型时，要获得一致且可重复的结果颇具挑战，尤其是对于较长的输出内容，这会增加结果出现差异的风险。

虽然无法保证结果完全可重现，但还是有某些策略可以有助于降低这种差异性：

- 调整大语言模型的温度参数（temperature parameter）设置：在生成响应（推理）过程中降低温度参数，能缩小概率分布范围，减少随机性，进而产生更为一致的输出结果。不过，这样做也会限制响应的创造性与多样性，使输出内容变得更加重复，或者过于确定。
- 设置随机种子：部分大语言模型的实现方式允许为随机数生成器设置种子值，确保使用相同的伪随机（即具有确定性的随机值）序列，以此提高结果的可重复性。

要降低大语言模型（LLM）输出时产生幻觉及推理错误的风险，就必须处理这种非确定性行为。例如，可以对输出验证的部分环节进行自动化处理，以此确保评估过程既结构化又连贯一致。

## 3.2 生成式 AI 在软件测试中的数据隐私与安全风险

由于生成式 AI 在测试中需要处理敏感信息，且基于大语言模型的测试基础设施可能存在潜在漏洞，这就带来了数据隐私与安全方面的风险。因此，必须实施强有力的数据保护措施，以防止数据泄露、未经授权的访问以及机密数据的暴露

### 3.2.1 使用生成式 AI 涉及的数据隐私与安全风险

生成式 AI 能够处理海量数据，而这些数据可能包含敏感信息或个人身份识别信息。由此会引发以下数据隐私方面的担忧：

- 无意的数据泄露：生成式 AI 模型生成的输出内容，有可能会在无意间泄露敏感信息。
- 数据使用缺乏管控：生成式 AI 工具或许会在未获得用户明确同意或不受用户管控的情况下，存储并处理敏感数据。这可能引发潜在的滥用情况，或者导致数据遭到未经授权的访问。
- 合规风险：若使用生成式 AI 工具时未遵循数据保护法规，例如《通用数据保护条例》（GDPR，《欧盟条例》(EU) 2016/679 ），则可能引发法律争端。

此外，利用生成式 AI 进行测试时会出现特定的安全风险，比如：

- 基于大语言模型的测试基础设施可能易遭受安全攻击，像数据泄露或未经授权的访问。
- 恶意行为者能够利用大语言模型的漏洞，例如操纵性攻击（见 3.2.2 节），来改变其运行行为或提取敏感信息。
- 攻击者会蓄意引入恶意输入数据，以此误导大语言模型，破坏其准确性或安全性。

### 3.2.2 生成式 AI 用于测试过程与工具时的数据隐私及漏洞问题

下表给出了生成式 AI 测试过程和工具中一些攻击向量的示例。

| **攻击向量** | **描述** | **示例** |
|---|---|---|
| 数据泄露/Data exfiltration | 发送专门用于提取机密训练数据的请求。 | 例如，利用超长提示词使大语言模型的上下文窗口超出负荷，令其内存过载，这可能致使模型随机泄露训练数据片段，从而有暴露敏感信息的风险。 |

| | | |
|---|---|---|
| 请求篡改/Request manipulation | 引入会破坏 AI 输出的数据。 | 比如，一些图像会诱使人工智能进入不同的上下文语境，进而在验收准则等方面引发误判。 |
| 数据投毒/Data poisoning | 即对训练数据进行操控。 | 例如，在给 AI 生成的测试报告结果打分时，给出虚假评价。 |
| 恶意代码生成/Malicious code generation | 指在使用过程中操控大语言模型生成后门程序（比如外部命令调用）。 | 例如，生成代码以与某个特定的恶意 IP 地址建立通信信道。 |

### 3.2.3 使用生成式 AI 进行测试时保护数据隐私及提升安全性的缓解策略

随着生成式 AI 成为主流，以及其固有的风险，出现了旨在缓解这些风险的法规和标准（参见第 3.4.1 节）。

诸如《通用数据保护条例》（GDPR）之类的数据保护法规，并未明确对生成式 AI 的应用加以限制，不过确实提供了一些保障机制，这些机制可能会对相关行为构成约束，特别是在数据收集、处理和存储的合法性及目的限制方面。

为减轻这些风险，企业应当实施健全的数据隐私保护措施，其中包括：

- 数据最小化：除非获得法律许可，否则应避免处理敏感数据。在 AI 测试中，仅使用必要数量的非敏感数据，以此降低数据隐私风险。
- 数据匿名化与化名处理：采用不可识别的数据来遮蔽或替换敏感信息。
- 安全的数据存储与传输：实施强加密和访问控制。
- 人员培训：组织应制定清晰的培训计划和政策，确保生成式 AI 工具得到妥善使用，推广符合道德规范的用法，并降低潜在风险。

在运用生成式 AI 开展测试时，还可考虑以下额外的缓解策略：

- 对生成内容进行系统性评审：人工评估对于确保由生成式 AI 执行的测试任务的质量与准确性起着关键作用。

- 通过与其他大语言模型对比进行评估：此方法需针对特定任务启用多个大语言模型，通过对比它们的回答来评估输出结果。
- 选择安全操作环境：依据所需保密级别，组织可选择不同安全解决方案，如采用大语言模型供应商的商业安全服务，在安全的云环境中运行大语言模型，或在组织自有的基础设施中安装大语言模型。
- 定期安全审计与漏洞评估：及时发现并解决生成式 AI 系统的薄弱之处。
- 紧跟安全最佳实践：持续关注最新的安全指南和技术发展动态。

这些策略往往相辅相成，为确保在使用生成式 AI 过程中的数据安全，需将它们结合运用。强烈建议，若组织中有高级安全工程师、法律顾问、首席技术官（CTO）或首席信息安全官（CISO），应让他们参与其中。

**实践目标 3.2.3 (H0):在特定的生成式 AI 测试案例研究中，识别数据隐私与安全风险**

本演示意在阐明，在软件测试过程中运用生成式 AI 时，数据隐私与安全风险是如何产生的。参与者将分析案例，从而识别诸如模型漏洞、未经授权的数据访问，或是对生成输出结果的恶意使用等潜在威胁。他们还将探讨缓解策略，包括安全的数据处理方式、严密的访问控制，以及 AI 监测手段，同时思索其中涉及的伦理及实际影响。

最终，参与者将理解数据隐私原则，并学会在生成式 AI 测试环境中识别与应对安全风险。

## 3.3 生成式 AI 在软件测试中的能耗及对环境的影响

诸如（Luccioni 2024a）等研究显示，训练与处理大语言模型，需大量密集使用专业计算资源。大语言模型以基于网络的服务形式供人使用，这增加了设备、网络及数据中心的负担，致使能耗升高。

### 3.3.1 使用生成式 AI 对能耗与二氧化碳排放的影响

生成式 AI 对环境的影响不容小觑，因为随着其使用量的增加，能源消耗会急剧上升。任务的复杂程度以及所需的计算资源，都会影响能源消耗情况。例如，利用功能强大的 AI 模型生成一张图片所消

