Certified Tester Specialist Level

Testing with Generative AI

CT-GenAI Syllabus

V1.1

International Software Testing Qualifications Board

Aviso de Direitos Autorais

Histórico de revisões

Versão	Data	Observações
V1.0	25/07/2025	CT-GenAI v1.0 Release
V1.1	27/04/2026	Minor changes

Versão na Língua Portuguesa

Data	Observações
29/07/2025	Lançamento da versão traduzida
18/08/2025	Correção das traduções de Machine Learning, Deep Learning, Prompting e Acurácia.
29/04/2026	Correções pontuais em Objetivos de Aprendizagem

Sumário

Agradecimentos

Este documento foi formalmente divulgado pela Assembleia Geral do ISTQB® em 25/07/2025.

Ele foi produzido por uma equipe do International Software Testing Qualifications Board: Abbas Ahmad (product owner), Gualtiero Bazzana, Alessandro Collino, Olivier Denoo, and Bruno Legeard (gerente técnico).

A equipe agradece a Anne Kramer, Jedrzej Kwapinski, Samuel Ouko e Ina Schieferdecker por sua revisão técnica, à equipe de revisão e aos Conselhos de Membros por suas sugestões e contribuições.

As seguintes pessoas participaram da revisão, dos comentários e da votação deste syllabus:

Albert Laura, Aneta Derkova, Anne Kramer, Arda Ender Torçuk, Baris Sarialioglu, Claire Van Der Meulen, Daniel van der Zwan, Dario Galiero, 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, Massimo Di carlo, Matthias Hamburg, Meile Posthuma, Michael Stahl, Márton Siska, Nele Van Asch, Nils Röttger, Nishan Portoyan, Nicola De Rosa, 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.

Agradecimentos do BSTQB

O BSTQB® agradece à equipe do **BSTQB WGT** (Grupo de Trabalho de Traduções do BSTQB) pelo empenho em traduzir este material. Atuaram na tradução e revisão: Denise Kanashiro, George Fialkovitz Jr., Irene Nagase, Osmar Higashi, Paula Oliveira F., Rogério Athaide de Almeida, Stênio Viveiros, Thiago Cesar Andrade.

O BSTQB® também agradece aos **ISTQB® Accredited Training Providers** no Brasil que contribuíram na revisão da tradução com sugestões e questionamentos. No momento deste trabalho os seguintes provedores estavam credenciados: Conecteseaqui, Exseed, Iterasys.

O BSTQB® também agradece às empresas brasileiras credenciadas no **ISTQB® Partner Program** que contribuíram na revisão da tradução com sugestões e questionamentos. No momento deste trabalho as seguintes empresas estavam credenciadas: Cast, Deltapoint, Keeggo, Venturus.

0 Introdução

0.1 Objetivo deste syllabus

Este syllabus forma a base para a certificação do International Software Testing Qualifications Board for Testing with Generative AI (CT-GenAI). O ISTQB® fornece este syllabus da seguinte forma:

1. Aos conselhos membros, para traduzir para seu idioma local e credenciar provedores de treinamento. Os conselhos membros podem adaptar o syllabus às suas necessidades linguísticas específicas e modificar as referências para adaptá-las às suas publicações locais.
2. Aos órgãos de certificação, para que elaborem questões de exame em seu idioma local, adaptadas aos objetivos de aprendizagem deste syllabus.
3. Aos provedores de treinamento, para que produzam material didático e determinem os métodos de ensino adequados.
4. Aos candidatos à certificação, para se prepararem para o exame de certificação (como parte de um curso de treinamento ou de forma independente).
5. Para a comunidade internacional de engenharia de software e sistemas, para promover a profissão de teste de software e sistemas, e como base para livros e artigos.

0.2 Teste de software com AI generativa

A certificação Testing with Generative AI destina-se a qualquer pessoa envolvida no uso de AI generativa (GenAI) para testes de software. Isso inclui pessoas em funções como testadores, analistas de teste, engenheiros de automação de teste, gerentes de teste, testadores de aceite do usuário e desenvolvedores de software. A qualificação Testing with GenAI também é adequada para qualquer pessoa que queira ter uma compreensão básica do uso da GenAI para testes de software, como gerentes de projeto, gerentes de qualidade, gerentes de desenvolvimento de software, analistas de negócios, diretores de TI e consultores de gerenciamento.

0.3 Carreira para testadores

O programa ISTQB® oferece suporte a profissionais de teste em todos os estágios de suas carreiras, oferecendo amplitude e profundidade de conhecimento. Os indivíduos que obtiverem a certificação ISTQB® Certified Tester Testing with Generative AI também podem se interessar pelos níveis Core Advanced (Test Analyst, Technical Test Analyst, Test Manager, e Test Engineering) e, posteriormente, pelo nível Expert (Test Management ou Improving the Test Process). Acesse www.istqb.org para obter as informações mais recentes sobre o Esquema de Certificações em Teste do ISTQB®.

0.4 Resultados de negócio

Esta seção lista os resultados de negócio esperados de um candidato que tenha obtido a certificação Testing with Generative AI.

Um candidato que tenha obtido a certificação Testing with Generative AI pode:

GenAI -BO1	Compreender os conceitos fundamentais, os recursos e as limitações da AI generativa
GenAI -BO2	Desenvolver habilidades práticas para solicitar grandes modelos de linguagem para testes de software
GenAI -BO3	Obter informações sobre os riscos e as atenuações do uso de AI generativa para testes de software
GenAI -BO4	Obtenha insights sobre os aplicativos de soluções generativas de AI para testes de software
GenAI -BO5	Contribuir efetivamente para a definição e a implementação de uma estratégia e um roteiro de AI generativa para testes de software em uma organização

0.5 Objetivos de aprendizagem examináveis, objetivos práticos e nível cognitivo de conhecimento

Os objetivos de aprendizagem e práticos apoiam os resultados de negócio e são usados para criar exames de certificação Testing with Generative AI.

Em geral, todo o conteúdo deste syllabus é passível de exame nos níveis K1, K2 e K3, exceto a Introdução, os Objetivos Práticos e os Apêndices. As perguntas do exame confirmarão o conhecimento das palavras-chave no nível K1 (veja abaixo) ou dos objetivos de aprendizagem em todos os níveis K.

Os níveis específicos dos objetivos de aprendizagem são mostrados no início de cada capítulo, e classificados da seguinte forma:

- K1: Lembrar
- K2: Compreender
- K3: Aplicar

Mais detalhes e exemplos de objetivos de aprendizagem são fornecidos no Apêndice A.

Todos os termos listados como palavras-chave logo abaixo dos títulos dos capítulos devem ser lembrados, mesmo que não sejam explicitamente mencionados nos objetivos de aprendizagem.

Os objetivos práticos específicos (HO) são mostrados no início de cada capítulo. Cada HO está vinculado a um OA no nível K2 ou K3, com o objetivo de refinar o aprendizado por meio da prática. O nível de um HO é classificado da seguinte forma:

- H0: Pode incluir uma demonstração ao vivo de um exercício ou um vídeo gravado. Como isso não é realizado pelo trainee, não é estritamente um exercício.
- H1: Exercício guiado. Os trainees seguem uma sequência de etapas executadas pelo instrutor.
- H2: Exercício com dicas. O trainee recebe um exercício com dicas relevantes para permitir que o exercício seja resolvido dentro do prazo determinado.

0.6 O exame de certificação Certified Tester Testing with Generative AI

O exame do certificado Certified Tester Testing with Generative AI será baseado neste syllabus. As respostas às perguntas do exame podem exigir o uso de material baseado em mais de uma seção deste syllabus. Todas as seções do syllabus são passíveis de exame, exceto a Introdução, os objetivos práticos e os Apêndices. Normas, livros e artigos são incluídos como referências, mas seu conteúdo não é passível de exame, além do que está resumido no próprio syllabus.

Consulte o documento Exam Structures and Rules V1.0 do Certified Tester Testing with Generative AI para obter mais detalhes.

Observação sobre requisitos de entrada: O certificado ISTQB® Foundation Level deve ser obtido antes de fazer o exame de certificação ISTQB® Certified Tester Testing with Generative AI.

0.7 Credenciamento

Um Conselho de Membros do ISTQB® pode credenciar provedores de treinamento cujo material do curso segue este syllabus. Os provedores de treinamento devem obter diretrizes de credenciamento do Conselho de Membros ou do órgão que realiza o credenciamento. Um curso credenciado é reconhecido como estando em conformidade com este syllabus e pode ter um exame ISTQB® como parte do curso.

As diretrizes de credenciamento para este syllabus estão definidas no documento ISTQB CT-GenAI Accreditation Guidelines.

0.8 Manuseio de padrões

Existem normas associadas às características de qualidade e aos testes de software, ou seja, aquelas referenciadas no syllabus do Nível Básico, como as do IEEE e da ISO. O objetivo dessas referências é fornecer uma estrutura ou uma fonte de informações adicionais, se o leitor desejar. Observe que os syllabi estão usando os documentos padrão como referência. Os documentos de normas não se destinam a exame. Consulte o Capítulo 6 para obter mais informações sobre as normas.

0.9 Nível de detalhamento

O nível de detalhes deste syllabus permite cursos e exames consistentes em nível internacional. Para atingir essa meta, o syllabus consiste em:

- Objetivos gerais de instrução que descrevem a intenção da certificação ISTQB® Certified Tester Testing with Generative AI.
- Uma lista de termos que os alunos devem ser capazes de lembrar.
- Objetivos de aprendizagem para cada área de conhecimento, descrevendo o resultado cognitivo de aprendizagem a ser alcançado.
- Uma descrição dos principais conceitos, incluindo referências a fontes como literatura ou padrões aceitos.
- Uma descrição para cada objetivo prático da prática recomendada para apoiar o aprendizado.

O conteúdo do syllabus não é uma descrição de toda a área de conhecimento de testes com GenAI; ele reflete o nível de detalhes a ser abordado nos cursos de treinamento ISTQB® Certified Tester Testing with Generative AI. Ele se concentra em conceitos e técnicas de teste que podem ser aplicados a todos os projetos de software ao usar a AI generativa para testes.

O syllabus usa a terminologia (ou seja, o nome e o significado) dos termos usados em testes de software e garantia de qualidade de acordo com o Glossário ISTQB®.

0.10 Como este syllabus está organizado

Há 5 capítulos com conteúdo passível de exame. O título de nível superior de cada capítulo especifica o tempo para o capítulo; o tempo não é fornecido abaixo do nível do capítulo. Para cursos de treinamento credenciados, o syllabus exige um mínimo de 13,6 horas de instrução, distribuídas pelos 5 capítulos da seguinte forma:

- Capítulo 1: 100 minutos - Introdução à AI Generativa para teste de software
  - Aprender os conceitos básicos de Large Language Models (LLMs), incluindo tokenização e recursos multimodais.
  - Explorar as aplicações da AI generativa (GenAI) em testes de software, distinguindo o chatbot de AI das ferramentas de teste com LLM e experimentando a tokenização, as janelas de contexto e os prompts multimodais.
- Capítulo 2: 365 minutos - Engenharia de prompts para Testes de Software Eficientes
  - Aprender a criar prompts eficazes e estruturados para a GenAI em testes de software.
  - Adquirir experiência prática com técnicas de engenharia de prompts para tarefas de teste de software e as aplica.
- Capítulo 3: 160 minutos - Gerenciando riscos de AI Generativa em Testes de Software
  - Aprender a identificar e mitigar alucinações, erros de raciocínio e vieses ao testar com GenAI.
  - Aprender a lidar com questões de privacidade de dados e segurança da GenAI em testes de software.
  - Aprender sobre o consumo de energia e o impacto ambiental da GenAI em testes de software.
  - Aprender os regulamentos, os padrões e as práticas recomendadas de AI para o uso ético, transparente e seguro da GenAI em testes de software.

- Capítulo 4: 110 minutos - Infraestrutura de teste com tecnologia LLM para Teste de Software
  - Explorar a arquitetura da GenAI como a Geração Aumentada por Recuperação e os agentes da GenAI.
  - Aprender o processo de ajuste fino dos LLMs para tarefas de teste de software.
  - Aprender os conceitos do Large Language Model Operations (LLMOps) para implantar e gerenciar LLMs em testes de software.
- Capítulo 5: 80 minutos - Implementação e integração da AI Generativa em Organizações de Teste
  - Aprender um roteiro estruturado para integrar a GenAI aos processos de teste.
  - Aprender a transformação organizacional para a integração da GenAI nos processos de teste.

1 Introdução à AI Generativa para Teste de Software - 100 min

Palavras-chave

Nenhuma

Palavras-chave específicas da AI generativa

AI chatbot, janela de contexto, deep learning, incorporação, recurso, fundação LLM, AI generativa, transformador generativo pré-treinado, LLM ajustado por instrução, modelo de linguagem grande, machine learning de máquina, modelo multimodal, raciocínio LLM, AI simbólica, tokenização, transformador

Objetivos de aprendizagem e objetivos práticos para o Capítulo 1:

1.1 Fundamentos e conceitos-chave da AI generativa

GenAI-1.1.1	(K1)	Relembrar os diferentes tipos de IA: AI simbólica, machine learning clássico, deep learning e AI generativa
GenAI-1.1.2	(K2)	Explicar os conceitos básicos de AI generativa e Large Language Models
HO-1.1.2	(H1)	Praticar a tokenização e a avaliação da contagem de tokens ao usar um LLM para uma tarefa de teste de software
GenAI-1.1.3	(K2)	Distinguir entre fundação, ajuste de instrução e raciocínio LLMs
GenAI-1.1.4	(K2)	Resumir os princípios básicos de LLMs multimodais e modelos de visão-linguagem
HO-1.1.4	(H1)	Escrever e executar um prompt para um LLM multimodal utilizando entradas tanto textuais quanto de imagens para uma tarefa de teste de software

1.2 Aproveitamento da AI generativa em testes de software: Princípios básicos

GenAI-1.2.1	(K2)	Dar exemplos dos principais recursos do LLM para tarefas de teste
GenAI-1.2.2	(K2)	Comparar modelos de interação ao usar o GenAI para testes de software

1.1 Fundamentos e conceitos-chave da AI generativa

A Inteligência Artificial Generativa (GenAI) é um ramo da inteligência artificial que usa modelos grandes e pré-treinados para gerar resultados semelhantes aos humanos, como texto, imagens ou código. Os Large Language Model (LLMs) são modelos de GenAI pré-treinados em grandes conjuntos de dados textuais, o que lhes permite determinar o contexto e produzir respostas relevantes de acordo com as solicitações do usuário.

Os principais conceitos incluem tokenização (ou seja, dividir o texto em unidades para um processamento eficiente), janelas de contexto (limitar a quantidade de informações consideradas de uma só vez para manter a relevância) e modelos multimodais (capazes de processar vários tipos de dados, como texto, imagens e áudio, para interações ricas).

Nos testes de software, esses LLMs podem dar suporte a tarefas como a revisão e o aprimoramento dos critérios de aceite, a geração de casos de teste ou scripts de teste, a identificação de possíveis defeitos, a análise de padrões de defeitos, a geração de dados de teste sintéticos ou a geração de documentação de suporte em todo o processo de teste.

1.1.1 Espectro de AI: AI simbólica, machine learning clássico, deep learning e AI generativa

A Inteligência Artificial (AI) é um campo amplo que engloba diferentes tipos de tecnologias, cada uma com sua maneira exclusiva de resolver problemas, como a AI simbólica, o machine learning clássico, o deep learning e a AI GenAI (entre outras tecnologias que estão fora do escopo deste syllabus):

- A AI simbólica usa um sistema baseado em regras para imitar a tomada de decisão humana. Essencialmente, a AI simbólica representa o conhecimento usando símbolos e regras lógicas.
- O Machine Learning (aprendizado de máquina) clássico é uma abordagem orientada por dados que requer preparação de dados, seleção de recursos e treinamento de modelos, e pode ser usado para tarefas como categorização de defeitos e previsão de problemas de software.
- A Deep Learning usa estruturas de aprendizagem de máquina chamadas redes neurais para aprender automaticamente os recursos dos dados. Os modelos de deep learning podem encontrar padrões em conjuntos de dados muito grandes e complexos, como imagens, vídeo, áudio ou texto, sem a necessidade de os usuários definirem manualmente os recursos, embora, na prática, isso ainda possa exigir o envolvimento humano em tarefas como anotação de dados, ajuste de modelos ou validação de resultados.
- A AI generativa usa técnicas de deep learning para criar conteúdo (texto, imagens, código) aprendendo e imitando padrões de seus dados de treinamento. Modelos como os LLMs podem gerar texto, escrever código e simular o raciocínio ou a solução de problemas dentro do escopo de seu treinamento.

Em resumo, o campo da AI evoluiu em várias direções, cada uma com diferentes pontos fortes e limitações. A principal vantagem de usar a GenAI para testes de software é que ela usa modelos pré-treinados que podem ser aplicados diretamente a tarefas de teste sem a necessidade de uma fase de treinamento adicional, embora isso tenha alguns riscos (consulte a Seção 3.1).

1.1.2 Noções básicas de AI generativa e LLMs

Com base no transformador generativo pré-treinado modelo de deep learning, os LLMs são treinados em conjuntos de dados muito grandes, incluindo livros, artigos e sites. Os modelos de linguagem pequenos (SLMs) são modelos compactos com menos parâmetros em comparação com os Large Language Models, projetados para fornecer soluções GenAI leves e focadas.

Os LLMs podem lidar com as nuances do idioma e gerar conteúdo coerente. Dois conceitos fundamentais que ajudam os LLMs a processar e gerar conteúdo são a tokenização e os embeddings. A tokenização e os embeddings convertem a linguagem em uma forma numérica que o modelo pode processar com eficácia.

- A tokenização em modelos de linguagem é o processo de dividir o texto em unidades menores chamadas tokens. Os tokens podem ser tão pequenos quanto um caractere ou tão grandes quanto uma subpalavra ou palavra. Quando um LLM processa uma frase, ele primeiro tokeniza a entrada para que cada token possa ser entendido individualmente, mantendo o contexto geral.
- Embeddings são representações numéricas de tokens que codificam suas relações semânticas, sintáticas e contextuais em um formato adequado para processamento por modelos de AI generativos. Cada token é

transformado em um vetor em um espaço de alta dimensão, capturando informações diferenciadas sobre seu significado e uso. Os tokens com significados ou funções contextuais semelhantes têm embeddings que estão posicionados próximos uns dos outros nesse espaço. Essa proximidade permite que os LLMs entendam as relações entre as palavras, retenham o contexto e gerem respostas coerentes e contextualmente apropriadas.

Os LLMs utilizam uma arquitetura de rede neural conhecida como modelo transformador. Os modelos transformadores são excelentes em tarefas de linguagem, processando o contexto de sequências extensas de texto e aprendendo como os tokens se relacionam entre si. Durante a inferência, os LLMs preveem o próximo token em uma sequência, aproveitando essas relações aprendidas para gerar um texto coerente e contextualmente apropriado. O modelo do transformador pode ser usado para gerar um novo texto que seja estatisticamente plausível, com base nos dados de treinamento e no prompt. Mas plausível não é necessariamente correto.

Os LLMs apresentam comportamento não determinístico, principalmente devido à natureza probabilística de seus mecanismos de inferência e configurações de hiper parâmetros. Essa aleatoriedade inerente pode levar a variações nos resultados, mesmo quando a mesma entrada é fornecida várias vezes.

No âmbito dos LLMs, a janela de contexto refere-se à quantidade de texto anterior, medida em tokens, que o modelo pode considerar ao gerar respostas. Uma janela de contexto maior permite que o modelo mantenha a coerência em passagens mais longas, por exemplo, ao analisar grandes registros de testes. No entanto, aumentar o número de tokens na janela de contexto também aumenta a complexidade computacional e o tempo de processamento necessários para que o modelo funcione de forma eficaz.

Objetivo prático 1.1.2 (H1): Praticar a tokenização e a avaliação da contagem de tokens

Esta atividade prática foi criada para ajudar os participantes a desenvolverem uma compreensão prática da tokenização e suas implicações ao trabalhar com LLMs. O exercício é dividido em duas partes principais:

- Tokenização: Use um tokenizador para dividir um texto de amostra em tokens individuais. Examine o resultado para ver como as palavras, a pontuação e as frases são representadas e identifique padrões ou nuances na tokenização.
- Avaliação da contagem de tokens: Meça o número de tokens gerados a partir de vários textos de entrada. Analise como a contagem de tokens influencia o desempenho do modelo, especialmente em relação aos limites da janela de contexto do modelo e às considerações de eficiência.

Ao final deste exercício, os alunos poderão prever melhor como diferentes estruturas de texto e comprimentos de entrada podem afetar as interações com os LLMs.

1.1.3 LLMs básicos, ajustados por instrução e de raciocínio

Os Large Language Models são desenvolvidos por meio de estágios de treinamento progressivamente especializados para aumentar sua eficácia em uma ampla gama de tarefas. Esses estágios dão origem a três categorias principais: LLMs básicos, LLMs ajustados por instrução e LLMs de raciocínio.

- LLMs básicos: São modelos de uso geral treinados em conjuntos de dados vastos e diversos que incluem texto, código, imagens e outras modalidades. Seu extenso pré-treinamento permite que eles suportem várias tarefas em domínios como processamento de linguagem natural, visão computacional e reconhecimento de fala. Embora sejam eficientes e flexíveis, os modelos básicos geralmente exigem mais adaptações para atender aos requisitos de tarefas específicas.
- LLMs ajustados por instrução: Derivados dos modelos básicos, os LLMs ajustados por instruções são aperfeiçoados com o uso de conjuntos de dados que combinam solicitações com respostas esperadas. Esse estágio aprimora seu alinhamento com as instruções humanas, melhorando a usabilidade em aplicativos do mundo real. O processo de ajuste envolve a otimização da aderência à tarefa, o seguimento da instrução e a coerência da resposta, melhorando, assim, a capacidade do modelo de interpretar e agir de forma eficaz com base na intenção do usuário.

- LLMs de raciocínio: Os modelos de raciocínio ampliam os modelos ajustados por instruções, enfatizando as habilidades cognitivas estruturadas, como inferência lógica, solução de problemas em várias etapas e raciocínio em cadeia. Esses modelos são ainda mais treinados ou aperfeiçoados em tarefas cuidadosamente selecionadas que exigem compreensão contextual, etapas intermediárias de raciocínio e síntese de informações complexas. Como resultado, eles são mais adequados para tarefas de alta carga cognitiva, incluindo aquelas em domínios técnicos.

No contexto dos aplicativos GenAI para teste de software, são utilizados LLMs ajustados por instrução (às vezes chamados de sem raciocínio) e de raciocínio. A seleção depende da complexidade e das demandas de raciocínio da tarefa de teste específica em questão.

1.1.4 LLMs multimodais e modelos de linguagem visual

Os LLMs multimodais estendem o modelo tradicional de transformador para processar várias modalidades de dados, incluindo texto, imagens, som e vídeo. Esses modelos são treinados em conjuntos de dados grandes e diversificados que lhes permitem aprender as relações entre diferentes tipos de dados. Para lidar com várias modalidades, a tokenização é adaptada para cada tipo de dado - por exemplo, as imagens são convertidas em embeddings usando modelos de linguagem visual antes de serem processadas no modelo transformador.

Os modelos de linguagem visual, um subconjunto de LLMs multimodais, integram especificamente informações visuais e textuais para executar tarefas como legendas de imagens, respostas a perguntas visuais e análise da consistência entre entradas textuais e visuais.

Nos testes de software, os LLMs multimodais, especialmente os LLMs incrementados com modelos de linguagem e visual, oferecem oportunidades significativas. Eles podem analisar elementos visuais de aplicativos, como capturas de tela e wireframes de GUI, juntamente com descrições textuais associadas, como relatórios de defeitos ou histórias de usuários. Esse recurso permite que os testadores identifiquem discrepâncias entre os resultados esperados e os elementos visuais reais em uma captura de tela. Além disso, os LLMs aumentados com modelos de linguagem visual podem gerar casos de teste ricos e realistas que incorporam dados textuais e dicas visuais, aumentando assim a cobertura geral.

Objetivo prático HO-1.1.4 (H1): Analisar e executar um determinado prompt abordando uma tarefa de teste usando um modelo multimodal LLM

Este exercício envolve a revisão e a execução de um determinado prompt para um LLM multimodal usando entradas de texto e imagem para resolver uma tarefa de teste em duas etapas:

- Analisar as entradas: Examine o prompt e os dados de entrada (texto e imagem).
- Execute o prompt e verifique o resultado: Use um LLM multimodal para inserir imagem e texto e verifique a resposta do LLM.

Este exercício demonstra como usar LLMs multimodais para uma tarefa que envolve a entrada de texto e imagem em casos de uso de teste de software, incluindo o reconhecimento dos benefícios e dos possíveis desafios envolvidos.

1.2 Aproveitamento da AI generativa em testes de software: Princípios básicos

A GenAI oferece recursos transformadores em várias atividades de teste. Os LLMs são excelentes no processamento de linguagem natural e código, gerando texto e código coerentes, respondendo a perguntas, resumindo informações, traduzindo idiomas e analisando imagens em um contexto multimodal.

Os profissionais de teste em todas as funções podem aproveitar o GenAI de duas maneiras complementares: por meio de chatbots do GenAI que fornecem respostas instantâneas a consultas e por meio de aplicativos com tecnologia LLM integrados a ferramentas de teste.

1.2.1 Principais recursos do LLM para tarefas de teste

Os LLMs podem interpretar requisitos, especificações, capturas de tela, código, casos de teste e relatórios de defeitos, tornando-os ferramentas para compreender e esclarecer as informações necessárias durante todo o processo de teste e gerar elementos do testware. Veja a seguir alguns dos principais recursos do LLM relevantes para o teste de software:

- Análise e aprimoramento de requisitos: Os LLMs podem ajudar a analisar os requisitos e outros elementos da base de teste, identificando ambiguidades, inconsistências ou informações ausentes. Eles podem gerar perguntas significativas para ajudar a esclarecer os requisitos durante as discussões com os stakeholders.
- Suporte à criação de casos de teste: Os LLMs podem ajudar a gerar casos de teste e sugerir objetivos de teste com base nos requisitos do sistema, histórias de usuários ou quaisquer outros elementos da base de teste.
- Geração de oráculos de teste: Os LLMs podem ajudar a gerar os resultados esperados.
- Geração de dados de teste: Os LLMs podem gerar conjuntos de dados, definir valores limite e criar diferentes combinações de dados de teste.
- Suporte à automação de testes: Os LLMs podem ajudar a gerar scripts de teste a partir da descrição do caso de teste e melhorar os scripts de teste existentes, sugerindo alterações e identificando técnicas apropriadas de projeto de teste.
- Análise de resultados de testes: Os LLMs podem ajudar a analisar os resultados dos testes, criando resumos e classificando anomalias com base na gravidade e na prioridade.
- Criação de testware: Os LLMs podem ajudar a criar vários documentos, incluindo planos de teste, relatórios de teste e relatórios de defeitos, e mantê-los atualizados à medida que o projeto evolui.

Esses recursos demonstram como os LLMs podem afetar vários aspectos dos testes de software durante todo o processo de teste.

1.2.2 Chatbots com AI e aplicativos de teste com tecnologia LLM para teste de software

Os chatbots com AI e os aplicativos de teste com LLM podem ajudar os testadores, embora sejam diferentes em termos de funcionalidade, flexibilidade e abordagens de integração.

Os chatbots com AI oferecem uma interface de conversação fácil de usar que permite que os testadores se comuniquem diretamente com os LLMs. Essa interação de linguagem natural permite que os testadores insiram perguntas, comandos ou solicitações e recebam respostas imediatas e contextualizadas. Por meio de técnicas como o encadeamento de prompts, os testadores podem refinar os resultados de forma iterativa, tornando os chatbots particularmente eficazes para tarefas de rotina, testes exploratórios e até mesmo para a integração de novos testadores, fornecendo acesso rápido a conhecimentos e práticas de teste.

Esses chatbots de AI são especialmente úteis em cenários que exigem feedback rápido, esclarecimento de conceitos de teste ou exploração dinâmica de requisitos e possíveis casos de teste. Sua interface intuitiva os torna acessíveis até mesmo para os stakeholders não técnicos, ampliando a base de usuários em potencial e incentivando uma adoção mais ampla.

Os aplicativos de teste com LLM, por outro lado, envolvem a integração de recursos de LLM por meio de APIs para executar tarefas de teste bem definidas e, muitas vezes, automatizadas. Esses aplicativos oferecem maior personalização e escalabilidade, permitindo que as organizações e os fornecedores de ferramentas incorporem a AI generativa às estruturas de teste existentes. Isso permite a automação de tarefas repetitivas ou complexas, como geração de casos de teste, análise de defeitos ou síntese de dados de teste. Em implementações mais avançadas, as organizações podem criar agentes de AI projetados especificamente para desempenhar determinadas funções de teste (consulte o Capítulo 4).

Independentemente da forma como o testador interage com os LLMs - seja por meio de chatbots ou aplicativos integrados com LLMs -, a implementação bem-sucedida da AI generativa em testes requer uma sólida engenharia de prompts (consulte o Capítulo 2). Prompts cuidadosamente projetados e instruções claras e específicas são essenciais para garantir que os resultados gerados pelo LLM sejam precisos, relevantes e alinhados aos objetivos do teste. Essa prática ajuda o site a maximizar o valor derivado da AI generativa e garante suporte consistente e confiável para uma ampla gama de atividades de teste.

2 Engenharia de Prompts para Testes de Software Eficientes - 365 min.

Palavras-chave

critérios de aceite, script de teste, caso de teste, condição de teste, dados de teste, projeto de teste, relatório de teste

Palavras-chave específicas de AI generativa

prompting de poucos disparos, meta prompting, processamento de linguagem natural, prompting de um disparo, prompt, encadeamento de prompts, engenharia de prompts, prompt do sistema, prompt do usuário, prompting de zero disparo.

Objetivos de aprendizagem e objetivos práticos do Capítulo 2:

2.1 Desenvolvimento eficaz de prompts

GenAI -2.1.1	(K2)	Dar exemplos da estrutura dos prompts usados na AI generativa para teste de software
HO-2.1.1	(H0)	Observar vários prompts dados para tarefas de teste de software, identificando os componentes de função, contexto, instrução, dados de entrada, restrições e formato de saída em cada um deles.
GenAI -2.1.2	(K2)	Diferenciar as principais técnicas de prompting para testes de software
HO-2.1.2**a**	(H0)	Observar demonstrações do encadeamento de prompts, prompts de poucos disparos e meta prompting aplicados a tarefas de teste de software
HO-2.1.2b	(H1)	Identificar quais técnicas de engenharia de prompts estão sendo utilizadas nos exemplos apresentados
GenAI -2.1.3	(K2)	Distinguir entre prompts de sistema e prompts de usuário

2.2 Aplicação de técnicas de engenharia de prompts a tarefas de teste de software

GenAI- 2.2.1	(K3)	Aplicar a AI generativa para testar tarefas de análise
HO-2.2.1a	(H2)	Praticar o prompting multimodal para gerar critérios de aceite para uma história de usuário com base em um wireframe de GUI
HO-2.2.1b	(H2)	Praticar o encadeamento do prompt e a verificação humana para analisar progressivamente uma determinada história de usuário e refinar os critérios de aceite
GenAI- 2.2.2	(K3)	Aplicar a AI generativa para testar o projeto e testar as tarefas de implementação
HO-2.2.2a	(H2)	Praticar a geração de casos de teste funcionais a partir de histórias de usuários com AI usando encadeamento de prompts, prompts estruturados e meta prompting
HO-2.2.2b	(H2)	Usar a técnica de prompting de poucos disparos para gerar condições de teste no estilo Gherkin e casos de teste a partir de histórias de usuários
HO-2.2.2c	(H2)	Usar o encadeamento de solicitações para priorizar os casos de teste em um determinado conjunto de testes, levando em conta suas prioridades e dependências específicas
GenAI -2.2.3	(K3):	Aplicar AI generativa a testes de regressão automatizados

HO-2.2.3a	(H2)	Praticar a prompting de poucos disparos para criar e gerenciar scripts de teste orientados por palavras-chave
HO-2.2.3b	(H2)	Praticar a engenharia de prompt estruturado para análise de relatório de teste
GenAI -2.2.4	(K3)	Observar as métricas de monitoramento de teste elaboradas por AI generativa a partir dos dados de teste
HO-2.2.4	(H0)	Observar métricas de monitoramento de teste preparadas pela AI a partir de dados de teste
GenAI -2.2.5	(K3)	Selecionar e aplicar técnicas de prompting apropriadas para um determinado contexto e tarefa de teste
HO-2.2.5	(H1)	Selecionar e aplicar técnicas de estímulo adequadas ao contexto para uma determinada tarefa de teste

2.3 Avaliar os resultados da AI generativa e refinar as instruções para as tarefas de teste de software

GenAI- 2.3.1	(K2)	Compreender as métricas para avaliar os resultados da AI generativa em tarefas de teste
HO-2.3.1	(H0)	Observar como as métricas podem ser usadas para avaliar o resultado da AI generativa em uma tarefa de teste
GenAI- 2.3.2	(K2)	Dar exemplos de técnicas para avaliar e refinar iterativamente os prompts
HO-2.3.2	(H1)	Avaliar e otimizar um prompt para uma determinada tarefa de teste

2.1 Desenvolvimento eficaz de prompts

O design eficaz do prompt garante que as ferramentas do GenAI executem tarefas de teste de software com precisão e eficiência e que os testadores obtenham resultados úteis do chatbot. Um prompt estruturado inclui diferentes componentes (consulte a seção 2.1.1). Cada um desses componentes contribui para a clareza e a precisão de um prompt que comunica efetivamente os requisitos e as expectativas aos LLMs.

Várias técnicas de engenharia de prompt aumentam a eficácia dos prompts nos testes de software. Técnicas como encadeamento de prompts, prompts de poucos disparos e meta prompting ajudam a enfrentar desafios complexos de teste (consulte a seção 2.1.2).

A combinação de prompts estruturados (consulte a seção 2.1.1) com as principais técnicas de prompt tem como objetivo obter bons resultados ao consultar um LLM para tarefas de teste de software (consulte a seção 2.1.3).