耗的能源，可能与将一部智能手机完全充满电所消耗的能源相当，而生成文本却仅相当于消耗智能手机电量的一小部分（Heikkilä 2023）。

尽管很难获取生成式 AI 对环境影响的精确数据（Luccioni 2024b），但显然，这些高能耗操作共同导致了大量二氧化碳排放（Berthelot 2024）。虽然单次搜索或文本生成任务可能看似微不足道，但全球数百万用户进行这些操作所产生的累积效应，会给环境带来巨大压力。

采取诸如限制不必要的模型交互等最佳实践，对于减轻生成式 AI 带来的环境风险至关重要。

**实践目标 3.3.1 (H1)：使用模拟器，计算生成式 AI 执行给定测试任务时的能耗及二氧化碳排放量**

本练习聚焦于评估软件测试中各种生成式 AI 任务的能源消耗及相关二氧化碳排放情况。参与者将通过模拟运算得出这些数据指标，并研究不同的任务特点和模型使用方式对环境造成的影响。

通过观察不同因素对能源消耗和排放量的作用，参与者将理解影响大语言模型能源消耗的因素。

## 3.4 AI 法规、标准与最佳实践框架

生成式AI通过辅助测试人员执行各类测试任务（详见第 2 章），正在重塑软件测试行业。然而，这些机遇也带来了诸如推理错误、数据隐私、安全漏洞和环境影响等重大风险（见 3.1、3.2 和 3.3 节）。要应对这些风险，需要考虑适用于 AI 的通用法规、标准和最佳实践框架。

### 3.4.1 软件测试中与生成式 AI 相关的 AI 法规、标准及框架

以下是软件测试中使用生成式 AI 所涉及的关键指南概述：

| **名称 / 类型** | **描述** | **在软件测试中的应用** |
| --- | --- | --- |
| **ISO/IEC 42001:2023《信息技术 - 人工智能 - 管理体系》/ISO/IEC 42001:2023 Information technology - Artificial intelligence-Management system**<br><br>类型：标准 | 该标准规定了组织内部管理人工智能系统的各项要求。 | 在软件测试中，此标准确保用于测试的生成式 AI 遵循推荐的操作规范，以提升测试的一致性与可靠性。 |
| **ISO/IEC 23053:2022《使用机器学习的人工智能（AI）系统框架》/ ISO/IEC 23053:2022 Framework for Artificial Intelligence (AI) Systems Using Machine Learning**<br><br>类型：标准 | 该标准为人工智能生命周期过程搭建框架，着重强调容错性与透明度。 | 在软件测试中运用生成式人工智能时，它为数据质量、透明度以及容错性构建了一个框架。 |
| **《欧盟人工智能法案》/EU AI Act**<br><br>类型：法规 | 该法案构建起应对人工智能风险的法律框架，依据风险程度对各类应用进行分类。<br><br>来源：(AI Act 2024) | 在软件测试中，该法案要求所使用的生成式人工智能，在透明度、责任归属以及偏差消除等方面必须符合规定。 |
| **NIST 人工智能风险管理框架（美国）/NIST AI Risk Management Framework (US)**<br><br>类型：框架 | 该框架提供了管理人工智能风险的指导方针，重点关注公平性、透明度和安全性。<br><br>来源： (NIST AI RMF 1.0) | 在软件测试中，它确保生成式人工智能的公平性，降低风险，避免测试结果出现偏差。 |

随着 AI 技术及其监管环境的不断演进，对于测试组织而言，及时掌握法规、标准、国家法律以及最佳实践框架（诸如本表所涉内容）的发展动态至关重要。

# 4. 基于大语言模型（LLM）驱动的软件测试基础架构 - 110 分钟

**关键词**

测试基础设施（test infrastructure）

**生成式 AI 专用关键词**

微调（fine-tuning），大语言模型驱动的智能体（LLM-powered agent），大语言模型运维（Large Language Model Operations），检索增强生成（retrieval-augmented generation），向量数据库（vector database）

**第四章 学习目标：**

### 4.1 基于大语言模型驱动的测试基础设施架构方法

| | | |
|---|---|---|
| GenAI-4.1.1 | (K2) | 解释基于大语言模型（LLM）驱动的测试基础设施的关键架构组件和概念。 |
| GenAI-4.1.2 | (K2) | 总结检索增强生成技术。 |
| HO-4.1.2 | (H1) | 针对给定的测试任务，对检索增强生成技术进行实验。 |
| GenAI-4.1.3 | (K2) | 解释基于大语言模型（LLM）驱动的智能体在自动化测试过程中的作用和应用场景。 |
| HO-4.1.3 | (H0) | 观察基于大语言模型驱动的智能体在协助自动化重复性测试任务方面的表现。 |

### 4.2 微调与大语言模型运维（LLMOps）：生成式 AI 在软件测试中的应用实践

| | | |
|---|---|---|
| GenAI-4.2.1 | (K2) | 解释语言模型针对特定测试任务的微调策略。 |
| HO-4.2.1 | (H0) | 观察针对给定测试任务与语言模型的微调过程示例。 |
| GenAI-4.2.2 | (K2) | 解释大语言模型运维（LLMOps）及其在测试任务中部署和管理大语言模型（LLM）所发挥的作用。 |

# 4.1 基于大语言模型驱动的测试基础设施架构方法

AI 聊天机器人以及由大语言模型驱动的测试工具，是两类运用大语言模型的测试基础设施（详见 1.2.2 节）。

除了基于大语言模型驱动的测试基础设施的基本架构（详见 4.1.1 节）之外，检索增强生成（详见 4.1.2 节）以及大语言模型驱动的智能体架构（详见 4.1.3 节），拓展了大语言模型在软件测试中的功能及实用性。

## 4.1.1 基于大语言模型驱动的测试基础设施的关键架构组件与概念

基于大语言模型驱动的测试基础设施，是指将大语言模型融入软件测试过程中，用以增强自动化程度、推理能力和决策能力的系统。与主要专注于对话交互的传统 AI 聊天机器人不同，基于大语言模型驱动的测试工具旨在通过处理测试相关的询问、分析需求、生成测试用例以及评估输出，为软件测试提供支持。

基于大语言模型驱动的测试基础设施的典型架构采用多组件设计，以实现与大语言模型安全且高效的交互。该架构由前端和后端组件、外部数据源以及集成的大语言模型构成：

- 前端作为用户界面，测试人员可通过该界面输入查询内容或指令与系统进行交互。
- 后端负责处理用户输入，并管理身份验证、数据检索、提示词准备以及与大语言模型的交互等关键功能。
- 大语言模型可作为第三方服务（通过 API 访问）或定制的内部模型来部署，并基于结构化的提示词生成响应。

这种架构超越了传统的客户端 - 服务器模型，融入了诸如大语言模型和多源后端等智能处理组件：

1. 大语言模型不只是一个服务器，更是一个能够基于测试产品进行解读与推理的智能处理组件。
2. 基于规则的聊天机器人按预设脚本做出回应，与之不同的是，由大语言模型驱动的测试基础设施，能够依据需求、代码或测试结果等上下文信息，动态生成对测试的深刻见解。
3. 后端集成了多种数据源，例如：