2.1.1 Estrutura de prompts para AI generativa em testes de software

Um prompt estruturado para teste de software normalmente inclui seis componentes:

- Função: A função define a perspectiva ou a persona que o modelo GenAI deve adotar ao gerar uma resposta. A especificação da função ajuda o LLM a determinar suas responsabilidades e adotar um tom ou uma abordagem apropriada, como atuar como testador, gerente de testes ou engenheiro de automação de testes.
- Contexto: O contexto fornece as informações básicas de que o modelo GenAI precisa para determinar as condições de teste. Isso inclui detalhes sobre o objeto de teste, a funcionalidade específica a ser testada e qualquer informação contextual relevante.
- Instrução: As instruções são diretrizes dadas ao GenAI que descrevem a tarefa específica a ser executada. Instruções claras, imperativas e concisas incluem uma descrição da tarefa e todos os requisitos relevantes para a tarefa.
- Dados de entrada: Os dados de entrada incluem qualquer informação necessária para executar a tarefa, como histórias de usuários, critérios de aceite, capturas de tela, código, casos de teste existentes ou exemplos de resultados. O fornecimento de dados de entrada detalhados e estruturados ajuda o LLM a gerar resultados mais precisos e sensíveis ao contexto.
- Restrições: As restrições descrevem quaisquer restrições ou considerações especiais às quais o LLM deve aderir. As restrições ajudam a especificar como as instruções devem ser aplicadas aos dados de entrada.
- Formato de saída: As especificações de saída indicam o formato, a estrutura ou as características esperadas da resposta. Esses indicadores ajudam a moldar o resultado do LLM.

Esses componentes formam a estrutura básica do prompt. Essa estrutura deve ser combinada com a implementação de técnicas de prompting (consulte a Seção 2.1.2), dependendo da tarefa a ser executada e do LLM a ser usado.

Objetivo prático HO-2.1.1 (H0): Observar e analisar os componentes do prompt

Em uma demonstração, vários prompts estruturados são experimentados em um chatbot de IA, cada um adaptado a tarefas específicas de teste de software. Esses avisos seguem um formato estruturado que consiste em seis componentes principais: função, contexto, instrução, dados de entrada, restrições e formato de saída. A demonstração tem como objetivo facilitar a observação e a análise desses prompts estruturados, destacando como cada componente contribui para fornecer insights precisos, relevantes e acionáveis a um LLM usado para uma tarefa de teste de software.

2.1.2 Principais técnicas de prompting para teste de software

Nos últimos anos, muitas técnicas de prompting de LLM foram propostas para diferentes casos de uso da GenAI (Schulhoff 2024). Entre elas, três técnicas principais de prompting são comumente usadas para tarefas de teste com o GenAI em conjunto com a estrutura de prompting de 6 componentes descrita acima (consulte a seção 2.1.1): encadeamento de prompt, prompting de poucos disparos e meta prompting.

- O encadeamento de prompts envolve a divisão de uma tarefa em uma série de etapas intermediárias (vários prompts). O resultado de cada etapa é verificado e refinado manual ou automaticamente antes de prosseguir para a próxima etapa. Essa abordagem leva a uma maior acurácia, pois cada resposta informa o próximo prompt. O encadeamento de prompts é particularmente útil em processos de teste em que as tarefas são complicadas e exigem decomposição em subtarefas e verificação sistemática dos resultados intermediários do LLM. Ela também permite interações dinâmicas nos processos de teste.
- O prompting de poucos disparos envolve o fornecimento de exemplos ao LLM no prompt. Enquanto o prompt de zero disparo (sem exemplo) depende do conhecimento pré-existente do modelo para gerar uma resposta, o prompt de disparo único fornece um exemplo para demonstrar o resultado desejado para uma determinada entrada. Os prompts de poucos disparos contêm mais de um exemplo (alguns) para consolidar ainda mais o comportamento de resposta desejado do modelo.

  Essa técnica ajuda a orientar o modelo, fornecendo uma referência clara e garantindo que os resultados sejam consistentes e estejam de acordo com as expectativas. O prompting de poucos disparos é particularmente eficaz para tarefas em que os exemplos podem ilustrar o comportamento necessário, permitindo que o modelo generalize de forma eficaz e produza resultados confiáveis.
- O meta prompting aproveita a capacidade da AI de gerar ou refinar seus próprios prompts. Em um ciclo iterativo, o LLM pode gerar prompts que podem ser avaliados e refinados pelo testador. Essa abordagem otimiza a qualidade do prompt ao aproveitar o conhecimento dos LLMs sobre prompts otimizados. O meta prompting é especialmente vantajoso quando a eficiência e a otimização do prompt são fundamentais, pois reduz o esforço manual necessário para projetar prompts eficazes. Outra vantagem do meta prompting é que, se o testador não tiver certeza de como criar um prompt eficaz, ele poderá colaborar com o LLM para criá-lo em conjunto. Isso reflete uma forma de emparelhamento com a ferramenta GenAI, em que o testador e a AI trabalham juntos de forma interativa para atingir um objetivo compartilhado. Esse conceito de emparelhamento destaca uma nova maneira de colaborar com as ferramentas de IA, aprimorando a produtividade e o aprendizado, não apenas na engenharia de prompt, mas também na programação e nos testes em pares.

Essas técnicas de prompting podem ser usadas de forma eficaz em combinação para melhorar os resultados do LLM (consulte a seção 2.2.5).

Objetivo prático HO-2.1.2a (H0): Observar e discutir o encadeamento de prompts, prompts de poucos disparos e meta prompting **em tarefas de teste de software**

Os participantes terão uma experiência com o encadeamento de prompts, prompts de poucos disparos e meta prompting em um chatbot de IA, cada um aplicado a tarefas específicas de teste de software. A demonstração tem o objetivo de explorar e discutir essas técnicas de prompting no contexto de testes de software, enfatizando como cada técnica contribui para a acurácia e a integridade dos resultados do LLM.

Objetivo prático HO-2.1.2b (H1): Identificar técnicas de engenharia em exemplos dados

Os participantes lerão um conjunto de exemplos de instruções relacionados a testes de software para identificar as principais técnicas de instruções aplicadas. O foco está no reconhecimento de técnicas como encadeamento de prompt, prompting de poucos disparos e meta prompting, destacando suas características distintas e aplicações práticas.

Essa atividade tem como objetivo aprofundar a compreensão dos participantes sobre como as diferentes técnicas de prompting melhoram o uso eficaz da GenAI nos testes de software.

2.1.3 Prompt do sistema e prompt do usuário

Os prompts do sistema e os prompts do usuário têm finalidades diferentes nas interações com os LLMs, cada um desempenhando uma função distinta na formação da conversa. O prompt do sistema é normalmente definido pelo desenvolvedor ou testador, para orientar o comportamento geral do LLM, e não é visível ou editável pelo usuário do chatbot na maioria das interfaces.

Um prompt do sistema atua como um conjunto de comandos predefinidos que define o comportamento, a personalidade e os parâmetros operacionais do LLM. Os parâmetros operacionais determinam como o LLM responde - por exemplo, usando um tom formal, mantendo as respostas concisas, respeitando regras específicas do domínio ou evitando determinado comportamento. O prompt do sistema define as regras para toda a conversa. Ele pode conter partes de um prompt estruturado, como a função, o contexto e as restrições.

O prompt do sistema permanece constante durante toda a sessão de interação e estabelece a estrutura fundamental de como o LLM deve responder. Por exemplo, um prompt do sistema pode dizer: "Você é um assistente profissional de teste de software. Sempre responda com clareza, use linguagem formal e concentre-se nas práticas alinhadas ao ISTQB. Evite especulações e cite os princípios de teste quando for relevante."

O prompt do usuário, por outro lado, representa a entrada ou a pergunta real do usuário do chatbot. Ele muda a cada interação e pode incluir instruções, perguntas ou tarefas específicas que o usuário do chatbot deseja que o LLM resolva. Ao contrário do prompt do sistema, os prompts do usuário são diretamente visíveis e formam o contexto imediato de cada resposta.
Por exemplo, um prompt do usuário pode ser: "Liste as principais diferenças entre os testes de caixa preta e de caixa branca com exemplos."

O uso típico envolve a configuração do prompt do sistema uma vez no início da conversa e, em seguida, o envio de sucessivos prompts do usuário para cada interação. O LLM gera respostas considerando o prompt do sistema, que não muda, e o prompt do usuário atual juntos. Para uma implementação eficaz, os avisos do sistema devem ser claros e específicos sobre a função do LLM e as possíveis restrições. Também podem conter contexto e instruções gerais, por exemplo, sobre o resultado esperado.

Os prompts do usuário devem ser focados e bem estruturados, incluindo instruções explícitas, bem como instruções adicionais relevantes sobre contexto e formato de saída.

2.2 Aplicação de técnicas de engenharia de prompts a tarefas de teste de software

A aplicação de técnicas de engenharia de prompts ao teste de software permite que o GenAI ofereça suporte a tarefas de teste, como análise de teste, projeto de teste, automação de teste, priorização de casos de teste, detecção de defeitos, análise de cobertura, monitoramento e controle de teste. Ao usar e combinar técnicas como encadeamento de prompts, prompts de poucos disparos e meta prompting, as equipes podem adaptar os prompts de AI aos objetivos específicos do teste, tornando os resultados mais precisos, relevantes e eficazes. Uma entrada de alta qualidade é fundamental para obter resultados significativos de IA.

2.2.1 Análise de teste com AI generativa

A GenAI pode dar suporte a tarefas de análise de teste gerando e priorizando condições de teste, identificando defeitos na base de teste e fornecendo análise de cobertura. Os dados de entrada incluem requisitos, histórias de usuários, especificações técnicas, wireframes de GUI e outras informações relevantes. O resultado consiste em produtos de trabalho típicos de análise de teste, como condições de teste priorizadas (por exemplo, critérios de aceite).

Veja a seguir algumas tarefas típicas de análise de teste que podem ser apoiadas pelo GenAI:

- **Identificar possíveis defeitos na base de teste**: O GenAI pode ajudar a analisar a base do teste em busca de inconsistências, ambiguidades ou informações incompletas que possam levar a defeitos. Ao comparar padrões de requisitos semelhantes ou aplicar o conhecimento de relatórios de defeitos anteriores, o LLM pode sinalizar possíveis anomalias e sugerir melhorias.
- **Gerar condições de teste com base na base de teste,** por exemplo, em requisitos/histórias de usuários: Os LLMs podem analisar requisitos e histórias de usuários para gerar condições de teste. Usando o processamento de linguagem natural, eles podem interpretar o significado dos requisitos e dividi-los em

declarações mensuráveis e testáveis. Isso pode ajudar a traduzir os requisitos em condições de teste específicas.

- **Priorize as condições de teste com base no nível de risco:** Com informações sobre a probabilidade de risco e o impacto do risco de falha para cada condição de teste, um LLM pode ajudar a priorizar o esforço de teste. Ao considerar aspectos como conformidade regulamentar, recursos voltados para o usuário (por exemplo, funcionalidade de login ou processamento de pagamentos) e dados históricos de defeitos, o LLM pode recomendar níveis de prioridade.
- **Apoiar a análise de cobertura:** Ao mapear os requisitos e as histórias de usuários para testar as condições, o LLM pode realizar uma análise de cobertura para determinar se todos os aspectos da base de teste estão cobertos. Isso é particularmente útil para projetos com requisitos complexos, em que as lacunas na cobertura podem levar a defeitos escapados.
- **Sugerir técnicas de teste:** O GenAI pode sugerir técnicas de teste relevantes (por exemplo, análise de valor de limite, particionamento de equivalência) com base no tipo de requisito ou história de usuário que está sendo testada. Isso pode ajudar os testadores a aplicarem as técnicas de teste mais eficazes para condições de teste específicas.

A qualidade e a relevância das entradas fornecidas ao LLM em relação à tarefa a ser concluída afetam diretamente a exatidão e a acurácia da saída gerada pelo LLM.

Objetivo prático 2.2.1a (H2): Praticar a criação de prompts multimodais estruturados para gerar critérios de aceite para uma história de usuário com base em um wireframe de GUI

Este é um exercício para praticar a escrita de prompts estruturados usando entrada multimodal (texto e imagem). O objetivo é gerar critérios de aceite de alta qualidade (ou seja, bem formados, claros e completos) a partir de uma história de usuário e de um wireframe de GUI. Outros elementos de texto podem ser adicionados para fornecer contexto, como restrições em campos de entrada ou regras comerciais a serem aplicadas ao processamento de dados.

Os resultados obtidos com o LLM são comparados para avaliar o impacto de diferentes formulações do prompt estruturado (função, contexto, instrução, dados de entrada textuais e de imagem, restrições e formato de saída) para uma tarefa de análise de teste.

Esse exercício fornece experiência prática sobre a importância da estruturação do prompt, a contribuição de instruções precisas e a importância dos dados contextuais textuais e de imagem para a obtenção de resultados precisos e relevantes do LLM.

Objetivo prático 2.2.1b (H2): Praticar o encadeamento e a verificação humana para analisar progressivamente uma determinada história de usuário e refinar os critérios de aceite

Este é um exercício para praticar o encadeamento do prompt para analisar uma determinada história de usuário e refinar os critérios de aceite, primeiro identificando ambiguidades, depois avaliando a testabilidade e, por fim, avaliando a integridade. Esse exercício incentiva uma abordagem passo a passo, refinando a análise em cada etapa para garantir que os critérios de aceite sejam bem formados e acionáveis para atingir os objetivos do teste. Em cada etapa, os resultados fornecidos pelo LLM são verificados manualmente e corrigidos, se necessário, ajustando a saída ou por meio de um processo de encadeamento imediato com o LLM. Dessa forma, a próxima etapa usa um resultado limpo da etapa anterior para abordar outro aspecto do aprimoramento dos critérios de aceite.

Esse exercício proporciona uma experiência prática dos benefícios de dividir uma tarefa complexa em subtarefas, com verificação humana dos resultados de cada estágio.

2.2.2 Projeto de teste e implementação de teste com AI generativa

Conforme descrito em [ISTQB_CTFL_SYL], o projeto de teste envolve a elaboração e o refinamento das condições de teste, que são então traduzidas em casos de teste e em outros materiais de teste. A implementação do teste implica a criação ou aquisição do testware necessário para realizar os testes.

Tanto os testes manuais quanto os scripts de testes automatizados podem ser criados, priorizados e organizados em um cronograma de execução de testes com o apoio do GenAI. O GenAI pode dar suporte significativo a esse grande grupo de atividades de teste, auxiliando na criação e na avaliação de vários equipamentos de teste, incluindo casos de teste, dados de teste, scripts de teste e ambientes de teste.

Aqui estão algumas tarefas típicas de projeto de teste e de implementação de teste que podem ser apoiadas pelo GenAI:

- **Geração de casos de teste**: O processamento de linguagem natural permite que o GenAI crie rascunhos de casos de teste com base em requisitos funcionais e não funcionais. Quando solicitado com informações adequadas, um LLM pode sugerir pré-condições e entradas de teste, resultados esperados e critérios de cobertura, produzindo casos de teste que atendem a diferentes objetivos de teste, desde a verificação funcional básica até testes complexos de ponta a ponta.
- **Síntese de dados de teste**: O GenAI pode criar dados de teste sintéticos representativos e com preservação da privacidade dos dados que se assemelham aos dados de produção, abrangendo situações extremas e condições de teste variadas. Esses dados de teste sintéticos podem ser usados para testes funcionais e não funcionais. Os dados de teste gerados por AI podem ser adaptados aos requisitos do aplicativo, simulando cenários realistas sem expor informações confidenciais.
- **Geração automatizada de scripts de teste**: O GenAI pode gerar procedimentos de teste manuais e scripts de teste automatizados a partir de casos de teste estruturados, interpretando as etapas de teste e traduzindo-as em código compatível com várias estruturas de automação de teste. Esses scripts de teste podem ser atualizados ou ampliados com base em novos requisitos.
- **Programação e priorização da execução de testes**: O GenAI pode analisar os casos de teste e suas interdependências, otimizando os cronogramas de execução de teste com base na prioridade, nos riscos associados, na disponibilidade de recursos e nos objetivos do teste.

Objetivo prático 2.2.2a (H2): Praticar a geração de casos de teste funcionais a partir de histórias de usuários com AI usando o prompt encadeamento, prompts estruturados e meta prompting

Este exercício se concentra no desenvolvimento de casos de teste funcionais a partir de histórias de usuários com o GenAI, usando técnicas de encadeamento de prompts, prompts estruturados e meta prompting para garantir uma cobertura completa. A primeira etapa é criar um prompt que instrua a AI a gerar casos de teste funcionais com base em determinados critérios de aceite seguindo um formato de saída específico. Uma segunda etapa é verificar a integridade dos casos de teste gerados. Aqui, o prompt verifica se cada critério de aceite é coberto, fazendo com que a AI gere uma tabela resumindo a cobertura. Por fim, uma terceira etapa é criar um meta prompting para ajudar na criação de procedimentos de teste de ponta a ponta. Esse meta prompting ajuda a refinar o prompt para gerar testes abrangentes de ponta a ponta, incentivando melhorias iterativas para maximizar a eficácia.

Este exercício aprimora a compreensão do uso dos LLMs para a geração de casos de teste, validação de cobertura e testes de ponta a ponta.

Objetivo prático 2.2.2b (H2): Usar a técnica prompting **de poucos disparos para gerar casos de teste no estilo Gherkin a partir de determinadas histórias de usuários**

Este exercício trata do uso do prompting de poucos disparos para gerar casos de teste no estilo Gherkin a partir de determinadas histórias de usuários. Começando com uma revisão dos exemplos predefinidos e da sintaxe do Gherkin, a etapa 1 é selecionar n exemplos para incluir no prompt, cada um com uma história de usuário, condições de teste e casos de teste esperados no estilo given-when-then para modelar o resultado desejado. Esse prompt é então aplicado a uma nova história de usuário, gerando cenários Gherkin que refletem as condições de teste originais. Se os resultados forem imprecisos, o prompt ou os exemplos devem ser refinados.

Esse exercício ajuda a ganhar experiência na aplicação de técnicas de prompting de poucos disparos a projetos de teste realistas e tarefas de implementação de teste.

Objetivo prático 2.2.2c (H2): Usar o encadeamento de prompting **para priorizar os casos de teste em um determinado conjunto de testes, levando em conta suas prioridades e dependências específicas**

Este exercício se concentra no uso do GenAI para melhorar a priorização de casos de teste em um determinado conjunto de testes com análise de risco associada e dependências entre os casos de teste. A sessão começa com uma breve visão geral das diferentes abordagens de teste, como baseada em risco, baseada em cobertura e baseada em requisitos, e uma revisão do conjunto de testes fornecido. Em seguida, os participantes se envolverão na criação de prompts para gerar planos de priorização acionáveis para várias estratégias de priorização de testes. Os resultados do LLM com base no prompt e nos dados de entrada fornecidos devem ser verificados manualmente para detectar quaisquer erros no raciocínio do LLM.

O objetivo desse exercício é experimentar a GenAI em tarefas de teste que exigem recursos de raciocínio multicritério (aqui, os diferentes riscos e dependências a serem considerados para a priorização de casos de teste).

2.2.3 Teste de regressão automatizado com AI generativa

À medida que cada nova iteração ou versão é concluída, o número de casos de teste de regressão a serem executados geralmente aumenta, tornando-os candidatos ideais para automação, especialmente em pipelines de integração contínua/entrega contínua (CI/CD) devido à alta frequência de execução de testes. O GenAI pode simplificar esse processo, auxiliando na criação, manutenção e otimização de conjuntos de testes de regressão automatizados. Ao se adaptar dinamicamente às alterações na base de código e realizar a análise de impacto, o GenAI pode identificar quais áreas do software têm maior probabilidade de serem afetadas por modificações recentes, concentrando os esforços de teste de regressão onde eles são mais necessários.

Veja a seguir algumas atividades típicas de testes de regressão automatizados e relatórios de testes que podem ser apoiados pelo GenAI prompting:

- **Implementação de script de teste automatizado com automação orientada por palavras-chave**: Os LLMs podem ser usados para implementar scripts de teste com base em estruturas de automação de teste orientadas por palavras-chave, em que palavras-chave predefinidas representam etapas comuns de teste. O GenAI pode mapear essas palavras-chave para casos de teste específicos, gerar scripts de teste e auxiliar os testadores e engenheiros de automação de teste em seu trabalho.
- **Análise de impacto e otimização de testes**: O GenAI pode ser usado para analisar alterações de código a fim de identificar áreas de alto risco, permitindo assim testes de regressão direcionados aonde for mais necessário.
- **Testes de autocorreção e adaptativos**: O GenAI pode ser usado para ajustar automaticamente os scripts de teste para lidar com pequenas alterações na interface do usuário ou na API, evitando falhas desnecessárias decorrentes de pequenas modificações e garantindo que os conjuntos de testes permaneçam estáveis ao longo do tempo.

- **Relatórios de teste automatizados e insights**: O GenAI permite a geração de relatórios de teste detalhados e disponíveis em tempo hábil com métricas de sucesso, falhas e insights importantes, fornecendo aos stakeholders painéis que destacam as tendências de teste e oferecem insights preditivos sobre possíveis pontos de falha.
- **Relatórios aprimorados de defeitos e** análise de **causa raiz**: O GenAI pode oferecer suporte à compilação automática de relatórios abrangentes de defeitos com registros de teste, capturas de tela e dados do ambiente de teste.

Essas atividades podem ser aplicadas a uma variedade de testes de regressão, incluindo testes de regressão funcionais e não funcionais. No entanto, os testadores devem estar cientes de que o GenAI pode cometer erros. Portanto, o resultado gerado deve ser cuidadosamente verificado, dependendo do risco associado (consulte o capítulo 3).

Além disso, o GenAI pode auxiliar testes de regressão automatizados de ponta a ponta baseados em GUI e API, cada um com seus desafios e soluções distintos. Os testes de GUI frequentemente se tornam instáveis devido a alterações recorrentes na interface do usuário. O GenAI pode adaptar automaticamente os scripts de teste para lidar com alterações como localizadores dinâmicos e interações modificadas, reduzindo a necessidade de intervenção manual. Os testes de regressão de API enfrentam desafios como a alteração de formatos de prompt/resposta, pontos de extremidade e autenticação. O GenAI pode adaptar os scripts de teste automaticamente às especificações de API em evolução e gerar diversos dados de teste, mantendo uma cobertura abrangente e reduzindo a necessidade de atualizações manuais.

Objetivo prático 2.2.3a (H2): Praticar o prompting de poucos disparos para criar e gerenciar scripts de teste orientados por palavras-chave

Este exercício se concentra no desenvolvimento e na automação de scripts de teste para um determinado aplicativo da Web usando uma estrutura de automação de teste de GUI. O exercício está estruturado em duas seções principais: automação de testes e depuração de scripts de teste. A primeira parte do exercício fornece orientação sobre a criação de documentação para uma biblioteca de palavras-chave, a geração de scripts de teste iniciais, a validação desses scripts de teste pela AI e a expansão da cobertura com scripts de teste adicionais. A segunda parte enfatiza o suporte à depuração, usando prompts do sistema para criar um assistente de AI que possa verificar e corrigir scripts de teste.

Esse exercício combina a automação de teste tradicional com a validação assistida por IA, demonstrando como o prompting de poucos disparos pode ser usado com eficácia para criar, manter e depurar scripts de teste orientados por palavras-chave.

Objetivo prático 2.2.3b (H2): Praticar a escrita de prompts estruturados para relatórios de teste análise no contexto de testes de regressão

Este exercício ilustra uma abordagem metódica para analisar relatórios de testes de regressão, utilizando prompts estruturados. O processo começa com uma análise dos resultados de teste fornecidos e uma comparação com a especificação do teste. Em seguida, ele avança para o agrupamento de defeitos semelhantes, a manutenção de uma lista de anomalias conhecidas e uma verificação cruzada das descobertas. Cada etapa está vinculada à próxima em uma única conversa no LLM.

A abordagem passo a passo demonstra como os prompts estruturados podem ser usados para transformar os resultados dos testes de regressão e os registros de testes em insights acionáveis, apoiando, assim, uma análise eficaz do relatório de testes no contexto dos testes de regressão.

2.2.4 Monitoramento e controle de testes com AI generativa

As tarefas de monitoramento de testes exigem a recuperação de grandes quantidades de dados (às vezes não estruturados), que geralmente já estão disponíveis em ferramentas de gerenciamento de testes que o GenAI pode ajudar a analisar e sintetizar.

O GenAI facilita várias tarefas de monitoramento e controle de testes, incluindo:

- **Monitoramento de testes e análise de métricas:** O GenAI pode facilitar a automação do monitoramento de testes, bem como a análise de tendências para prever riscos potenciais e alertar as equipes sobre quaisquer desvios do plano. Isso permite que as equipes permaneçam informadas e tomem medidas para manter os padrões de qualidade.
- **Controle de testes:** O GenAI pode ajudar no controle de testes, fornecendo insights para redefinir as prioridades dos testes, ajustar os cronogramas de testes e realocar recursos conforme necessário. Isso garante que os testes permaneçam flexíveis e concentrados em áreas de alta prioridade.
- **Insights sobre a conclusão dos testes e aprendizado contínuo:** O GenAI pode ajudar gerando relatórios de conclusão de testes, destacando os sucessos e as lições aprendidas. Isso permite que as equipes refinem as estratégias de teste e melhorem os processos de teste futuros.
- **Visualização e relatórios aprimorados de métricas de teste:** O GenAI pode ajudar na criação de painéis dinâmicos e resumos em linguagem natural, garantindo que todos os stakeholders tenham acesso às métricas relevantes. Essa assistência fornece as informações necessárias para tomar decisões rápidas e oferece uma visão clara do progresso do teste.

Objetivo prático 2.2.4 (H0): Observar métricas de monitoramento de teste preparadas por AI a partir de dados de teste

Essa demonstração ilustra como o GenAI pode ajudar as equipes de teste, transformando os dados de teste em métricas de monitoramento de teste acionáveis, facilitando assim a tomada de decisões informadas. A partir dos dados de teste extraídos das ferramentas de teste, um LLM os processa para gerar métricas importantes, como progresso do teste, tendências de defeitos ou cobertura, destacando os riscos potenciais. Essas métricas geradas pela AI podem então ser exibidas em um painel e resumidas em linguagem natural para facilitar a compreensão de todos os stakeholders.

Esta demonstração ilustra como o GenAI transforma os dados de teste em insights práticos, ajudando as equipes de teste a monitorarem o progresso dos testes, gerenciar a qualidade e adaptar-se rapidamente às mudanças.

2.2.5 Escolha de técnicas de prompting para teste de software

A tabela a seguir mostra a adequação das três técnicas de prompting mencionadas na seção 2.1.2 de acordo com as características da tarefa de teste.

Técnica de prompting	**Caso de uso recomendado**	**Principais recursos e aplicativos**
Encadeamento de prompt	Tarefas complexas que exigem precisão com verificação humana em cada etapa	Divide as tarefas em etapas menores, úteis para análise de teste, projeto de teste e automação de teste, em que cada etapa de teste é verificada quanto à acurácia.
Prompting de poucos disparos	Tarefas de formato de saída repetitivas ou específicas/restritas	Fornece exemplos para o GenAI para geração repetitiva com um padrão específico, por exemplo, no caso de teste estilo Gherkin (por exemplo, baseado em cenário), teste orientado por palavras-chave ou relatório de teste com um formato de saída específico.
Meta prompting	Tarefas flexíveis e dinâmicas, úteis para criar avisos para novas tarefas	Descrição geral do objetivo e da tarefa a ser executada, que orienta o LLM na criação do prompt. Útil para todos os tipos de tarefas complexas, como análise de relatórios de teste e detecção de anomalias.

É possível até mesmo usar várias técnicas em um único caso de uso. Por exemplo, o meta prompting pode ser usado para criar um prompt inicial. Esse prompt gerado pode conter exemplos que devem ser adaptados e podem ser aprimorados (prompting de poucos disparos). Por fim, pode ser útil dividir a tarefa em subtarefas menores para permitir a validação das etapas intermediárias (encadeamento de prompt).

Objetivo prático 2.2.5 (H1): Seleção de técnicas de estímulo adequadas ao contexto para determinadas tarefas de teste

Este exercício se concentra na seleção de técnicas de estímulo apropriadas para diferentes tarefas de teste. Os participantes recebem várias tarefas de teste com diferentes desafios. Para cada tarefa de teste, os participantes devem avaliar a natureza da tarefa - se ela exige precisão ou estrutura repetitiva - e sugerir a(s) técnica(s) de estímulo que melhor se adapte(m) ao contexto e atenda(m) às necessidades específicas da tarefa. As escolhas são discutidas no grupo.

Esse exercício foi desenvolvido para aprofundar a compreensão de como as diferentes técnicas de prompting podem ser usadas de forma eficaz em esforços de testes práticos.

2.3 Avalie os resultados da AI generativa e refine as instruções para tarefas de teste de software

A avaliação do desempenho da GenAI em testes de software requer um conjunto claro de métricas para avaliar a qualidade, a relevância e a eficácia dos resultados gerados (Li 2024). Essas métricas, sejam elas gerais ou específicas da tarefa, ajudam a otimizar o prompting do LLM.

2.3.1 Métricas para avaliar os resultados da AI generativa em tarefas de teste

Várias métricas podem ser usadas para avaliar a qualidade e a eficiência dos resultados da GenAI em uma tarefa de teste:

Métrica	Descrição	Exemplo
Acurácia	Mede a correção geral do resultado gerado em relação a casos de teste, requisitos ou outros padrões escritos por especialistas.	O grau em que os casos de teste gerados abrangem todos os requisitos especificados.
Precisão	Avalia a correção do resultado gerado com relação a um objetivo específico.	O grau em que os casos de teste gerados identificam corretamente as anomalias.
Recuperação	Mede a capacidade de um modelo de identificar todas as instâncias relevantes em um conjunto de dados.	O grau em que os casos de teste gerados abrangem a partição de equivalência válida e inválida de uma classe de dados.
Relevância e ajuste contextual	Determina se o resultado gerado é aplicável e apropriado para um determinado contexto.	O grau em que os casos de teste gerados são consistentes com a base de teste e integram os requisitos específicos do domínio.
Diversidade	Garante a cobertura de uma ampla gama de entradas e cenários, evitando a repetição.	O grau em que os casos de teste gerados abrangem vários comportamentos do usuário e exploram casos extremos.
Taxa de sucesso na execução	Mede a proporção de artefatos de teste gerados que podem ser executados com sucesso, tal como estão, no ambiente de teste.	Determinar quantos dos scripts de teste gerados podem ser executados sem erros de sintaxe ou problemas de formato de saída em um ambiente de teste

Métrica	Descrição	Exemplo
Eficiência de tempo	Avalia o tempo economizado em comparação com os esforços de teste manual.	O tempo exigido pela AI para gerar casos de teste em comparação com o tempo que um ser humano levaria para criar manualmente testes equivalentes.

Além dessas métricas gerais, as métricas específicas da tarefa podem ser adaptadas para avaliar se o GenAI oferece suporte a atividades de teste específicas.

Para avaliar essas métricas de forma eficaz, os testadores podem realizar revisões manuais ou automatizá-las, por exemplo, comparando a saída do LLM com uma referência predefinida. Dada a natureza não determinística do GenAI, as métricas devem se basear em dados estatisticamente relevantes.

Objetivo prático 2.3.1 (H0): Observar como as métricas podem ser usadas para avaliar o resultado da AI generativa em uma tarefa de teste

Durante uma demonstração em uma determinada tarefa de teste, são mostradas métricas adaptadas à tarefa para avaliar os resultados do GenAI, bem como sua aplicação concreta aos resultados obtidos com um LLM nessa tarefa de teste.

Essa demonstração ilustra a importância das métricas de avaliação para proporcionar confiança nos resultados da AI generativa para testes de software.

2.3.2 Técnicas de avaliação e refinamento iterativo de prompts

Com base nas métricas apresentadas acima, técnicas específicas para avaliação e refinamento de prompts são usadas para melhorar os resultados da IA:

- **Modificação iterativa do prompt:** Comece com um prompt básico e modifique-o iterativamente com base nos resultados observados, adicionando gradualmente mais contexto ou ajustando o texto (por exemplo, com relação à terminologia) para melhorar a especificidade e a relevância.
- **Teste A/B de prompts:** Crie várias versões de prompts e avalie qual versão produz melhor resultado com base em métricas predefinidas. Essa abordagem ajuda a determinar qual fraseado ou estrutura de prompt produz os resultados mais precisos e relevantes.
- **Análise de resultados:** Examine a saída gerada pela AI em busca de imprecisões ou inconsistências, por exemplo, com relação à base de teste. Compreender os tipos de erros e inconsistências pode ajudar a refinar os prompts para evitar defeitos semelhantes em iterações futuras.
- **Integre o feedback do usuário:** Reúna informações dos testadores sobre a utilidade e a clareza do resultado gerado, por exemplo, com relação ao nível de detalhes dos testes gerados. Analise suas percepções e use-as para refinar os prompts e atender melhor às necessidades de testes do mundo real.
- **Ajuste o comprimento e a especificidade do prompt:** Faça experiências com diferentes comprimentos e níveis de detalhes dos prompts. Às vezes, adicionar mais contexto pode melhorar a qualidade da resposta. Em outros casos, solicitações mais curtas podem gerar uma melhor generalização.

Ao usar essas técnicas, as equipes de teste podem organizar sessões de avaliação e otimização do prompt para garantir o aprimoramento contínuo dos prompts do GenAI. O compartilhamento de práticas entre a equipe de teste ou a organização de teste não só ajuda a padronizar as técnicas de prompt e a manter a qualidade consistente, mas também promove uma cultura de aprendizado e melhoria iterativa. Essa abordagem colaborativa contribui para a evolução das metodologias de teste do GenAI, pois permite que as equipes de teste se baseiem em percepções coletivas, evitem erros repetidos e refinem o uso das ferramentas do GenAI de forma mais eficaz ao longo do tempo, por exemplo, compartilhando bibliotecas de prompts.

Objetivo prático 2.3.2 (H1): Avaliar e otimizar um prompt para uma determinada tarefa de teste

Este exercício se concentra na aplicação de técnicas de otimização do prompt em uma determinada tarefa de teste. Os participantes começarão com um prompt inicial e o refinarão iterativamente para melhorar os resultados gerados pela IA. Eles usarão técnicas como teste A/B e verificação humana para avaliar e melhorar a qualidade dos prompts. O objetivo é que os participantes experimentem como o refinamento iterativo leva a uma geração de casos de teste mais eficaz e contextualmente relevante.