- 关系型数据库（用于存储测试中用到的结构化数据，例如测试用例）
- 向量数据库（借助嵌入式技术实现相关内容的语义检索；见 4.1.2 节）。

4. 后端对大语言模型的原始输出进行后处理，确保其回复在呈现给前端之前，符合测试过程中的测试条件。

## 4.1.2 检索增强生成

检索增强生成（RAG）通过在大语言模型生成回复的过程中引入额外数据源，提升了大语言模型的能力（Zhao 2024），进而提高了其输出内容的相关性和准确性。

检索增强生成（RAG）将检索系统与大语言模型相结合，生成具备上下文感知能力的响应结果。在预处理阶段，大型文档会被拆分为较小的片段（例如 256-512 个词元(tokens)），以确保检索具有针对性，并与模型的上下文窗口兼容。通过预训练模型，对每个片段进行清洗、处理，然后编码为高维向量（嵌入）。这些嵌入向量可存储于向量数据库中，在运行时（推理阶段）实现基于相似度的高效检索。用户查询被编码后，系统根据语义相似度检索出相关文本块，并将这些文本块作为上下文，供语言模型生成有依据的响应结果。

所谓相关响应本质上是语言模型生成的输出，它深度依赖检索过程中收集到的相关、准确且符合上下文的信息。它确保响应不仅基于模型已有的预训练内容，还融入了与提示词相关的精准数据。检索与生成之间的协同效应，提高了响应结果的准确性和相关性，让用户得到的回复更可靠、信息更丰富。

在用户提示词处理阶段，检索增强生成系统的工作过程分为以下两步：

1. 检索：对于用户的查询，系统会从之前创建好的向量数据库里检索相关信息。这种检索一般依据提示词的嵌入向量与文本片段嵌入向量之间的语义相似性来进行。
2. 生成：接着，把检索到的信息输入到大语言模型中。大语言模型会将自身已有的知识和新获取的数据相结合，进而生成更为准确、契合上下文的输出内容作为回应。

在软件测试领域，检索增强生成（RAG）让基于大语言模型的测试基础设施能够接入公司的各类企业数据源，如数据库、文档资料及存储库等，从而实时检索上下文信息。这样一来，诸如测试分析、测试设计之类的测试任务，就能与最新的规范、需求以及现有的测试数据相契合。

**实践目标 4.1.2 (H1)：针对给定测试任务，试用检索增强生成技术**

本实践练习着重于将检索增强生成（RAG）技术应用于给定的测试任务。参与者将通过引入文档，对 RAG 系统进行试验，观察其如何依据复杂信息生成准确度各异的答案。参与者需针对给定测试任务，比较大语言模型在使用和未使用 RAG 时的输出。此练习旨在明确 RAG 系统在处理不同类型测试任务时的优势与局限。

通过审视检索到的数据及生成的结果，参与者将深入了解 RAG 在强化大语言模型驱动的测试过程中所起的作用。

### 4.1.3 大语言模型驱动的智能体在自动化测试过程中的作用

大语言模型驱动的智能体（Wang 2024）是一类由大语言模型赋能的特定生成式 AI 应用，旨在半自主或自主处理既定任务计。其核心机制在于：依托大语言模型实现自然语言的理解与生成，同时具备处理指令、检索上下文信息并采取智能行动的能力。

与传统 AI 聊天机器人不同，后者仅仅专注于问答式交互，而大语言模型驱动的智能体可通过调用一组预先设定好的功能（通常称为“工具”）来执行任务或“采取行动”。这种能力使得它们能够与外部系统进行交互并加以操控，进而在任务执行方面展现出极高的通用性。大语言模型驱动的智能体在自主程度上存在差异：

- 独立运行的自主智能体，依据预定义规则、强化学习以及自适应反馈循环，在几乎无需人工干预的情况下执行各项任务。
- 半自主智能体，执行任务期间，需要人工定期进行监督，以此确保产出符合用户预设的目标。

多智能体架构打造的是一个协作系统。在这个系统里，多个智能体各有专长，它们相互沟通、协同作业，相比单个智能体，能更高效地解决复杂问题。多个 AI 智能体间这种协调合作的方式，就叫做“编排”（orchestration）。

在测试过程里，靠大语言模型驱动的智能体，能通过模仿人类的推理与决策过程，实现测试任务自动化。但这类智能体也摆脱不了使用大语言模型时会出现的问题，像可能产生幻觉、推理出错以及出现偏差（见 3.1 节）。这些智能体给出的结果可能有误或具误导性，这就会降低自动化测试过程的

可靠性。要减轻这些风险，可以给智能体得出的结果安排自动化验证规程，或者在关键任务中使用半自主式智能体。

**实践目标 4.1.3 (H0)：观察由大语言模型驱动的智能体如何助力重复性测试任务的自动化**

本演示着重展示大语言模型驱动的智能体执行的一项测试任务。为说明如何在测试过程中融入基于智能体的解决方案，我们会展示输入给智能体的数据、智能体的行为以及其操作产生的结果。

本演示给出了大语言模型驱动的智能体在测试任务场景中的一个具体示例。

# 4.2 微调与大语言模型运维（LLMOps）：生成式 AI 在软件测试中的实践

若要把大语言模型驱动的测试架构运用到实际测试中，有两个关键做法：一是对大语言模型进行微调，二是借助大语言模型运维（LLMOps）来管控操作流程（Mailach，2024）。

## 4.2.1 针对测试任务微调大语言模型

微调，就是让预训练好的语言模型，比如大语言模型（LLM）或者小语言模型（SLM，详见 1.1.2 节），能够执行特定任务，或是适配特定领域（Parthasarathy 2024）。具体做法是，在特定的数据集上进一步训练模型，让它学到特定领域的知识，把握其中的细微之处。经过微调，模型在专业应用场景中的表现会更好，在既定的使用场景下，输出会更准确，也更符合实际需求。

在实际操作中，微调能让通用的大语言模型拥有特定领域的专业推理能力，或者让它适应这个领域独有的词汇。微调对资源需求较低的小语言模型（SLMs）同样适用。微调小语言模型后，执行特定任务时，不用像大语言模型那样耗费大量计算资源，就能有更好的性能表现。这种对比体现出，根据任务的具体需求选择大语言模型或小语言模型，可以兼顾灵活性和效率。

例如，在软件测试中，微调能够让大语言模型或小语言模型，按照组织的特定情境，将用户故事转化为特定输出格式的测试用例。通过使用组织内的用户故事以及相应的测试用例来训练模型，模型就能与组织特定的测试过程和术语相契合。