Ao final do exercício, os participantes terão realizado várias iterações de refinamento do prompt e avaliado cada iteração usando as métricas discutidas para melhorar a qualidade do resultado da IA.

3 Gerenciando os Riscos da AI Generativa em Testes de Software - 160 min.

Palavras-chave

segurança, vulnerabilidade, privacidade de dados

Palavras-chave específicas da AI generativa

alucinação, temperatura, raciocínio erro, viés

Objetivos de aprendizagem e objetivos práticos do Capítulo 3:

3.1 Alucinações, erros de raciocínio e vieses

GenAI- 3.1.1	(K1)	Relembrar as definições de alucinações, raciocínio erros e vieses em sistemas de AI generativa
GenAI- 3.1.2	(K3)	Identificar alucinações, erros de raciocínio e vieses nos resultados dos modelos de linguagem de grande escala (LLM)
HO-3.1.2a	(H1)	Experimentar alucinações em testes com a GenAI
HO-3.1.2b	(H1)	Experiência com raciocínio erros em testes com a GenAI
GenAI- 3.1.3	(K2)	Resumir as técnicas de atenuação da GenAI alucinações, raciocínio erros e vieses em tarefas de teste de software
GenAI- 3.1.4	(K1)	Relembrar técnicas de atenuação para comportamento não determinístico de LLMs

3.2 Privacidade de dados e riscos de segurança da AI generativa em testes de software

GenAI- 3.2.1	(K2)	Explicar os principais riscos de privacidade de dados e de segurança associados ao uso de AI generativa em testes de software
GenAI- 3.2.2	(K2)	Dar exemplos de privacidade de dados e vulnerabilidades no uso da AI generativa em testes de software
GenAI- 3.2.3	(K2)	Resumir as estratégias de mitigação para proteger a privacidade dos dados e aprimorar a segurança na AI generativa para teste de software
HO-3.2.3	(H0)	Reconhecer a privacidade dos dados e os riscos de segurança em um determinado estudo de caso de AI generativa para teste

3.3 Consumo de energia e impacto ambiental da AI generativa para teste de software

GenAI- 3.3.1	(K2)	Explicar o impacto das características da tarefa e do uso do modelo no consumo de energia da AI generativa em testes de software
HO-3.3.1	(H1)	Use um simulador para calcular a energia e as emissões de CO_2 para determinadas tarefas de teste com AI generativa

3.4 Regulamentos, padrões e estruturas de práticas recomendadas de IA

GenAI- 3.4.1	(K1)	Relembrar exemplos de regulamentações, normas e estruturas de melhores práticas de AI relevantes para a AI generativa em testes de software

3.1 Alucinações, erros de raciocínio e vieses

Os sistemas de GenAI, especialmente os LLMs, são propensos a certos defeitos, incluindo alucinações, erros de raciocínio e vieses. Esses defeitos reduzem a qualidade da saída do GenAI nas tarefas de teste, resultando em um testware gerado que não atende às expectativas dos testadores. Essas alucinações, erros de raciocínio e vieses precisam ser identificados pelos testadores na saída do LLM, e medidas devem ser tomadas para mitigar esses riscos.

O comportamento não determinístico dos LLMs (consulte a seção 1.1.2) dificulta a correção desses tipos de defeitos; eles podem parecer corrigidos em uma saída do LLM, mas reaparecem em outra conversa com o mesmo LLM.

3.1.1 Alucinações, erros de raciocínio e vieses na AI generativa

As alucinações ocorrem quando um LLM gera resultados que parecem factualmente incorretos ou irrelevantes para uma determinada tarefa. Nos testes de software, as alucinações podem se manifestar como LLMs criando casos de teste fictícios ou irrelevantes, gerando scripts de teste incorretos ou que não funcionam ou sugerindo casos de teste que verificam critérios de aceite inexistentes. Isso pode enganar os testadores e comprometer a validade dos resultados dos testes.

Erros de raciocínio ocorrem quando os LLMs interpretam erroneamente estruturas lógicas, como relações de causa e efeito, lógica condicional ou processos de solução de problemas passo a passo, levando a conclusões incorretas. Diferentemente dos seres humanos, os LLMs não têm um raciocínio lógico verdadeiro e dependem da correspondência de padrões, o que pode levar a uma lógica errônea ao realizar tarefas como raciocínio matemático (Mirzadeh 2024). Planejamento de testes e priorização de casos de teste são exemplos de tarefas de teste que exigem raciocínio lógico e onde os LLMs podem cometer erros de raciocínio.

Vieses do LLM (Gallegos 2024) vêm dos dados nos quais o modelo foi treinado. Esses vieses podem levar a resultados que favorecem determinados tipos de informações, abordagens ou suposições. Por exemplo, os LLMs treinados principalmente em dados em inglês podem sub-representar perspectivas que não sejam em inglês. Nos testes de software, os vieses podem influenciar as respostas dos LLMs quando, por exemplo, geram dados de teste ou refinam os critérios de aceite para casos de teste.

As alucinações, os erros de raciocínio e os vieses na saída do GenAI resultam da natureza de seus dados de treinamento e das limitações inerentes ao modelo do transformador (consulte o Capítulo 1). O reconhecimento e a abordagem desses desafios aumentam a qualidade dos resultados da AI generativa nos processos de teste.

3.1.2 Identificar alucinações, erros de raciocínio e vieses na saída do LLM

A integração eficaz dos sistemas de GenAI nos testes de software requer a capacidade de detectar alucinações, erros de raciocínio e vieses na saída do LLM. Dependendo do tipo de problema, diferentes abordagens de detecção podem ser aplicadas. A seguir, apresentamos abordagens comuns que são aplicadas por meio de revisão ou de uma combinação de revisão e verificação automatizada:

Detecção de alucinação:

- Verificação cruzada: Comparar os resultados gerados pela IA com a documentação existente, os requisitos e o comportamento conhecido do sistema. Ferramentas automatizadas podem ajudar a cruzar os resultados com fontes de dados confiáveis para identificar discrepâncias.
- Consulta a especialistas no domínio: Envolva especialistas no assunto para validar a acurácia do conteúdo gerado. O conhecimento deles é essencial para capturar percepções diferenciadas que os sistemas automatizados podem ignorar.
- Verificações de consistência: Verifique se os resultados gerados são consistentes entre si e com as informações conhecidas. Os sistemas automatizados podem ajudar a identificar padrões e sinalizar inconsistências.

Detecção de erros de raciocínio:

- Validação lógica: Avalie o fluxo lógico (por exemplo, a consistência, a coerência e o raciocínio estruturado no texto gerado) do conteúdo gerado pela AI quanto à coerência e à correção por meio de ciclos de

revisão. Ferramentas automatizadas podem ajudar, mas casos complexos podem exigir julgamento humano.

- Teste de saída: Por exemplo, executar os casos de teste gerados ou os scripts de teste nos objetos de teste para verificar os resultados do teste. Isso pode ser parcial ou totalmente automatizado, dependendo do tipo de testware que está sendo gerado.

Detecção de viés:

- Revisão de como o material de teste gerado, como dados de teste sintéticos, é representado de forma justa e precisa em relação à estratégia de teste
- Avaliação de vieses relacionados aos tipos de teste, como testes não funcionais sub-representados na saída gerada do LLM.

A implementação real desses métodos de detecção dependerá do nível de risco estimado de alucinações, raciocínio, erros ou vieses na tarefa de teste que está sendo realizada com o GenAI.

Objetivo prático 3.1.2a (H1): Experimentar alucinações de AI generativa relacionadas a uma tarefa de teste de software

Este exercício se concentra na experimentação de exemplos de alucinações da GenAI em relação ao conjunto de conhecimentos de teste de software. Os participantes tentarão confrontar pelo menos dois LLMs com uma situação em que os LLMs inventam elementos irrelevantes, por exemplo, adicionam critérios não relacionados que não existem nos dados contextuais fornecidos. As variações no prompting são testadas para examinar a influência do prompt nas alucinações.

Esse exercício aumenta a compreensão da identificação de alucinações do GenAI em testes de software.

Objetivo prático 3.1.2b (H1): Experimentar o raciocínio da AI generativa erros em uma tarefa de planejamento de teste

Este exercício se concentra em apresentar um exemplo de erro de raciocínio da GenAI. Um exemplo de um problema a ser resolvido na área de planejamento de testes, como a estimativa do esforço de teste e a priorização dos casos de teste (consulte [ISTQB_CTFL] - Capítulo 5). O exercício foi projetado com uma certa complexidade de dados de entrada, o que exige habilidades de resolução de problemas e destaca as limitações dos LLMs para essa finalidade. O resultado do LLM será comparado com o resultado exato que deve ser obtido. Três tipos diferentes de LLM serão testados (LLM, SLM e modelo de raciocínio), e variações do prompt serão usadas para tentar melhorar os resultados.

Esse exercício aumenta a compreensão de como identificar erros de raciocínio do GenAI em tarefas de teste de software que exigem habilidades de resolução de problemas lógicos.

3.1.3 Técnicas de atenuação das alucinações, do raciocínio e dos vieses da GenAI em tarefas de teste de software

Para minimizar os resultados indesejáveis da GenAI em testes de software, várias estratégias podem ser empregadas para reduzir as alucinações, os erros de raciocínio e os vieses. É mais provável que esses problemas ocorram quando os prompts não são projetados adequadamente (consulte o Capítulo 2) ou quando faltam dados de entrada contextuais relevantes para uma determinada tarefa de teste. As principais técnicas para atenuar os riscos associados a alucinações, erros de raciocínio e vieses da AI incluem:

- Fornecer contexto completo: Certifique-se de que o prompt contenha todas as informações relevantes (consulte a seção 2.1.1), oferecendo um contexto abrangente para orientar a AI na produção de resultados precisos.
- Dividir os prompts em segmentos gerenciáveis: Divida prompts complexos em etapas menores usando técnicas de encadeamento de prompts (consulte a seção 2.1.2), verificando sistematicamente cada saída

antes de passar para a próxima. Essa abordagem passo a passo pode ajudar a detectar erros de raciocínio no início do processo de geração.

- Use formatos de dados claros e interpretáveis: Evite formatos que possam ser ambíguos ou difíceis de serem interpretados pelo GenAI. Formatos estruturados e diretos ajudam o modelo a se concentrar nos aspectos essenciais da tarefa.
- Selecione o modelo GenAI apropriado para a tarefa: Use um LLM especificamente treinado para a tarefa em questão (consulte a seção 5.1.3).
- Compare os resultados entre os modelos: Quando apropriado, avaliar o prompt com vários LLMs e comparar os resultados ajuda a detectar erros de saída e a selecionar os resultados mais confiáveis.

O Capítulo 4 apresenta duas técnicas complementares para aprimorar os resultados do LLM: Geração Aumentada por Recuperação e Ajuste Fino.

3.1.4 Mitigação do comportamento não determinístico dos LLMs

O comportamento não determinístico inerente dos LLMs (Shuyin 2023) pode levar a variações na saída, mesmo quando a mesma entrada é fornecida. Isso decorre dos processos de amostragem probabilística usados durante a inferência. Consequentemente, obter resultados consistentes e reproduzíveis ao usar LLMs pode ser um desafio, especialmente para resultados longos, o que aumenta o risco de variabilidade.

Embora a reprodutibilidade total não possa ser garantida, algumas estratégias podem ajudar a reduzir a variabilidade:

- Ajuste do parâmetro de temperatura do LLM configurações: Diminuir a temperatura durante a geração de respostas (inferência) reduz a distribuição de probabilidade, reduzindo a aleatoriedade e resultando em resultados mais consistentes. Entretanto, isso também limitará a criatividade e a diversidade das respostas, tornando os resultados mais repetitivos ou excessivamente determinísticos.
- Definição de sementes aleatórias: Algumas implementações do LLM permitem definir um valor de semente para o gerador de números aleatórios, garantindo que a mesma sequência pseudoaleatória (ou seja, valores aleatórios determinísticos) seja usada, o que melhora a reprodutibilidade.

A redução do risco de alucinações e erros de raciocínio na saída do LLM envolve a abordagem desse comportamento não determinístico, por exemplo, automatizando alguns aspectos da verificação de saída para garantir um processo de avaliação estruturado e consistente.

3.2 Privacidade de dados e riscos de segurança da AI generativa em testes de software

A GenAI em testes apresenta riscos relacionados à privacidade de dados e à segurança devido ao manuseio de informações confidenciais e possíveis vulnerabilidades na infraestrutura de testes com LLM. Uma proteção robusta dos dados é essencial para evitar violações, acesso não autorizado e exposição de dados confidenciais.

3.2.1 Riscos de segurança e privacidade de dados associados ao uso de AI generativa

A GenAI pode processar grandes quantidades de dados que podem conter informações confidenciais ou de identificação pessoal. Isso gera as seguintes preocupações com a privacidade dos dados:

- Exposição não intencional de dados: os modelos do GenAI podem gerar resultados que revelam acidentalmente informações confidenciais.
- Falta de controle sobre o uso de dados: As ferramentas do GenAI podem armazenar e processar dados confidenciais sem o consentimento ou controle explícito do usuário. Isso pode levar a um possível uso indevido ou acesso não autorizado.
- Riscos de conformidade: O uso das ferramentas da GenAI sem o cumprimento das normas de proteção de dados, como o Regulamento Geral de Proteção de Dados (GDPR, Regulamento (UE) 2016/679), pode levar a disputas legais.

Além disso, surgem riscos específicos de segurança ao testar com o GenAI, como:

- A infraestrutura de teste com tecnologia LLM pode ser vulnerável a ataques de segurança, como violações de dados ou acesso não autorizado.
- Agentes mal-intencionados podem explorar vulnerabilidades nos LLMs, como ataques manipulativos (consulte a seção 3.2.2), para alterar seu comportamento ou extrair informações confidenciais.
- Os invasores podem introduzir intencionalmente dados de entrada maliciosos para enganar os LLMs e comprometer sua acurácia ou segurança.

3.2.2 Privacidade de dados e vulnerabilidades na AI generativa para processos e ferramentas de teste

A tabela a seguir apresenta alguns exemplos de vetores de ataque nos processos e ferramentas de teste da GenAI.

Vetor de ataque	Descrição	Exemplo
Manipulação do contexto	Envio de solicitações destinadas a extrair dados de treinamento confidenciais.	Exceder a janela contextual do LLM com solicitações longas para sobrecarregar a memória da AI poderia levá-la a revelar trechos aleatórios de seus dados de treinamento e, possivelmente, expor informações sensíveis.
Manipulação de solicitações	Introduzir dados que atrapalhem o resultado da IA.	Imagens que atraem a AI para um contexto diferente, provocando alucinações sobre, por exemplo, critérios de aceite.
Envenenamento de dados	Manipulação de dados de treinamento.	Fornecimento de avaliações falsas ao classificar os resultados de um relatório de teste gerado por IA.
Geração de código malicioso	Manipulação de um LLM para gerar backdoors (por exemplo, chamadas de comandos externos) durante o uso.	Geração de código para abrir um canal de comunicação com um IP específico e malicioso.

3.2.3 Estratégias de mitigação para proteger a privacidade dos dados e aumentar a segurança nos testes com AI generativa

À medida que a GenAI se torna popular, e com os riscos inerentes envolvidos, surgem normas e padrões para mitigá-los (consulte a seção 3.4.1).

Os regulamentos de proteção de dados, como o GDPR, não restringem explicitamente os aplicativos da GenAI, mas fornecem salvaguardas que podem limitar o que pode ser feito, principalmente em relação à legalidade e às limitações das finalidades de coleta, processamento e armazenamento de dados.

Para mitigar esses riscos, as organizações devem implementar medidas robustas de privacidade de dados, inclusive:

- Minimização de dados: Evitar o processamento de dados confidenciais, a menos que seja legalmente permitido, e usar apenas a quantidade necessária de dados não confidenciais em testes de AI para reduzir os riscos de privacidade de dados.
- Anonimização e pseudonimização de dados: Mascarar ou substituir informações confidenciais por dados não identificáveis.
- Armazenamento e transmissão seguros de dados: Implementação de criptografia forte e controles de acesso.

- Treinamento de recursos: As organizações devem estabelecer programas e políticas de treinamento claros para garantir o uso responsável das ferramentas da GenAI, promover práticas éticas e mitigar os possíveis riscos.

Estratégias adicionais de mitigação podem ser consideradas ao implementar a GenAI para testes:

- Revisão sistemática dos resultados gerados: A avaliação humana é essencial para garantir a qualidade e a acurácia das tarefas de teste com o GenAI.
- Avaliação por comparação com outro LLM: isso envolve o uso de vários LLMs em uma determinada tarefa para avaliar os resultados por meio da comparação de suas respostas.
- Escolha de um ambiente operacional seguro: Dependendo do nível de confidencialidade exigido, as organizações podem optar por diferentes soluções seguras: Usar uma oferta comercial segura de um provedor de LLM, operar o LLM em uma nuvem segura ou instalar o LLM na infraestrutura da organização.
- Auditorias regulares de segurança e avaliações de vulnerabilidade: Identificar e tratar os pontos fracos dos sistemas do GenAI.
- Manter-se atualizado com as práticas recomendadas de segurança: Manter-se atualizado com as mais recentes diretrizes e tecnologias de segurança.

As estratégias geralmente são complementares umas às outras e é necessária uma combinação delas para garantir a segurança dos dados ao usar o GenAI. É altamente recomendável envolver os engenheiros de segurança sênior, o conselho jurídico, o diretor de tecnologia (CTO) ou o diretor de segurança da informação (CISO), se houver na organização.

Objetivo prático 3.2.3 (H0): Reconhecer a privacidade dos dados e a segurança riscos em um determinado estudo de caso de AI generativa para testes

Esta demonstração ilustra como os riscos de privacidade de dados e de segurança podem surgir ao usar a GenAI em testes de software. Os participantes explorarão estudos de caso para identificar possíveis ameaças, como vulnerabilidades de modelos, acesso não autorizado a dados ou uso malicioso de resultados gerados. Eles explorarão estratégias de mitigação, incluindo o manuseio seguro de dados, controles de acesso robustos e práticas de monitoramento de IA, enquanto refletem sobre as implicações éticas e práticas.

Ao final, os participantes entenderão os princípios de privacidade de dados e aprenderão a reconhecer e abordar os riscos de segurança nas condições de teste da GenAI.

3.3 Consumo de energia e impacto ambiental da AI generativa em testes de software

Estudos como o de (Luccioni 2024a) mostram que o treinamento e o processamento de LLMs exigem o uso intensivo de um grande número de recursos de computação especializados. Os LLMs são disponibilizados como serviços baseados na Web, e seu uso aumenta a carga em dispositivos, redes e data centers, levando a um maior consumo de energia.

3.3.1 O impacto do uso do GenAI no consumo de energia e nas emissões de CO2

O impacto ambiental do GenAI não deve ser subestimado, pois o consumo de energia aumenta drasticamente à medida que o uso aumenta. A complexidade da tarefa e os recursos de computação necessários influenciam o consumo de energia. Por exemplo, gerar uma única imagem usando um modelo de AI avançado pode consumir tanta energia quanto carregar totalmente um smartphone, enquanto a geração de texto consome apenas uma pequena porcentagem da carga de um smartphone (Heikkilä 2023).

Mesmo que seja difícil obter dados precisos sobre o impacto ambiental da GenAI (Luccioni 2024b), está claro que essas operações que consomem muita energia contribuem coletivamente para emissões significativas de CO_2 (Berthelot 2024). Embora uma única tarefa de pesquisa ou de geração de texto possa parecer insignificante, seu efeito cumulativo em milhões de usuários em todo o mundo resulta em uma pressão ambiental substancial.

A adoção de práticas recomendadas, como a limitação de interações desnecessárias entre modelos, é fundamental para mitigar os riscos ambientais apresentados pela GenAI.

Objetivo prático 3.3.1 (H1): Usar um simulador para calcular a energia e as emissões de CO_2 para determinadas tarefas de teste com AI generativa

Este exercício se concentra na avaliação do consumo de energia e das emissões de CO_2 associadas a várias tarefas de AI generativa no teste de software. Os participantes usarão simulações para calcular essas métricas e examinarão como as diferentes características da tarefa e o uso do modelo afetam o impacto ambiental.

Ao observar como diferentes fatores afetam o consumo de energia e as emissões, os participantes entenderão os fatores que impulsionam o consumo de energia com LLMs.

3.4 Regulamentações, padrões e estruturas de práticas recomendadas de IA

A GenAI está transformando os testes de software ao auxiliar os testadores em uma variedade de tarefas de teste (consulte o Capítulo 2). Entretanto, essas oportunidades também trazem riscos significativos, como erros de raciocínio, privacidade de dados, vulnerabilidades e impactos ambientais (consulte as seções 3.1, 3.2 e 3.3). A abordagem desses riscos deve considerar regulamentações gerais, padrões e estruturas de práticas recomendadas para IA.

3.4.1 Regulamentações, padrões e estruturas de AI relevantes para a GenAI em testes de software

Abaixo está uma visão geral das principais diretrizes relevantes para o uso da GenAI em testes de software:

Nome/ Tipo	Descrição	Aplicação em testes de software
ISO/IEC 42001:2023 Tecnologia da informação - Inteligência artificial - Sistema de gerenciamento Tipo: Norma	Especifica os requisitos para o gerenciamento de sistemas de AI em uma organização.	Garante que o GenAI em testes adere às práticas recomendadas, promovendo consistência e confiabilidade.
ISO/IEC 23053:2022 Estrutura para sistemas de inteligência artificial (IA) usando machine learning Tipo: Padrão	Fornece uma estrutura para processos de ciclo de vida de IA, enfatizando a tolerância a falhas e a transparência.	Fornece uma estrutura para qualidade de dados, transparência e tolerância a falhas ao usar o GenAI para testes.
Lei de AI da UE Tipo: Regulamento	Estabelece uma estrutura legal que aborda os riscos de IA, classificando os aplicativos por nível de risco. Fonte: (AI Act 2024)	Obriga a conformidade com a transparência, a responsabilidade e a mitigação de viés para a GenAI usada em testes.
Estrutura de gerenciamento de riscos de AI do NIST (EUA) Tipo: Estrutura	Oferece diretrizes para o gerenciamento de riscos de IA, com foco em justiça, transparência e segurança. Fonte: (NIST AI RMF 1.0)	Garante a imparcialidade e reduz os riscos na GenAI, evitando resultados de testes tendenciosos.

Como as tecnologias de AI e seus cenários regulatórios continuam a evoluir, é imperativo que as organizações de teste se mantenham atualizadas sobre o desenvolvimento de regulamentações, padrões, leis nacionais e estruturas de práticas recomendadas, como as desta tabela.

4 Infraestrutura de teste com tecnologia LLM para Teste de Software - 110 min.

Palavras-chave

infraestrutura de teste

Palavras-chave específicas de AI generativa

ajuste fino, agente com tecnologia LLM, Large Language Model Operations, geração aumentada por recuperação, banco de dados vetorial

Objetivos de aprendizagem e objetivos práticos do Capítulo 4:

4.1 Abordagens arquitetônicas para Infraestrutura de teste com tecnologia LLM

GenAI- 4.1.1	(K2)	Explicar os principais componentes e conceitos arquitetônicos da infraestrutura de teste com tecnologia LLM
GenAI- 4.1.2	(K2)	Resumir a geração aumentada por recuperação
HO-4.1.2	(H1)	Experimentar a geração aumentada por recuperação para uma determinada tarefa de teste
GenAI- 4.1.3	(K2)	Explicar a função e a aplicação de agentes com tecnologia LLM na automação de processos de teste
HO-4.1.3	(H0)	Observar como um agente com tecnologia LLM auxilia na automação de uma tarefa de teste repetitiva

4.2 Ajuste fino e LLMOps: Operacionalização da AI generativa para teste de software

GenAI- 4.2.1	(K2)	Explicar o ajuste fino dos modelos de linguagem para tarefas de teste específicas
HO-4.2.1	(H0)	Observar um exemplo de um processo de ajuste fino para uma determinada tarefa de teste e modelo de idioma
GenAI- 4.2.2	(K2)	Explicar o LLMOps e sua função na implementação e no gerenciamento de LLMs para tarefas de teste

4.1 Abordagens arquitetônicas para infraestrutura de teste com tecnologia LLM

Os chatbots de AI e as ferramentas de teste com tecnologia LLM são dois tipos de infraestruturas de teste que usam LLMs. (consulte a seção 1.2.2).

Além da arquitetura básica de uma infraestrutura de teste com tecnologia LLM (consulte a seção 4.1.1), a Geração Aumentada por Recuperação (consulte a seção 4.1.2) e as arquiteturas de agentes com tecnologia LLM (consulte a seção 4.1.3) ampliam a funcionalidade e a utilidade do uso de LLMs em testes de software.

4.1.1 Principais componentes e conceitos arquitetônicos da infraestrutura de teste com tecnologia LLM

Uma infraestrutura de teste com tecnologia LLM refere-se a um sistema que integra um LLM ao processo de teste de software para aprimorar a automação, o raciocínio e a tomada de decisões. Ao contrário de um chatbot de AI tradicional, que se concentra principalmente em interações de conversação, uma ferramenta de teste com LLM foi projetada para dar suporte a testes de software, processando consultas relacionadas a testes, analisando requisitos, gerando casos de teste e avaliando resultados.

A arquitetura típica de uma infraestrutura de teste com tecnologia LLM segue um design de vários componentes que facilita a interação segura e eficiente com o LLM. A arquitetura consiste em componentes de front-end e back-end, juntamente com fontes de dados externas e um LLM integrado:

- O front-end funciona como a interface do usuário em que os testadores interagem com o sistema inserindo consultas ou comandos.
- O back-end processa a entrada do usuário e gerencia funções críticas, como autenticação, recuperação de dados, preparação do prompt e interação com o LLM.
- O LLM, que pode ser hospedado como um serviço de terceiros (acessado via API) ou um modelo interno personalizado, gera respostas com base em prompts estruturados.

Essa arquitetura vai além de um modelo tradicional de cliente-servidor ao incorporar componentes de processamento inteligente, como LLMs e back-ends de várias fontes:

1. O LLM não é apenas um servidor, mas um componente de processamento inteligente que interpreta e raciocina com base em produtos de teste.
2. Ao contrário dos chatbots baseados em regras que seguem respostas roteirizadas, uma infraestrutura de teste com tecnologia LLM gera insights de teste dinamicamente a partir do contexto, como requisitos, código ou resultados de testes.
3. O back-end integra várias fontes de dados, como:
    - Bancos de dados relacionais (para dados estruturados usados em testes, como casos de teste).
    - Bancos de dados vetoriais (para recuperação semântica de conteúdo relacionado usando embeddings; consulte a seção 4.1.2).
4. O back-end aprimora a saída bruta do LLM por meio de pós-processamento, garantindo que suas respostas estejam alinhadas com as condições de teste do processo de teste antes de apresentá-las ao front-end.

4.1.2 Geração aumentada por recuperação

O Geração Aumentada por Recuperação (RAG - Retrieval-Augmented Generation) aprimora os LLMs incorporando fontes de dados adicionais em seu processo de geração de respostas (Zhao 2024), aumentando assim a relevância e a acurácia de seus resultados.

O RAG combina sistemas de recuperação com modelos de linguagem para gerar respostas com reconhecimento de contexto. Durante o pré-processamento, os documentos grandes são divididos em partes menores (por exemplo, 256-512 tokens) para garantir a recuperação focada e a compatibilidade com a janela de contexto do modelo. Cada bloco é limpo, processado e codificado em um vetor de alta dimensão (embedding) usando modelos pré-treinados. Essas incorporações, que podem ser armazenadas em bancos de dados de vetores, permitem a recuperação eficiente baseada em similaridade em tempo de execução (inferência). Uma consulta do usuário é

codificada, os trechos relevantes são recuperados com base na similaridade semântica e esses trechos são usados como contexto para o modelo de linguagem gerar uma resposta fundamentada.

Uma resposta relevante é essencialmente um resultado gerado pelo modelo de linguagem que está profundamente enraizado em informações relevantes, precisas e contextualmente apropriadas coletadas durante o processo de recuperação. Isso garante que a resposta não se baseie apenas no treinamento pré-existente do modelo, mas também seja enriquecida com dados precisos pertinentes ao prompt. Essa sinergia entre a recuperação e a geração aumenta a acurácia e a relevância das respostas, tornando-as mais confiáveis e informativas para o usuário.

Na fase de processamento do prompt do usuário, um sistema RAG funciona por meio de um processo de duas etapas:

1. Recuperação: Dada uma consulta do usuário, o sistema recupera informações relevantes dos bancos de dados vetoriais criados anteriormente. Normalmente, essa recuperação se baseia na semelhança semântica entre os embeddings do prompt e os dos blocos.
2. Geração: As informações recuperadas são então fornecidas ao LLM, que gera uma resposta que combina seu conhecimento existente com os dados recém-adquiridos, resultando em uma saída mais precisa e contextualmente apropriada.

O RAG em testes de software permite que a infraestrutura de testes do LLM acesse as fontes de dados corporativos da empresa, como bancos de dados, documentação e repositórios, para recuperar informações contextuais em tempo real, garantindo que as tarefas de teste, como análise ou projeto de teste, estejam alinhadas com as especificações, os requisitos e os dados de teste mais recentes.

Objetivo prático 4.1.2 (H1): Experimentar a Geração Aumentada por Recuperação para uma determinada tarefa de teste

Este exercício prático concentra-se na aplicação das técnicas do RAG para uma determinada tarefa de teste. Os participantes farão experiências com um sistema RAG incorporando documentos e observarão como ele gera respostas mais ou menos precisas com base em informações complexas. Os participantes compararão o resultado do LLM com e sem RAG em uma determinada tarefa de teste. Esse exercício tem o objetivo de identificar os pontos fortes e as limitações do sistema RAG no tratamento de diferentes tipos de tarefas de teste.

Ao examinar os dados recuperados e os resultados gerados, os participantes terão uma visão da função do RAG no aprimoramento dos processos de teste do LLM.

4.1.3 A função dos agentes com tecnologia LLM na automação dos processos de teste

Os agentes com tecnologia LLM (Wang 2024) são aplicativos especializados de GenAI com tecnologia LLMs e projetados para o processamento semiautônomo ou autônomo de tarefas definidas. Em sua essência, esses agentes contam com LLMs para compreensão e geração de linguagem natural, complementados pela possibilidade de processar instruções, recuperar contexto e realizar ações inteligentes.

Diferentemente dos chatbots de AI tradicionais que se concentram apenas em interações de pergunta-resposta, os agentes com tecnologia LLM podem executar tarefas ou "agir" invocando um conjunto predefinido de funções, comumente chamadas de "ferramentas". Esse recurso permite que eles interajam e manipulem sistemas externos, tornando-os altamente versáteis na execução de tarefas. O grau de autonomia dos agentes com LLM pode variar:

- Os agentes autônomos operam de forma independente, executando tarefas com o mínimo de intervenção humana usando regras predefinidas, aprendizado por reforço e loops de feedback adaptativos.
- Os agentes semiautônomos executam tarefas com supervisão humana periódica para garantir que o resultado atenda às metas definidas pelo usuário.

As arquiteturas multiagentes envolvem um sistema colaborativo no qual múltiplos agentes, cada um com funções especializadas, se comunicam e se coordenam para resolver problemas complexos de forma mais eficiente do que um único agente. Esse esforço coordenado entre múltiplos agentes de IA é conhecido como “orquestração”,

em que agentes alimentados por LLM podem automatizar tarefas de teste ao emular o raciocínio e a tomada de decisões humanos.

Nas ferramentas de teste modernas, os agentes baseados em LLM são frequentemente disponibilizados aos usuários por meio de assistentes de AI integrados ao fluxo de trabalho de testes. Esses assistentes podem automatizar as atividades de teste ao longo de todo o ciclo de vida dos testes, transformando histórias do usuário ou requisitos em artefatos de teste. Isso inclui realizar análises de teste, modelagem de teste, implementação de testes, execução de testes e geração de relatórios de teste de maneira semiautônoma. Isso permite um fluxo de trabalho de automação contínuo e de ponta a ponta que reduz o esforço manual e encurta os ciclos de feedback, ao mesmo tempo em que muda a automação de testes da execução baseada em scripts para uma automação orientada por objetivos e baseada em agentes.

No entanto, esses agentes enfrentam os mesmos problemas de possíveis alucinações, erros de raciocínio e vieses observados no uso de LLMs (ver Seção 3.1). Esses agentes podem produzir resultados incorretos ou enganosos, o que pode comprometer a confiabilidade dos processos de teste automatizados. Esses riscos podem ser mitigados por meio da implementação de procedimentos de verificação automatizados para os resultados dos agentes ou do uso de agentes semiautônomos para tarefas críticas.

Objetivo prático 4.1.3 (H0): Observar como um agente com tecnologia LLM auxilia na automação de uma tarefa de teste repetitiva

A demonstração se concentra em uma tarefa de teste executada por um agente **com tecnologia** LLM. Os dados de entrada passados para o agente, seu comportamento e os resultados de suas ações serão demonstrados para ilustrar os vários aspectos da integração de soluções baseadas em agentes em um processo de teste.

Essa demonstração mostra um exemplo concreto de um agente **com tecnologia** LLM no contexto de uma tarefa de teste.

4.2 Ajuste fino e LLMOps: Operacionalização da AI generativa para teste de software

Duas práticas fundamentais para operacionalizar a infraestrutura de teste com tecnologia LLMs para testes incluem o ajuste fino LLMs e o gerenciamento do pipeline operacional por meio de LLMOps (Mailach 2024).

4.2.1 Ajuste fino dos LLMs para tarefas de teste

O ajuste fino adapta um Modelo de Linguagem (LM) pré-treinado, como um LLM ou um Modelo de Linguagem Pequeno (SLM, consulte a seção 1.1.2), para executar tarefas específicas ou adaptá-lo a domínios específicos (Parthasarathy 2024). Isso envolve o treinamento adicional do modelo em um conjunto de dados direcionado, permitindo que ele aprenda conhecimentos e nuances específicos do domínio. Ao fazer o ajuste fino, o desempenho do modelo é aprimorado para aplicativos especializados, tornando-o mais preciso e relevante para o caso de uso pretendido.