为软件测试微调生成式 AI 模型会遇到不少挑战：

- 要确保使用高质量、特定任务的训练数据集，以此避免出现有偏差或不准确的结果。

- 缓解过拟合现象（模型过度适应训练数据，会对其处理新的、未见过的数据的表现产生负面影响），确保模型在不同场景中都具有泛化能力。

- 解决模型推理过程中的不透明问题（大语言模型的决策逻辑和输出生成机制缺乏透明度），这一问题会使调试与确认工作变得复杂。

- 管理微调过程中（针对大语言模型）会需要大量计算资源。

**实践目标 4.2.1 (HO)：观察针对给定测试任务及大语言模型 / 小语言模型的微调过程示例**

本演示将呈现针对特定测试任务，对大语言模型或小语言模型进行微调所涉及的各个步骤。首先，挑选合适的大语言模型或小语言模型；接下来，展示为给定测试任务量身定制的数据集，随后呈现微调过程的示例解决方案（如某个机器学习框架），最后向微调后的模型发送提示，进而探讨生成输出的质量。

本次针对测试任务的大语言模型 / 小语言模型微调流程演示，展示了该过程的几个关键方面，尤其聚焦于训练数据的质量问题。

## 4.2.2 大语言模型运维：面向软件测试的大语言模型部署与管理

大语言模型运维（LLMOps，Large Language Model Operations）指一系列在生产环境中开发、部署和维护大语言模型而设计的方法、工具和过程（Sinha 2024）。

在组织的测试过程中应用生成式 AI 可通过几种不同方式实现，不同方式将影响大语言模型运维决策的制定。以下是三种可行的方法：

- 使用 AI 聊天机器人：此方法的主要考量在于，在优化成本的同时管控数据隐私与安全风险。若能获得必要的保障，企业可选用大语言模型即服务（LLM-as-a-Service）平台；若想实现更自主的管控，也可利用开源许可的大语言模型搭建内部基础设施。为降低数据隐私与安全风险（详见 3.2 节）并确保运营效率，对供应商保障能力或自身内部能力进行严格评估至关重要。

- 使用具备生成式AI能力的测试工具：此方法与使用 AI 聊天机器人有类似考量，如数据隐私、安全及运营成本。此外，企业必须评估测试工具供应商所提供的数据安全与性能保障。这类测试工具通常作为现有测试过程的补充，需要对其开展全面的成本效益分析和风险评估。

- 自主研发基于生成式 AI 的测试工具：此方式着重于全面把控数据隐私与安全风险，同时针对计算资源、数据存储、人员培训等人工智能运营成本，进行细致规划。企业还必须构建结构化过程，对生成式 AI 特定的开发成果予以验证与维护。自主研发解决方案，需要掌握基于大语言模型构建测试基础设施的实施与部署专业技能。

这些方法并非相互排斥。企业可能在一些任务上选用 AI 聊天机器人，而在另一些任务中自行开发定制工具。所以，企业可依据具体的测试活动，同时运用这些方法。此外，还能融入像检索增强生成（RAG）、大语言模型或小语言模型微调等其他技术，来提升生成式 AI 在测试过程里的效果和适应能力。

# 5. 在测试组织开展生成式 AI 的部署与集成工作 - 80 分钟

**关键词**

无

**生成式 AI 专用关键词**

影子 AI（shadow AI）

**第五章 学习目标：**

**5.1 在软件测试中采用生成式 AI 的路线图**

| | | |
|---|---|---|
| GenAI-5.1.1 | (K1) | 回顾影子 AI 的风险。 |
| GenAI-5.1.2 | (K2) | 阐述制定软件测试生成式 AI 策略时需重点考量的关键要素。 |
| GenAI-5.1.3 | (K2) | 总结在特定场景下为软件测试任务选择大语言模型 / 小语言模型的关键标准。 |
| HO-5.1.3 | (H1) | 估算在给定测试任务中使用生成式 AI 的周期性成本。 |
| GenAI-5.1.4 | (K1) | 回顾测试组织运用生成式 AI 的关键阶段。 |

**5.2 在软件测试中采用生成式 AI 的变更管理**

| | | |
|---|---|---|
| GenAI-5.2.1 | (K2) | 阐述测试人员在测试过程中有效运用生成式 AI 所需的核心技能与知识领域。 |
| GenAI-5.2.2 | (K1) | 回顾在测试团队中培养 AI 技能以支持在测试活动中采用生成式 AI 的策略。 |
| GenAI-5.2.3 | (K1) | 认识到测试组织在采用生成式 AI 时，测试过程和职责如何演变。 |

# 5.1 在软件测试中采用生成式 AI 的路线图

要制定一份采用生成式 AI 的测试策略，必须仔细考虑几个关键因素，包括预期达成的测试目标、如何挑选合适的大语言模型（LLM）、用作提示词的输入数据可能存在的问题，以及是否符合 AI 相关标准与法规。基于该策略，组织便可规划相关的路线图，并跟踪将生成式 AI 融入测试过程的进展。

## 5.1.1 影子 AI 的风险

影子 AI 可能带来安全、依从性和数据隐私方面的风险：

- 信息安全与数据隐私泄露：个人 AI 工具可能缺乏完善的安全防护，进而引发潜在的数据泄露风险。
- 依从性与监管难题：使用未经批准的AI工具可能导致违反行业标准和法规要求（参见第 3.4.1 节），并可能引发法律后果。
- 知识产权模糊：使用许可协议不明确的 AI 工具，可能使大语言模型用户面临知识产权纠纷，尤其是在未经恰当授权就处理受版权保护的数据时。

一套整合与部署生成式 AI 的策略和具体步骤，可以帮助测试组织规避影子 AI 的风险。

## 5.1.2 制定软件测试生成式 AI 策略时的关键要素

若要在测试工作中成功推行生成式 AI 策略，组织必须仔细考虑几个关键因素，以保障顺利整合并取得最佳成效。这首先要为生成式 AI 定义可衡量的测试目标，例如提高测试生产率、缩短测试周期和提升测试质量。选择合适的大语言模型至关重要（参见第 5.1.3 节），所选模型应与这些测试目标相契合，同时确保与现有测试基础设施的兼容性并满足系统可扩展性要求。

数据质量至关重要，因为基于大语言模型驱动的测试效果取决于准确、相关的输入数据，同时这些数据需有完善的安全规程予以保护。因此，维持高质量的输入数据，是获得可靠且正确结果的关键。

应提供全面的培训计划，以确保测试团队拥有有效使用生成式 AI 工具所需的技术和职业道德技能。除了培训之外，还应收集特定的指标，用以衡量生成式 AI 成果的有效性（参见第 2.3.1 节）。

为了确保符合监管标准和遵守伦理准则，组织应制定使用生成式 AI 的流程指南，包括敏感数据使用规则、透明度要求（例如，标明哪些内容由生成式 AI 生成），以及针对生成的测试件进行评审的质量阀。

### 5.1.3 为软件测试任务选择大语言模型/小语言模型

大语言模型/ 小语言模型种类繁多，各自具备不同的功能特性（例如，多模态输入、推理能力）、技术特征（例如，上下文窗口大小）以及许可类型（例如，商业许可与开源许可）。虽然有许多基准可用于评估 大语言模型/ 小语言模型在自然语言处理、代码生成或图像分析等任务上的表现，但专门针对软件测试任务的基准并不多（Wenhan 2024）。因此，为测试任务选择 大语言模型/ 小语言模型需要仔细考虑以下几个关键标准：

- 模型性能：依据组织设定的基准，采用 2.3.1 节所述度量，评估模型在目标测试任务中的表现。
- 微调潜力：考量使用领域特定数据对语言模型（LLM 或 SLM）进行微调的可行性与价值，旨在提升特定场景中的性能，增强在专业场景下的准确性和相关性。
- 周期性成本：考虑使用语言模型（LLM 或 SLM）的持续成本，包括许可费用和运营开支，以确保其符合组织针对目标测试任务的预算。
- 社区与支持：选择具有活跃社区支持和详细文档的模型，以助力模型的实施与故障排查。

通过仔细评估这些标准，测试组织可以选择一个或多个满足其特定需求和组织约束的语言模型（LLM 或 SLM）。

**实践目标 5.1.3 (H1)：估算在给定测试任务中使用生成式 AI 的周期性成本**

本练习主要围绕基于多种假设，估算在特定测试任务中使用生成式 AI 的周期性成本。这些假设因素包括输入和输出数据中的词元（tokens）数量、使用的提示词（prompts），以及任务执行的频率。我们将研究并对比多个大语言模型 / 小语言模型供应商的定价模式，其中至少有一个商业解决方案和一个开源许可模型。

本练习让大家有机会结合实际测试条件，计算和研究生成式 AI 的周期性成本，从而更好地理解不同方式和供应商带来的成本影响。

### 5.1.4 在软件测试中采用生成式 AI 的阶段

在测试组织中采用生成式 AI 包含三个关键阶段：

1. 探索阶段：第一阶段重点在于提升认知和构建能力。具体活动涵盖对测试团队开展生成式 AI 概念培训，提供语言模型（LLM 或 SLM）的访问权限以及试验初始用例，帮助测试人员熟悉生成式 AI 并建立信心。

2. 启动与使用定义阶段：一旦测试人员对生成式 AI 有了基本了解，第二阶段则重于识别生成式 AI 在软件测试中的实际应用场景并确定其优先级。此阶段还包括评估基于大语言模型的测试基础设施、培养专业技能，并确保与组织的需求保持一致（参见 [ISTQB_CTFL_SYL] 第 6 节）。

3. 应用与迭代阶段：在这一高级阶段，各组织会将生成式 AI 全面融入测试过程中。持续跟踪生成式 AI 在软件测试 及相关工具方面的进展情况，并对转型进行量化评估和管理，以确保能够获得可持续的效益并实现可扩展性。

对于不同使用场景，这些阶段可以并行推进。例如，测试报告分析在路线图上的进展可能较为靠前，而测试自动化还处于早期阶段。另外，意识到并处理像担心失业这类初期顾虑也至关重要，因为它们可能会影响生成式 AI 的应用情况以及团队士气。

## 5.2 在软件测试中采用生成式 AI 时的变革管理

测试组织若想成功实施生成式 AI，需要采用结构化的方式来处理变革管理过程。其中关键在于培养必要的生成式 AI 技能，以及推动传统测试角色的转变，使其适应融入 AI 的测试过程。这一转变涉及技术技能与组织层面两个方面。

### 5.2.1 运用生成式 AI 进行测试所需的核心技能与知识

要成功地将生成式 AI 整合到测试过程中，测试人员必须掌握提示工程技术，了解模型上下文窗口，以及开发测试评审方法。在诸如测试用例生成、缺陷报告分析和测试数据生成等任务中，测试人员需将领域专业知识和测试专长与 AI 技能相结合，以此评估大语言模型驱动的测试的效果。

核心能力包括评估大语言模型的能力、理解提示词优化技术以及评估 AI 生成的测试件。必备知识包括理解生成式 AI 的固有风险，以及对常见缓解策略的理解。测试人员应理解与大语言模型共享测试件的所涉及的数据安全影响，采取适当的数据清理措施（如删除或屏蔽敏感、个人或机密信息），并

遵循保护数据隐私的提示词工程实践。从环境角度考虑，要优化模型选择和使用模式，以降低计算开销，为测试任务选择规模合适的模型，并平衡生成式 AI 自动化带来的好处与对成本和能源消耗的影响。

### 5.2.2 在测试团队中打造使用生成式 AI 的能力

要从战略上培训测试团队使用生成式 AI 开展测试，运用实践操作的方法必不可少。具体来说，团队成员要实际操作各类大语言模型和小语言模型，按照结构化的学习路径学习，并通过在组织内交流分享，来逐步积累专业知识。培训重点是，借助有指导的练习、同伴之间的相互学习，以及将 AI 逐步融入日常测试任务中，来培养实际操作技能。

测试团队成员应先掌握基础的提示词编写方法，然后逐渐学会运用更具针对性的技巧，比如编写针对测试的提示词。提示词模式是一种可复用的模板，用它构建有效的提示词，能引导生成式 AI 给出连贯、可靠的输出。内部实践社区应支持知识持续共享，定期开会，重点分享生成式 AI 的成功应用案例，一起讨论遇到的挑战，完善最佳实践方案。社区通过共享提示词模式库，记录跨项目的、在跨领域的测试实施中从生成式 AI 获得的经验教训等，以此来推动持续改进。

### 5.2.3 在 AI 赋能的测试组织中测试过程的演变

生成式 AI 的融入，改变了测试组织内测试工程师和测试经理的传统测试过程。

测试工程师从专注于测试设计和测试执行的专家，转变为 AI 辅助测试专家，将自身在测试技术方面的专长，与指导和验证 AI 生成的测试件的技能相结合。他们的测试任务有所拓展，涵盖对基于 AI 的整体输出进行评审、优化提示词以及维护测试专用提示词库。

测试经理的职责也随之更新，需制定基于 AI 的测试策略、开展基于 AI 的风险管理，以及监控基于 AI 的测试过程。测试经理着重于平衡人与 AI 的能力、针对不同应用场景建立 AI 治理框架，确保测试团队既保持传统的测试能力，又具备 AI 知识素养。测试经理不仅要领导人类测试工程师，还要与生成式 AI 驱动的测试主体进行协调，这需要他们掌握新的管理技能，以监管由人类工程师和生成式 AI 工具组成的混合团队。

# 6. 参考文献

## 标准

ISO/IEC 42001:2023 (2023), Information technology — Artificial intelligence — Management system

ISO/IEC 23053:2022 (2022), Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)

## ISTQB®文档

[ISTQB_CTFL_SYL] ISTQB® Foundation Level Syllabus v4.0, 2023

## 术语表

ISTQB® Glossary https://glossary.istqb.org/

## 书籍

Winteringham M. (2024) Software Testing with Generative AI, Manning Publications (5 Mar. 2025), ISBN-13 : 978-1633437364, 10 Dec. 2024 - 304 pages

## 文章

(Berthelot 2024) Berthelot, Adrien, et al. "Estimating the environmental impact of Generative-AI services using an LCA-based methodology." Procedia CIRP 122 (2024): 707-712.