Na prática, o ajuste fino é adequado para equipar LLMs genéricos com habilidades de raciocínio especializadas relevantes para um domínio específico ou para adotar um vocabulário exclusivo desse campo. O ajuste fino também pode ser aplicado a modelos menores, conhecidos como SLMs, que consomem menos recursos. Com o ajuste fino de um SLM, é possível obter níveis de desempenho mais altos para tarefas específicas sem a mesma sobrecarga computacional necessária para LLMs. Essa comparação destaca a flexibilidade e a eficiência do uso de LLMs e SLMs com base nos requisitos específicos da tarefa.

Por exemplo, no teste de software, o ajuste fino pode permitir que um LLM ou SLM gere casos de teste a partir de histórias de usuários em um formato de saída específico para o contexto da organização. Ao treinar o modelo com as histórias de usuários da organização e os casos de teste correspondentes, o modelo se alinha ao processo de teste e à terminologia específicos da organização.

O ajuste fino de um modelo GenAI para teste de software apresenta vários desafios:

- Evitar resultados tendenciosos ou imprecisos, garantindo o uso de conjuntos de dados de treinamento de alta qualidade e específicos da tarefa.
- Reduzir o excesso de ajuste (o modelo se torna muito especializado nos dados de treinamento, afetando negativamente seu desempenho em dados novos e não vistos) para manter a generalização em diferentes cenários.
- Abordar a opacidade (falta de transparência em como um LLM toma suas decisões ou produz seus resultados) no raciocínio do modelo, o que complica a depuração e a validação
- Gerenciar os recursos computacionais significativos necessários para o processo de ajuste fino (para LLMs).

Objetivo prático 4.2.1 (H0): Observar um exemplo de processo de ajuste fino para uma determinada tarefa de teste e LLM/SLM

Esta demonstração mostra as várias etapas envolvidas no ajuste fino de um LLM para uma determinada tarefa de teste. Ela começa com a seleção de um LLM ou SLM apropriado. Em seguida, é apresentado um conjunto de dados adaptado à tarefa de teste em questão. Em seguida, é mostrada uma solução exemplar para o processo de ajuste fino (por exemplo, uma estrutura de machine learning). Por fim, um prompt é enviado ao modelo ajustado e a qualidade do resultado gerado é discutida.

Essa demonstração do processo de ajuste fino do LLM/SLM para uma tarefa de teste mostra vários aspectos importantes desse processo e aborda, em particular, a qualidade dos dados de treinamento.

4.2.2 LLMOps ao implantar e gerenciar LLMs para teste de software

LLMOps, ou Large Language Model Operations, refere-se ao conjunto de práticas, ferramentas e processos projetados para simplificar o desenvolvimento, a implementação e a manutenção de LLMs em ambientes de produção (Sinha 2024).

O uso da AI generativa nos processos de teste de uma organização pode ser realizado de algumas maneiras diferentes, o que influenciará as decisões a serem tomadas em relação aos LLMOps. Aqui estão três abordagens possíveis:

- Uso de um chatbot de IA: As principais considerações dessa abordagem incluem o gerenciamento dos riscos de privacidade de dados e de segurança e a otimização do custo. As organizações podem usar plataformas de LLM -as-a-Service se as garantias necessárias forem dadas ou implantar uma infraestrutura interna usando LLMs licenciados de código aberto para maior controle. Uma avaliação rigorosa das garantias do fornecedor ou dos recursos internos é fundamental para mitigar os riscos de privacidade e segurança dos dados (consulte a seção 3.2) e garantir a eficiência operacional.
- Usar uma ferramenta de teste com recursos generativos de IA: Essa abordagem tem considerações semelhantes às dos chatbots de IA, como privacidade de dados, segurança e custos operacionais. Além disso, as organizações devem avaliar a segurança dos dados e as garantias de desempenho oferecidas pelo fornecedor da ferramenta de teste. Essas ferramentas de teste normalmente complementam os processos de teste existentes, que exigem uma análise completa de custo-benefício e avaliação de riscos.
- Desenvolvimento interno de uma ferramenta de teste baseada em AI generativa: Essa abordagem enfatiza o controle abrangente dos riscos de privacidade e segurança dos dados, bem como o planejamento cuidadoso dos custos operacionais da IA, como recursos computacionais, armazenamento de dados e treinamento de pessoal. As organizações também devem estabelecer processos estruturados para validar e manter os desenvolvimentos específicos da GenAI. O desenvolvimento de soluções internas requer conhecimento especializado na implementação e implantação de uma infraestrutura de teste com LLM.

Essas abordagens não são mutuamente exclusivas, pois uma organização pode utilizar um chatbot de AI para algumas tarefas e desenvolver ferramentas personalizadas para outras. Assim, elas podem ser implementadas simultaneamente, dependendo das atividades de teste específicas envolvidas. Além disso, elas podem incorporar tecnologias adicionais, como o RAG e o ajuste fino dos LLMs/SLMs, para aumentar a eficácia e a adaptabilidade dos processos de teste com a GenAI.

5 Implementação e Integração da AI Generativa em Organizações de Teste - 80 min.

Palavras-chave

Nenhuma

Palavras-chave específicas da AI generativa

Shadow AI

Objetivos de aprendizagem e objetivos práticos do Capítulo 5:

5.1 Roteiro para adoção de AI generativa em testes de software

GenAI- 5.1.1	(K1)	Relembrar os riscos da Shadow AI
GenAI- 5.1.2	(K2)	Explicar os principais aspectos a serem considerados ao definir uma estratégia de AI generativa para testes de software
GenAI- 5.1.3	(K2)	Resumir os principais critérios para selecionar LLMs/SLMs para tarefas de teste de software em um determinado contexto
HO-5.1.3	(H1)	Estimar os custos recorrentes do uso da AI generativa para uma determinada tarefa de teste
GenAI- 5.1.4	(K1)	Relembrar as principais fases da adoção da AI generativa em uma organização de testes

5.2 Gerenciar mudanças ao adotar a AI generativa para testes de software

GenAI- 5.2.1	(K2)	Explicar as habilidades essenciais e as áreas de conhecimento necessárias para que os testadores trabalhem de forma eficaz com a AI generativa nos processos de teste
GenAI- 5.2.2	(K1)	Relembrar estratégias para cultivar as habilidades de AI nas equipes de teste para apoiar a adoção da AI generativa nas atividades de teste
GenAI- 5.2.3	(K1)	Reconhecer como os processos e as responsabilidades de teste mudam em uma organização de teste ao adotar a AI generativa

5.1 Roteiro para a adoção da AI generativa em testes de software

Uma estratégia de teste com a GenAI deve considerar cuidadosamente os principais aspectos, como os objetivos de teste a serem alcançados, a seleção apropriada do LLM, os problemas com os dados de entrada usados para o prompt e a conformidade com os padrões e regulamentos de IA. Com base nessa estratégia, a organização pode estabelecer um roteiro e monitorar o progresso da integração da GenAI aos processos de teste.

5.1.1 Riscos da Shadow AI

A Shadow AI pode levar a riscos relacionados à segurança, à conformidade e à privacidade dos dados:

- Segurança da informação e privacidade de dados pontos fracos: As ferramentas pessoais de AI podem não ter uma segurança robusta, o que pode levar a possíveis violações de dados.
- Problemas de conformidade e regulamentação: O uso de ferramentas de AI não aprovadas pode levar à não conformidade com os padrões e regulamentos do setor (consulte a Seção 3.4.1), o que pode resultar em consequências legais.
- Propriedade intelectual vaga: O uso de ferramentas de AI com contratos de licenciamento pouco claros pode expor os usuários do LLM a disputas de propriedade intelectual, especialmente se os dados protegidos por direitos autorais forem processados sem a devida autorização.

Uma estratégia e etapas para integrar e implantar a GenAI podem ajudar as organizações de teste a evitarem o risco de Shadow AI.

5.1.2 Principais aspectos de uma estratégia de AI generativa em testes de software

Para implementar com sucesso uma estratégia de GenAI nos testes, as organizações devem considerar cuidadosamente vários fatores-chave para garantir uma integração tranquila e resultados ideais. Isso começa com a definição de objetivos de teste mensuráveis para a GenAI, como o aumento da produtividade do teste, a redução dos ciclos de teste e a melhoria da qualidade do teste. A seleção dos LLMs certos é fundamental (consulte a seção 5.1.3) e deve estar alinhada com esses objetivos de teste, garantindo a compatibilidade com a infraestrutura de teste existente e atendendo aos requisitos de escalabilidade do sistema.

A qualidade dos dados desempenha um papel fundamental, pois a eficácia dos testes realizados com o LLM depende de dados de entrada precisos e relevantes, protegidos por procedimentos robustos de segurança. Portanto, a manutenção de dados de entrada de alta qualidade é fundamental para a obtenção de resultados confiáveis e corretos.

Devem ser oferecidos programas de treinamento abrangentes para garantir que as equipes de teste tenham as habilidades técnicas e éticas necessárias para usar as ferramentas do GenAI de forma eficaz. Além do treinamento, métricas específicas devem ser coletadas para medir a eficácia dos resultados do GenAI (consulte a seção 2.3.1).

Para garantir a conformidade com os padrões regulatórios e a adesão às diretrizes éticas, as organizações devem estabelecer diretrizes de processo para o uso do GenAI, incluindo regras para o uso de dados confidenciais, obrigações de transparência (por exemplo, o que foi gerado usando o GenAI) e portas de qualidade com revisão do testware gerado.

5.1.3 Seleção de LLMs/SLMs para tarefas de teste de software

Há uma grande variedade de LLMs/SLMs, cada um com diferentes recursos funcionais (por exemplo, entrada multimodal, raciocínio), recursos técnicos (por exemplo, tamanho da janela de contexto) e tipos de licenciamento (por exemplo, comercial vs. código aberto). Embora muitos benchmarks estejam disponíveis para avaliar LLMs/SLMs para tarefas como processamento de linguagem natural, geração de código ou análise de imagem, apenas alguns são especificamente voltados para tarefas de teste de software (Wenhan 2024). Portanto, a seleção de LLMs/SLMs para tarefas de teste requer a consideração cuidadosa de vários critérios importantes:

- Desempenho do modelo: Avaliar o desempenho do modelo para as tarefas de teste visadas em relação aos padrões de referência da organização usando métricas como as apresentadas na seção 2.3.1.

- Potencial de ajuste fino: Avalie se é possível e útil fazer o ajuste fino do modelo de linguagem (LLM ou SLM) com dados específicos do domínio para melhorar o desempenho de um determinado caso de uso, aumentando a acurácia e a relevância em contextos especializados.
- Custo recorrente: Considere os custos recorrentes do uso do LLM/SLM, incluindo taxas de licenciamento e despesas operacionais, para garantir que ele se enquadre no orçamento da organização para as tarefas de teste visadas.
- Comunidade e suporte: Escolha modelos com suporte ativo da comunidade e documentação detalhada para ajudar na implementação e na solução de problemas.

Ao avaliar cuidadosamente esses critérios, as organizações de teste podem selecionar um ou mais LLMs/SLMs que atendam às suas necessidades específicas e às restrições organizacionais.

Objetivo prático 5.1.3 (H1): Estimar os custos recorrentes do uso da AI generativa para uma determinada tarefa de teste

Este exercício se concentra em estimar os custos recorrentes do uso da GenAI para uma tarefa de teste específica com base em várias suposições. Isso inclui fatores como o número de tokens nos dados de entrada e saída, os prompts usados e a frequência da tarefa. Os modelos de preços de vários fornecedores de LLM/SLM serão explorados e comparados, incluindo pelo menos uma solução comercial e um modelo licenciado de código aberto.

Esse exercício oferece uma oportunidade de calcular e experimentar os custos recorrentes da GenAI usando condições práticas de teste, ajudando a entender as implicações financeiras de diferentes abordagens e fornecedores.

5.1.4 Fases da adoção da AI generativa em testes de software

A adoção da GenAI em uma organização de testes envolve três fases principais:

1. Descoberta: A primeira fase se concentra na conscientização e no desenvolvimento de capacidades. As atividades incluem o treinamento das equipes de teste nos conceitos da GenAI, fornecendo acesso aos LLMs/SLMs e experimentando os casos de uso iniciais para familiarizar os testadores com a GenAI e criar confiança.
2. Iniciação e definição de uso: Depois que o entendimento básico é estabelecido, a segunda fase se concentra na identificação e na priorização de casos de uso práticos para a GenAI em testes de software. Essa fase inclui a avaliação da infraestrutura de teste com tecnologia LLM, o desenvolvimento de conhecimento especializado e a garantia de alinhamento com as necessidades da organização (consulte a seção 6 **do [ISTQB_CTFL_SYL]**).
3. Utilização e iteração: Nessa fase avançada, as organizações integram totalmente o GenAI em seus processos de teste. O monitoramento contínuo do progresso do GenAI para testes de software e ferramentas relacionadas está em vigor, bem como a medição e o gerenciamento da transformação para garantir benefícios sustentáveis e escalabilidade.

Essas fases podem ser executadas em paralelo para diferentes casos de uso. Por exemplo, a análise do relatório de teste pode estar mais adiantada no roteiro, enquanto a automação de teste está nas fases iniciais. Também é importante reconhecer e abordar as preocupações iniciais, como o medo de perda de emprego, que pode afetar a adoção e o moral da equipe.

5.2 Gerencie as mudanças ao adotar a AI generativa para testes de software

A implementação bem-sucedida da GenAI em uma organização de testes requer uma abordagem estruturada dos processos de gerenciamento de mudanças. Os principais aspectos incluem a criação de habilidades essenciais de GenAI e a evolução das funções de teste tradicionais para adotar processos de teste habilitados para IA. A transformação envolve habilidades técnicas e aspectos organizacionais.

5.2.1 Habilidades e conhecimentos essenciais para testes com AI generativa

A integração bem-sucedida da GenAI nos testes exige o domínio das técnicas de engenharia, a compreensão das janelas de contexto do modelo e o desenvolvimento de métodos de revisão de testes. Os testadores devem combinar o domínio e a experiência em testes com as habilidades de AI para avaliar os testes orientados por LLM em tarefas como geração de casos de teste, análise de relatórios de defeitos e geração de dados de teste.

As principais competências incluem a avaliação dos recursos do LLM, a compreensão das técnicas de refinamento imediato e a avaliação do testware gerado por IA. O conhecimento essencial inclui a compreensão dos riscos inerentes à GenAI, juntamente com a conscientização das estratégias comuns de mitigação. Os testadores devem entender as implicações de segurança de dados do compartilhamento de testware com LLMs, implementar a higienização adequada dos dados (remover ou mascarar informações sensíveis, pessoais ou confidenciais) e seguir as práticas de engenharia imediata de preservação da privacidade de dados. As considerações ambientais incluem otimizar a seleção de modelos e os padrões de uso para reduzir a sobrecarga computacional, selecionar modelos do tamanho certo para as tarefas de teste e equilibrar os benefícios da automação da GenAI com o impacto no custo e no consumo de energia.

5.2.2 Criação de recursos de AI generativa em equipes de teste

Uma abordagem prática é essencial para treinar estrategicamente as equipes de teste na GenAI para testes. Isso inclui praticar com vários LLMs/SLMs, seguir caminhos de aprendizado estruturados e desenvolver gradualmente o know-how por meio do compartilhamento dentro da organização. O foco do treinamento está no desenvolvimento de habilidades práticas por meio de exercícios orientados, aprendizado entre pares e integração gradual da AI nas tarefas diárias de teste.

Os membros da equipe de teste progridem do domínio da criação de prompts básicos para o uso de técnicas mais focadas, como prompts específicos de teste. Um padrão de prompt é um modelo reutilizável para a criação de prompts eficazes para orientar o GenAI em direção a resultados consistentes e confiáveis. As comunidades internas de prática apoiam o compartilhamento contínuo de conhecimento, com reuniões regulares para destacar aplicativos bem-sucedidos do GenAI, discutir desafios e refinar as práticas recomendadas. Essas comunidades promovem o aprimoramento contínuo, compartilhando bibliotecas de padrões rápidos e documentando as lições aprendidas com o GenAI para implementações de testes em projetos e domínios.

5.2.3 Evolução dos processos de teste em organizações de teste habilitadas para IA

A integração do GenAI transforma os processos de teste tradicionais dos testadores e gerentes de teste nas organizações de teste.

Os testadores evoluem de especialistas em design de testes e execução de testes para especialistas em testes assistidos por IA, combinando sua experiência em técnicas de teste com habilidades para orientar e verificar o testware gerado por IA. Suas tarefas de teste se expandem e passam a incluir a revisão do resultado geral baseado em IA, o refinamento do prompt e a manutenção de bibliotecas de prompt específicas do teste.

As responsabilidades dos gerentes de teste são atualizadas para incluir o desenvolvimento de uma estratégia de teste baseada em IA, gerenciamento de risco baseado em AI e monitoramento e controle dos processos de teste baseados em IA. Os gerentes de teste se concentram em equilibrar os recursos humanos e de IA, estabelecendo estruturas de governança de AI para casos de uso e garantindo que suas equipes de teste mantenham as competências tradicionais de teste e a alfabetização em IA. Os gerentes de teste não apenas liderarão os testadores humanos, mas também coordenarão com os agentes de teste com tecnologia GenAI, o que exige novas habilidades de gerenciamento para supervisionar equipes híbridas de pessoas e ferramentas GenAI.