(Gallegos 2024) Gallegos, Isabel O., et al. "Bias and fairness in large language models: A survey." Computational Linguistics (2024): 1-79.

(Li 2024) Yihao Li, Pan Liu, Haiyang Wang, Jie Chu, W. Eric Wong, Evaluating Large Language Models for Software Testing, Computer Standards & Interfaces (2024), doi: https://doi.org/10.1016/j.csi.2024.103942

(Luccioni 2024a) Luccioni, Sasha, Yacine Jernite, and Emma Strubell. "Power hungry processing: Watts driving the cost of AI deployment?." The 2024 ACM Conference on Fairness, Accountability, and Transparency. 2024.

(Mailach 2024) Mailach, Alina, et al. "Practitioners' Discussions on Building LLM-based Applications for Production." arXiv preprint arXiv:2411.08574 (2024).

(Mirzadeh 2024) Mirzadeh, Iman et al. "GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models." ArXiv abs/2410.05229 (2024)

(NIST AI RMF 1.0) National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1, U.S. Department of Commerce, 2023, https://doi.org/10.6028/NIST.AI.100-1.

(Parthasarathy 2024) Parthasarathy, Venkatesh Balavadhani, et al. "The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities." arXiv preprint arXiv:2408.13296 (2024).

(Schulhoff 2024) Schulhoff, S., "The Prompt Report: A Systematic Survey of Prompting Techniques", Art. no. arXiv:2406.06608, 2024. doi:10.48550/arXiv.2406.06608.

(Shuyin 2023) Ouyang, Shuyin, et al. "LLM is Like a Box of Chocolates: the Non-determinism of ChatGPT in Code Generation." arXiv preprint arXiv:2308.02828 (2023).

(Sinha 2024) Sinha, Megha, Sreekanth Menon, and Ram Sagar. "LLMOps: Definitions, Framework and Best Practices." 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET. IEEE, 2024.

(Wang 2024) Wang, Yanlin, et al. "Agents in Software Engineering: Survey, Landscape, and Vision." arXiv preprint arXiv:2409.09030 (2024).

(Wenhan 2024) Wang, Wenhan, et al. "TESTEVAL: Benchmarking Large Language Models for Test Case Generation." arXiv preprint arXiv:2406.04531 (2024).

(Zhao 2024) Zhao, Penghao, et al. "Retrieval-augmented generation for ai-generated content: A survey." arXiv preprint arXiv:2402.19473 (2024).

## 网页

(AI Act 2024) European Commission. "European Approach to Artificial Intelligence." Shaping Europe's Digital Future, European Commission, https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence. Accessed 24 Nov. 2024.

(Heikkilä 2023) Heikkilä, M. (2023, December 1). Making an image with generative AI uses as much energy as charging your phone. MIT Technology Review. Retrieved from https://www.technologyreview.com/2023/12/01/1084189/making-an-image-with-generative-ai-uses-as-much-energy-as-charging-your-phone/

(Luccioni 2024b) Luccioni, S. (2024, February 22). Generative AI's environmental costs are soaring. Nature. Retrieved from https://www.nature.com/articles/d41586-024-00478-x

(Google Dev Glossary 2024) Google Developers. (n.d.). Machine learning glossary: Generative AI. Retrieved November 24, 2024, from https://developers.google.com/machine-learning/glossary/generative

(MIT 2024) "Glossary of Terms: Generative AI Basics." *MIT Sloan Teaching & Learning Technologies*, MIT Sloan School of Management, https://mitsloanedtech.mit.edu/ai/basics/glossary. Accessed 24 Nov. 2024.

上述参考文献，所涉信息源自互联网及其他渠道。虽然在本大纲发布之际，已对这些参考文献进行过核验，但若日后这些参考文献无法访问，国际软件测试认证委员会（ISTQB®）不承担相关责任。

# 7. 附录 A - 学习目标/知识认知级别

本大纲的具体学习目标列于各章开头。教学大纲中的每个主题都将根据其对应的学习目标进行考核。

学习目标以动词开头，该动词对应其知识认知级别的知识水平，具体如下所示。

**级别 1：牢记（K1）** - 考生要能牢记、认识和回顾一个术语或概念。

**行为动词：**回顾，认识

| 示例 | 注释 |
|---|---|
| 回顾测试金字塔的概念。 | |
| 识别典型的测试目的。 | |

**级别 2：理解（K2）** - 考生应能挑选出与主题相关陈述的原因或解释，还应能针对测试概念进行总结、对比、分类以及举例。

**行为动词：**分类、比较、区分、辨别、解释、举例说明、阐释、总结

| 示例 | 注释 |
|---|---|
| 根据测试工具的用途以及它们所支持的测试活动对其进行分类。 | |
| 比较不同的测试级别。 | 可用于寻找相似点、差异点或两者兼有。 |
| 区分测试与调试。 | 寻找概念之间的差异。 |
| 辨别项目风险和产品风险。 | 能够将两个（或更多）概念分别归类。 |
| 解释背景信息对测试过程的影响。 | |
| 举例说明为何测试是必要的。 | |
| 从给定的失效信息推断缺陷的根本原因。 | |
| 总结工作产品评审过程的各项活动。 | |

**级别 3：应用（K3）** - 考生在面对熟悉的任务时，能够执行某个程序，或者选择正确的程序并将其应用于给定的情境中。

**行为动词：**应用、实施、准备、使用

| 示例 | 注释 |
| --- | --- |
| 应用边界值分析方法，从给定需求中导出测试用例。 | 应参照某一规程/技术/过程等。 |
| 实施度量数据收集方法，以满足技术和管理需求。 | |
| 为移动应用程序准备易安装性测试。 | |
| 利用可追溯性来监控测试进度，确保其测试目的、测试策略和测试计划的完备性和一致性。 | 可用于期望考生能够运用某种技术或程序的学习目标中。与 “应用”类似。 |

**参考资料**

(适用于学习目标的认知水平)