6 Referências

Normas

- **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)

Documentos ISTQB®

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

Glossário

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

Livros

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

Artigos

- (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).

Web Pages

(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.

The previous references point to information available on the Internet and elsewhere. Even though those references were checked at the time of publication of this syllabus, the ISTQB® cannot be held responsible if the references are not available anymore.

7 Apêndice A: Objetivos de aprendizagem/nível cognitivo de conhecimento

Os objetivos específicos de aprendizado que se aplicam a este syllabus são mostrados no início de cada capítulo. Cada tópico do syllabus será examinado de acordo com o objetivo de aprendizado para ele.

Os objetivos de aprendizagem começam com um verbo de ação correspondente ao seu nível cognitivo de conhecimento, conforme listado abaixo.

Nível 1: Lembrar (K1)

O candidato se lembrará, reconhecerá e recordará um termo ou conceito.

Verbos de ação: Recordar, reconhecer.

Exemplos
Recordar os conceitos da pirâmide de testes.
Reconhecer os objetivos típicos dos testes.

Nível 2: Compreensão (K2)

O candidato pode selecionar as razões ou explicações para declarações relacionadas ao tópico e pode resumir, comparar, classificar e dar exemplos do conceito de teste.

Verbos de ação: Classificar, comparar, diferenciar, distinguir, explicar, dar exemplos, interpretar, resumir

Exemplos	Observações
Classifique as ferramentas de teste de acordo com sua finalidade e as atividades de teste que elas suportam.	
Comparar os diferentes níveis de teste.	Pode ser usado para procurar semelhanças, diferenças ou ambos.
Diferenciar teste de depuração.	Procura diferenças entre os conceitos.
Distinguir entre riscos de projeto e de produto.	Permite que dois (ou mais) conceitos sejam classificados separadamente.
Explicar o impacto do contexto no processo de teste.	
Dar exemplos de por que o teste é necessário.	
Inferir a causa raiz dos defeitos a partir de um determinado perfil de falhas.	
Resumir as atividades do processo de revisão do produto de trabalho.	

Nível 3: Aplicar (K3)

O candidato pode executar um procedimento quando confrontado com uma tarefa familiar ou selecionar o procedimento correto e aplicá-lo a um determinado contexto.

Verbos de ação: Aplicar, implementar, preparar, usar

Exemplos	Observações

Aplicar a análise de valor limite para derivar casos de teste a partir de determinados requisitos.	Deve se referir a um procedimento/ técnica/ processo etc.
Implementar métodos de coleta de métricas para dar suporte aos requisitos técnicos e gerenciais.	
Preparar testes de instalação para aplicativos móveis.	
Usar a rastreabilidade para monitorar o progresso do teste quanto à integridade e à consistência com os objetivos, a estratégia e o plano de teste.	Pode ser usado em um LO que deseja que o candidato seja capaz de usar uma técnica ou um procedimento. Semelhante a "aplicar".

Referência

(Para os níveis cognitivos dos objetivos de aprendizagem)

Anderson, L. W. e Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and

Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

8 Apêndice B: Matriz de rastreabilidade entre Resultados de Negócio e Objetivos de Aprendizagem

Esta seção relaciona a rastreabilidade entre os Resultados do Negócio e os Objetivos de Aprendizagem do Certified Tester Testing with Generative AI. Os objetivos práticos não são mencionados nesta tabela, pois cada HO está associado a um único OA. A rastreabilidade entre um HO e um BO é feita por meio do LO ao qual o HO está associado.

Resultados de negócio: Teste de testador certificado com AI generativa			BO1	BO2	BO3	BO4	BO5
GenAI -BO1	Compreender os conceitos fundamentais, os recursos e as limitações da AI Generativa		8				
GenAI -BO2	Desenvolver habilidades práticas para solicitar Large Language Models para testes de software			10			
GenAI -BO3	Obter informações sobre os riscos e as atenuações do uso da AI generativa para testes de software				11		
GenAI -BO4	Obtenha insights sobre os aplicativos de soluções de AI generativa para testes de software					19	
GenAI -BO5	Contribuir efetivamente para a definição e a implementação de uma estratégia e um roteiro de AI generativa para testes de software em uma organização						13

Resultados de negócio: Teste de testador certificado com AI generativa			BO1	BO2	BO3	BO4	BO5
LO exclusivo	Objetivo de aprendizado	Nível K					
1	Introdução à AI generativa para teste de software						
1.1	Fundamentos e conceitos-chave da AI generativa						
GenAI - 1.1.1	Lembre-se dos diferentes tipos de IA: AI simbólica, machine learning clássico, deep learning e AI generativa	K1	X				
GenAI - 1.1.2	Explicar os conceitos básicos de AI generativa e Large Language Model	K2	X				
GenAI - 1.1.3	Distinguir entre fundação, instrução ajustada e raciocínio LLMs	K2	X				
GenAI - 1.1.4	Resumir os princípios básicos dos LLMs multimodais e modelos de visão-linguagem	K2	X				
1.2	Aproveitamento da AI generativa em testes de software: Princípios básicos						
GenAI - 1.2.1	Dê exemplos de recursos do LLM para tarefas de teste	K2	X			X	
GenAI - 1.2.2	Comparar modelos de interação ao usar o GenAI para testes de software	K2	X			X	
2	Engenharia imediata para testes de software eficazes						
2.1	Desenvolvimento eficaz de prompts						
GenAI - 2.1.1	Dê exemplos da estrutura dos prompts usados na AI generativa para testes de software	K2		X			
GenAI - 2.1.2	Diferenciar as principais técnicas de prompting para teste de software	K2		X			

Resultados de negócio: Teste de testador certificado com AI generativa			BO1	BO2	BO3	BO4	BO5
GenAI - 2.1.3	Distinguir entre prompts de sistema e prompts de usuário	K2		X			
2.2	Aplicação de técnicas de engenharia de prompts a tarefas de teste de software						
GenAI - 2.2.1	Aplicar a AI generativa às tarefas de análise de teste	K3		X			
GenAI - 2.2.2	Aplicar a AI generativa para testar o projeto e testar as tarefas de implementação	K3		X			
GenAI - 2.2.3	Aplicar AI generativa a testes de regressão automatizados	K3		X			
GenAI - 2.2.4	Aplicar a AI generativa para tarefas de monitoramento de testes	K3		X			
GenAI - 2.2.5	Selecionar e aplicar técnicas de prompting apropriadas para um determinado contexto e tarefa de teste	K3		X		X	
2.3	Avaliar os resultados da AI generativa e refinar as instruções para tarefas de teste de software						
GenAI - 2.3.1	Compreender as métricas para avaliar os resultados da AI generativa em tarefas de teste	K2		X	X	X	
GenAI - 2.3.2	Dar exemplos de métodos para avaliar e refinar iterativamente os prompts	K2		X	X	X	
3	Gerenciamento de riscos de AI generativa em testes de software						
3.1	Alucinações, erros de raciocínio e vieses						
GenAI - 3.1.1	Relembrar as definições de alucinações, raciocínio erros e vieses em sistemas de AI generativa	K1	X		X	X	

Resultados de negócio: Teste de testador certificado com AI generativa			BO1	BO2	BO3	BO4	BO5
GenAI - 3.1.2	Identificar alucinações, erros de raciocínio e vieses nos resultados dos modelos LLM.	K3			X	X	
GenAI - 3.1.3	Resumir as técnicas de mitigação de alucinações, erros de raciocínio e vieses do GenAI em tarefas de teste de software	K2			X	X	
GenAI - 3.1.4	Relembrar técnicas de mitigação para comportamento não determinístico de LLMs	K1	X		X	X	
3.2	Privacidade de dados e riscos de segurança da AI generativa em testes de software						
GenAI - 3.2.1	Explicar os principais riscos de privacidade de dados e segurança associados ao uso da AI generativa em testes de software	K2			X	X	
GenAI - 3.2.2	Dar exemplos de privacidade de dados e vulnerabilidades no uso da AI generativa em testes de software	K2			X	X	
GenAI - 3.2.3	Resumir as estratégias de mitigação para proteger a privacidade dos dados e aprimorar a segurança na AI generativa para teste de software	K2			X	X	
3.3	Consumo de energia e impacto ambiental da AI generativa em testes de software						
GenAI - 3.3.1	Explicar o impacto das características da tarefa e do uso do modelo no consumo de energia da AI generativa em testes de software	K2			X	X	
3.4	Regulamentos, padrões e estruturas de práticas recomendadas de IA						
GenAI - 3.4.1	Relembrar exemplos de regulamentações, padrões e estruturas de melhores práticas de AI relevantes para a AI generativa em testes de software	K1			X	X	X
4	LLM - Infraestrutura de teste com alimentação para teste de software						

Resultados de negócio: Teste de testador certificado com AI generativa			BO1	BO2	BO3	BO4	BO5
4.1	Abordagens arquitetônicas para o LLM - Infraestrutura de teste com potência						
GenAI - 4.1.1	Explicar os principais componentes e conceitos arquitetônicos da intraestrutura de teste com tecnologia LLM	K2				X	X
GenAI - 4.1.2	Resumir a geração aumentada por recuperação	K2				X	X
GenAI - 4.1.3	Explicar a função e a aplicação dos agentes com tecnologia LLM na automatização dos processos de teste	K2				X	X
4.2	Ajuste fino e LLMOps: Operacionalização da AI generativa para teste de software						
GenAI - 4.2.1	Explicar o ajuste fino dos modelos de linguagem para tarefas de teste específicas	K2				X	X
Ge2AI- 4.2.2	Explicar o LLMOps e sua função na implantação e no gerenciamento de LLMs para tarefas de teste	K2				X	X
5	Implementação e integração de AI generativa em organizações de teste						
5.1	Roteiro para a adoção de AI generativa em testes de software						
GenAI- 5.1.1	Relembrar os riscos da AI sombra	K1					X
GenAI - 5.1.2	Explicar os principais aspectos a serem considerados ao definir uma estratégia de AI generativa para testes de software	K2					X
GenAI- 5.1.3	Resumir os principais critérios para selecionar LLMs/SLMs para tarefas de teste de software em um determinado contexto	K2					X
GenAI - 5.1.4	Relembrar as principais fases da adoção da AI generativa em uma organização de testes	K1					X

Resultados de negócio: Teste de testador certificado com AI generativa			BO1	BO2	BO3	BO4	BO5
5.2	Gerenciar mudanças ao adotar a AI generativa para teste de software						
GenAI - 5.2.1	Explicar as habilidades essenciais e as áreas de conhecimento necessárias para que os testadores trabalhem de forma eficaz com a AI generativa nos processos de teste	K2					X
GenAI- 5.2.2	Relembrar estratégias para cultivar habilidades de AI nas equipes de teste para apoiar a adoção da AI generativa nos processos de teste	K1					X
GenAI - 5.2.3	Reconhecer como os processos e as responsabilidades de teste mudam em uma organização de teste ao adotar a AI generativa para testes	K1					X

9 Appendix C: Release Notes

A versão é a V1.1. As notas de lançamento desta primeira versão estão disponíveis para download no ISTQB.

10 Apêndice D: Termos específicos AI Generativa

Termo original	Termo	Definição
AI chatbot	AI chatbot	Um agente de conversação que usa LLMs para processar consultas e gerar respostas de texto semelhantes às humanas, permitindo a comunicação interativa com os usuários.
Context window	Context window	A extensão do texto, medida em tokens, que um modelo de linguagem considera ao gerar respostas, influenciando a relevância e a coerência de seus resultados.
Deep learning	Deep learning	ML usando redes neurais com várias camadas.
Embedding	Embedding	Uma técnica usada para representar tokens como vetores densos em um espaço contínuo, aprendida durante o treinamento para capturar relações semânticas, sintáticas e contextuais.
Feature	Funcionalidade	Um atributo mensurável individual dos dados de entrada usados para treinamento por um algoritmo de ML e para previsão por um modelo de ML
Few-shot prompting	Prompting de poucos disparos	Uma técnica em que um modelo recebe alguns exemplos dentro do prompt para orientá-lo na geração de respostas adequadas.
Fine-tuning	Ajuste fino	Um processo de aprendizado supervisionado que usa um conjunto de dados de exemplos rotulados para atualizar os pesos do LLM e adaptá-los a tarefas ou domínios específicos.
Foundation LLM	Foundation LLM	Modelos de uso geral pré-treinados em uma ampla gama de dados de texto, capazes de prever a próxima palavra com base em padrões linguísticos aprendidos. Sinônimo: Base LLM
Generative AI (GenAI)	Generative AI (GenAI)	Um tipo de sistema de inteligência artificial que usa modelos de machine learning para gerar (novo) conteúdo intelectual que se assemelha ao conteúdo criado por humanos.
Generative pre-trained transformer (GPT)	Generative pre-trained transformer (GPT)	Um tipo de modelo de deep learning baseado em transformador pré-treinado em grandes quantidades de dados de texto para entender e gerar texto semelhante ao humano.
Hallucination	Alucinação	Informações incorretas criadas por um LLM.
Instruction-tuned LLM	Instruction-tuned LLM	Um LLM básico treinado para seguir instruções, geralmente reforçadas por feedback para estimular respostas corretas.
Large language model (LLM)	Large language model (LLM)	Um programa de computador que usa coleções muito grandes de dados linguísticos para entender e produzir textos de forma semelhante à dos humanos.

Termo original	Termo	Definição
LLM powered agent	Agentes com tecnologia LLM	Um aplicativo que integra o raciocínio, a tomada de decisões e a memória do LLM, usando ferramentas para executar tarefas.
LLMOps	LLMOps	Práticas e ferramentas voltadas para a implantação, o monitoramento e a manutenção de LLMs em ambientes de produção.
machine learning (ML)	machine learning (ML)	O processo que utiliza técnicas computacionais para permitir que os sistemas aprendam com os dados ou com a experiência (ISO/IEC TR 29119-11).
Meta prompting	Meta prompting	A elaboração de instruções de nível superior que geram prompts específicos para explorar ou automatizar recursos.
Multimodal model	Modelo multimodal	Modelos de GenAI que são capazes de processar e gerar conteúdo em vários tipos de dados, como texto, imagens e áudio.
natural language processing (NLP)	natural language processing (NLP)	O processamento de dados codificados em linguagem natural por computadores para recuperar informações e para a representação do conhecimento.
One-shot prompting	Prompting de um disparo	Uma técnica de redação de perguntas em que a pergunta contém um exemplo para orientar a resposta do LLM.
Prompt	Prompt	Uma entrada de linguagem natural fornecida para obter uma resposta específica em AI generativa e Large Language Models.
Prompt chaining	Cadeia de prompt	Uma técnica de prompt que envolve o uso do resultado de um prompt como entrada para outro, criando uma sequência de prompts.
Prompt engineering	Engenharia de prompt	O processo de criação e refinamento de prompts de entrada para orientar os LLMs na produção dos resultados desejados.
Reasoning LLM	Reasoning LLM	Um LLM baseado em modelos ajustados por instruções, refinando sua capacidade de emular processos de raciocínio semelhantes aos humanos
Retrieval-augmented generation	Geração aumentada por recuperação	Uma técnica que combina recursos de LLM com um recuperador para buscar dados relevantes para gerar respostas precisas e contextualmente relevantes.
Shadow AI	Shadow AI	O uso de ferramentas ou sistemas GenAI em uma organização sem aprovação ou supervisão formal.
Small language model (SML)	Small language model (SML)	Modelos de linguagem que são intencionalmente projetados e treinados para serem pequenos, oferecendo um equilíbrio entre eficiência e compreensão de linguagem específica da tarefa.

Termo original	Termo	Definição
Symbolic AI	AI simbólica	Uma abordagem de AI que usa símbolos, regras e conhecimento estruturado para modelar o raciocínio.
System prompt	System prompt	Um conjunto de instruções predefinido, normalmente oculto para os usuários do chatbot, que estabelece consistentemente o contexto, o tom e os limites das respostas de um LLM e orienta seu comportamento durante as interações.
Temperature	Temperatura	Um parâmetro que controla a aleatoriedade ou a criatividade dos resultados de um LLM.
Tokenization	Tokenização	O processo de dividir o texto em unidades menores para processamento por modelos de linguagem.
Transformer	Transformer	Uma arquitetura de modelo de deep learning que utiliza mecanismos de auto atenção para capturar dependências de longo alcance em sequências de entrada.
User prompt	Prompt do usuário	Uma instrução ou consulta inserida por um usuário em um Modelo de Linguagem Ampla (LLM) que direciona a resposta do modelo para cumprir tarefas específicas ou fornecer as informações desejadas.
Vector database	Banco de dados vetorial	Um banco de dados otimizado para armazenar e consultar representações vetoriais de dados de alta dimensão
Vision-language model	Modelo de linguagem visual	Um sistema GenAI que processa conjuntamente dados visuais e textuais para executar tarefas vinculando e gerando conteúdo em ambas as modalidades.
Zero-shot prompting	Prompting de zero disparo	Uma técnica de redação de avisos em que o aviso não contém exemplos, contando com o conhecimento pré-existente do modelo para gerar uma resposta.

11 Apêndice E: Marcas registradas

ISTQB® é uma marca registrada do International Software Testing Qualifications Board

12 Glossário

Todos os termos de teste estão definidos no ISTQB® Glossary (http://glossary.istqb.org/).