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

# 8. 附录 B - 商业价值与学习目标追溯矩阵

本节列出了“认证测试工程师-生成式 AI 测试”的商业价值与学习目标之间的追溯关系。

实践目标未在此表中提及，因为每个实践目标都与单个学习目标相关联。实践目标与商业价值之间的追溯通过其关联的学习目标进行。

| 商业价值：使用生成式 AI 测试的认证测试工程师 | | | B01 | B02 | B03 | B04 | B05 |
|---|---|---|---|---|---|---|---|
| GenAI-B01 | 理解生成式 AI 的基本概念、能力及局限。 | | 8 | | | | |
| GenAI-B02 | 切实掌握针对软件测试场景，向大语言模型输入有效提示的实用技能。 | | | 10 | | | |
| GenAI-B03 | 深入了解在软件测试中使用生成式 AI 所面临的风险，以及相应的应对策略。 | | | | 11 | | |
| GenAI-B04 | 深入了解生成式 AI 解决方案在软件测试领域的具体应用。 | | | | | 19 | |
| GenAI-B05 | 切实助力组织内部软件测试生成式 AI 战略及路线图的制定与实施。 | | | | | | 13 |
| 专属 LO | 学习目标 | K-级别 | | | | | |
| **第 1 章** | **生成式 AI 在软件测试中的应用简介** | | | | | | |
| **1.1** | **生成式 AI 基础与关键概念** | | | | | | |
| GenAI-1.1.1 | 回顾机器学习的不同类型，包括传统机器学习、深度学习以及生成式 AI | K1 | X | | | | |
| GenAI-1.1.2 | 阐述生成式 AI（GenAI）与大语言模型（LLM）的基础原理 | K2 | X | | | | |
| GenAI-1.1.3 | 准确区分基础大语言模型、指令微调大语言模型以及大语言推理模型（LLM） | K2 | X | | | | |
| GenAI-1.1.4 | 归纳多模态大语言模型（LLM）与视觉 - 语言模型的基本原理 | K2 | X | | | | |
| **1.2** | **软件测试中生成式 AI 的应用：核心准则** | | | | | | |
| GenAI-1.2.1 | 列举将大语言模型（LLM）用于测试任务时所应展现的关键能力的示例 | K2 | X | | | X | |
| GenAI-1.2.2 | 对比在软件测试工作中采用生成式 AI（GenAI）时的交互模式 | K2 | X | | | X | |

| 商业价值：使用生成式 AI 测试的认证测试工程师 | | | B01 | B02 | B03 | B04 | B05 |
|---|---|---|---|---|---|---|---|
| **第 2 章** | **面向高效软件测试场景的提示词工程** | | | | | | |
| **2.1** | **高效提示词开发** | | | | | | |
| GenAI-2.1.1 | 列举在软件测试生成式 AI 应用中所采用的各类提示词结构示例 | K2 | | X | | | |
| GenAI-2.1.2 | 区分用于软件测试的核心提示技术 | K2 | | X | | | |
| GenAI-2.1.3 | 区分系统提示词和用户提示词 | K2 | | X | | | |
| **2.2** | **将提示词工程技术应用于软件测试任务** | | | | | | |
| GenAI-2.2.1 | 将生成式 AI 应用于测试分析任务中 | K3 | | X | | | |
| GenAI-2.2.2 | 将生成式 AI 应用到测试设计与测试实施任务中 | K3 | | X | | | |
| GenAI-2.2.3 | 将生成式 AI 应用于自动化回归测试任务中 | K3 | | X | | | |
| GenAI-2.2.4 | 将生成式 AI 应用于测试控制和监测任务中 | K3 | | X | | | |
| GenAI-2.2.5 | 基于给定的情境和测试任务，挑选适合的提示技术并加以应用 | K3 | | X | | X | |
| **2.3** | **评估生成式 AI 结果并优化软件测试任务提示词** | | | | | | |
| GenAI-2.3.1 | 理解用于评估生成式 AI 在测试任务结果上的度量 | K2 | | X | X | X | |
| GenAI-2.3.2 | 给出评估和迭代优化提示的方法示例 | K2 | | X | X | X | |
| **第 3 章** | **生成式 AI 在软件测试中的风险管理** | | | | | | |
| **3.1** | **幻觉、推理错误及偏差** | | | | | | |
| GenAI-3.1.1 | 回顾生成式 AI 系统中幻觉、推理错误及偏差的定义 | K1 | X | | X | X | |
| GenAI-3.1.2 | 识别大语言模型输出内容中的幻觉生成、推理错误及偏差问题。 | K3 | | | X | X | |
| GenAI-3.1.3 | 总结软件测试任务中应对生成式 AI 的幻觉、推理错误及偏差的缓解技巧 | K2 | | | X | X | |
| GenAI-3.1.4 | 回顾针对大语言模型非确定性行为的缓解技术 | K1 | X | | X | X | |
| **3.2** | **生成式 AI 在软件测试中的数据隐私与安全风险** | | | | | | |

| 商业价值：使用生成式 AI 测试的认证测试工程师 | | | B01 | B02 | B03 | B04 | B05 |
|---|---|---|---|---|---|---|---|
| GenAI-3.2.1 | 解释软件测试中运用生成式 AI 所引发的关键数据的隐私与安全风险 | K2 | | | X | X | |
| GenAI-3.2.2 | 举例说明在软件测试中使用生成式 AI 时的数据隐私与漏洞问题 | K2 | | | X | X | |
| GenAI-3.2.3 | 总结生成式 AI 应用于软件测试时，保障数据隐私与强化安全性的缓解策略 | K2 | | | X | X | |
| **3.3** | **生成式 AI 在软件测试中的能耗及对环境的影响** | | | | | | |
| GenAI-3.3.1 | 解释任务特性和模型使用方式，对软件测试中使用生成式 AI 能耗的影响 | K2 | | | X | X | |
| **3.4** | **AI 法规、标准与最佳实践框架** | | | | | | |
| GenAI-3.4.1 | 回顾软件测试中与生成式 AI 相关的 AI 法规、标准以及最佳实践框架示例 | K1 | | | X | X | X |
| **第 4 章** | **基于大语言模型（LLM）驱动的软件测试基础架构** | | | | | | |
| **4.1** | **基于大语言模型驱动的测试基础设施架构方法** | | | | | | |
| GenAI-4.1.1 | 解释基于大语言模型（LLM）驱动的测试基础设施的关键架构组件和概念 | K2 | | | | X | X |
| GenAI-4.1.2 | 总结检索增强生成技术 | K2 | | | | X | X |
| GenAI-4.1.3 | 解释基于大语言模型（LLM）驱动的智能体在自动化测试过程中的作用和应用场景 | K2 | | | | X | X |
| **4.2** | **微调与大语言模型运维（LLMOps）：生成式 AI 在软件测试中的实践** | | | | | | |
| GenAI-4.2.1 | 解释大语言模型针对特定测试任务的微调策略 | K2 | | | | X | X |
| Ge2AI-4.2.2 | 解释大语言模型运维（LLMOps）及其在测试任务中部署和管理大语言模型（LLM）所发挥的作用 | K2 | | | | X | X |
| **第 5 章** | **在测试组织开展生成式 AI 的部署与集成工作** | | | | | | |
| **5.1** | **在软件测试中采用生成式 AI 的路线图** | | | | | | |
| GenAI-5.1.1 | 回顾影子 AI 的风险 | K1 | | | | | X |
| GenAI-5.1.2 | 阐述制定软件测试生成式 AI 策略时需重点考量的关键要素 | K2 | | | | | X |
| GenAI-5.1.3 | 总结在特定场景下为软件测试任务选择大语言模型 / 小语言模型的关键标准 | K2 | | | | | X |

| 商业价值：使用生成式 AI 测试的认证测试工程师 | | | B01 | B02 | B03 | B04 | B05 |
|---|---|---|---|---|---|---|---|
| GenAI-5.1.4 | 回顾测试组织运用生成式 AI 的关键阶段 | K1 | | | | | X |
| **5.2** | **在软件测试中采用生成式 AI 时的变革管理** | | | | | | |
| GenAI-5.2.1 | 阐述测试人员在测试过程中有效运用生成式 AI 所需的核心技能与知识领域 | K2 | | | | | X |
| GenAI-5.2.2 | 回顾在测试团队中培养 AI 技能以支持在测试活动中采用生成式 AI 的策略 | K1 | | | | | X |
| GenAI-5.2.3 | 认识到测试组织在采用生成式 AI 时，测试过程和职责如何演变 | K1 | | | | | X |

# 9. 附录 C - 发布说明

本版本为 V1.0。首个版本暂无发布说明。

# 10. 附录 D - 生成式 AI 专用术语

| 术语名称 | 定义 |
| --- | --- |
| AI 聊天机器人 / AI chatbot | 一种对话式智能体，它利用大语言模型（LLMs）处理查询并生成类似人类的文本回复，从而实现与用户的交互式沟通。 |
| 上下文窗口 / Context window | 语言模型在生成响应时所考虑的文本范围，以词元（token）为单位进行衡量，该范围会影响模型输出内容的相关性与连贯性。 |
| 深度学习 / Deep learning | 使用多层神经网络的机器学习. |
| 嵌入 / Embedding | 一种用于将词元（token）表示为连续空间中稠密向量的技术。该技术在训练过程中学习，以捕捉语义、句法及上下文关系。 |
| 特征 / Feature | 输入数据中可供机器学习算法用于训练或机器学习模型用于预测的单个可度量属性。 |
| 少样本提示 / Few-shot prompting | 一种在提示中提供少量示例，以指导模型生成适当响应的技术。 |
| 微调 / Fine-tuning | 一种监督学习过程，它使用标注示例的数据集来更新大语言模型的权重，使其适应特定任务或领域。 |
| 基础大语言模型 / Foundation LLM | 在广泛文本数据上进行预训练的通用模型，能够基于学习到的语言模式预测下一个词。<br><br>同义词： 基础大语言模型（Base LLM） |
| 生成式人工智能（生成式 AI）/ Generative AI (GenAI) | 一类使用机器学习模型生成（新的）类似于人类创作内容的知识性内容的人工智能系统。 |
| 生成式预训练变换器（GPT）/ Generative pre-trained transformer (GPT) | 一种基于 Transformer 架构的深度学习模型，通过在海量文本数据上进行预训练，从而能够理解并生成类似人类语言的文本。 |

| | |
|---|---|
| 幻觉 / Hallucination | 由大语言模型生成的错误信息。 |
| 指令微调大语言模型/ Instruction-tuned LLM | 对基础大语言模型进行训练，使其能遵循各类指令。这一过程常借助反馈强化机制，以提升模型给出正确答案的能力。 |
| 大语言模型（LLM）/ Large language model (LLM) | 一种计算机程序，使用海量的语言数据集，以类似于人类的方式理解和生成文本。 |
| 大语言模型驱动的智能体 / LLM-powered agent | 一种集成大语言模型的推理、决策与记忆功能，并使用工具来执行任务的应用程序。 |
| 大语言模型运维 / LLMOps | 指一套专门针对大语言模型，聚焦于在生产环境中进行部署、监控与维护的实践方法及工具集合。此概念涵盖从模型上线到长期稳定运行过程中，确保大语言模型高效、可靠运作所涉及的各项技术与流程。 |
| 机器学习（ML） / machine learning (ML) | 指运用计算技术，使系统得以从数据或经验中学习的过程（ISO/IEC TR 29119 - 11） |
| 元提示 / Meta prompting | 一种更为高阶的指令编写技术，用于生成具备支持探索或自动化能力的特定提示词。 |
| 多模态模型 / Multimodal model | 指一类生成式 AI（GenAI）模型，该模型具备处理并生成多种数据类型（诸如文本、图像及音频等）内容的能力。 |
| 自然语言处理（NLP） / natural language processing (NLP) | 指计算机对以自然语言编码的数据进行处理，以此达成信息检索与知识表示的过程。 |
| 单样本提示 / One-shot prompting | 一种提示词编写技术，即在提示词中嵌入一个示例，以此引导大语言模型生成相应回复。 |
| 提示词 / Prompt | 在生成式 AI 及大语言模型范畴内，为引发特定回应所提供的自然语言输入内容。 |

| | |
|---|---|
| 提示词链 / Prompt chaining | 一种提示技术，此技术将某一提示词的输出作为另一提示词的输入，依此构建一系列连贯的提示词。 |
| 提示词工程 / Prompt engineering | 指旨在引导大语言模型生成预期输出，而对输入提示词进行设计与完善的过程。 |
| 推理型大语言模型 / Reasoning LLM | 一种依托指令微调模型构建而成的大语言模型，其核心在于提升模型模拟人类推理过程的能力。 |
| 检索增强生成（RAG）/ Retrieval-augmented generation (RAG) | 一种将大语言模型能力与检索工具相结合的技术。通过此技术，能够获取相关数据，从而生成准确且贴合上下文的回答。 |
| 影子 AI / Shadow AI | 指在组织内部，未获得正式批准或监管的情况下，使用生成式 AI 工具或系统的现象。 |
| 小语言模型（SLM） / Small language model (SML) | 指专门设计并训练的规模较小的语言模型，这类模型在效率与特定任务语言理解能力之间实现了平衡。 |
| 符号 AI / Symbolic AI | 一种借助符号、规则以及结构化知识对推理进行建模的人工智能方法。 |
| 系统提示词 / System prompt | 一套预先设定的指令集，通常对聊天机器人用户不可见。它持续为大语言模型的回复设定背景、确定语气、划定界限，并在整个交互过程中引导其行为。 |
| 温度参数 / Temperature | 一种控制大语言模型输出随机性或创造性的参数 。 |
| 词元化 / Tokenization | 将文本分解为较小单元以供语言模型处理的过程。 |
| Transformer 模型 / Transformer | 一种深度学习模型架构，它利用自注意力机制来捕获输入序列中的长程依赖关系。 |
| 用户提示词 / User prompt | 用户输入到大语言模型中的指令或查询，用于指导模型完成特定任务或提供所需的信息。 |

| | |
|---|---|
| 向量数据库 / Vector database | 一种经优化，专门用于存储和查询数据高维向量表示形式的数据库。 |
| 视觉语言模型 / Vision-language model | 一种生成式 AI 系统，该系统协同处理视觉与文本数据，借由跨模态的内容关联与生成来完成各项任务. |
| 零样本提示 / Zero-shot prompting | 一种提示词撰写技巧，所构造的提示词中不含任何示例，而是凭借模型已有的先验知识来生成回应。 |

# 11. 附录 E - 商标

ISTQB®是国际软件测试认证委员会的注册商标。