

Universidade de Pernambuco

Programa de Pós-Graduação em Engenharia da Computação (PPGEC)

Proposta de Tese de Doutorado

Área: Inteligência Computacional

Título: Um Framework Inteligente para Análise, Correção e Evolução Semântica de Features BDD Baseado em NLP e Modelos de Linguagem

Orientador – Cleyton Mário de Oliveira Rodrigues (cleyton.rodrigues@upe.br)

Coorientador – Wyllyams Barbosa Santos (wbs@upe.br)

Descrição

O Behavior-Driven Development (BDD) é amplamente utilizado para alinhar requisitos de negócio, desenvolvimento e testes por meio de cenários escritos em linguagem natural estruturada, promovendo uma comunicação mais eficaz entre os diferentes stakeholders do processo de software [1]. Apesar de seus benefícios, a adoção de BDD em larga escala enfrenta desafios relevantes relacionados à qualidade, consistência e evolução das features, especialmente quando os cenários são avaliados apenas de forma isolada. Nesse contexto, diversos defeitos emergem apenas no nível da feature, incluindo redundâncias entre cenários, lacunas comportamentais, inconsistências semânticas e violações de boas práticas de especificação, impactando a manutenibilidade e a confiabilidade das suítes de teste automatizadas.

Avanços recentes em Processamento de Linguagem Natural (NLP) e em Modelos de Linguagem de Grande e Pequeno Porte (LLMs e SLMs) ampliaram significativamente o potencial de sistemas capazes de analisar e gerar texto em contextos complexos e dependentes de domínio. Estudos recentes demonstram a viabilidade do uso dessas técnicas para apoiar atividades de engenharia de testes, como a geração automática de casos de teste, análise de especificações e suporte à evolução de artefatos de requisitos e testes [2,3,4]. Entretanto, a aplicação direta de modelos de linguagem no contexto da engenharia de software demanda mecanismos adicionais de controle, explicabilidade e mitigação de alucinações, especialmente em ambientes industriais críticos.

Diante desse cenário, esta pesquisa propõe o desenvolvimento de um framework inteligente para análise, correção e evolução semântica de features BDD, considerando a feature como unidade principal de raciocínio. O framework combinará técnicas baseadas em regras, aprendizado supervisionado e modelos de linguagem para identificar potenciais defeitos em nível de feature, sugerir correções automáticas em cenários Gherkin existentes e propor novos steps ou cenários que completem comportamentos ausentes ou sub especificados. A abordagem adota o paradigma de IA aumentada, mantendo o engenheiro de testes como agente central do processo decisório, apoiado por mecanismos de suporte explicáveis e auditáveis fornecidos pelo sistema.

Referências

1. NORTH, D. *Introducing BDD*. Dan North & Associates, 2006. Disponível em: <https://dannorth.net/blog/introducing-bdd/>. Acesso em: 27 jan. 2026.
2. SANTOS, J. G.; MACIEL, R. S. P. *AutomTest 3.0: an automated test-case generation tool from user story processing powered with LLMs*. In: **SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE (SBES)**, 38., 2024, Curitiba, PR. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. p. 769–774. DOI: <https://doi.org/10.5753/sbes.2024.3603>
3. LIMA, G.; SILVA, D.; MAR, C.; CORONEL, D. *Automated test case generation in a real-world system using a customized AI agent: an experience report*. In: **SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE (SBQS)**, 2025. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2025. p. 376–385. DOI: <https://doi.org/10.5753/sbqs.2025.15010>.
4. CELIK, A.; QUSAY, H. *A review of large language models for automated test case generation*. *Machine Learning and Knowledge Extraction*, v. 7, n. 3, p. 97, 2025. DOI: <https://doi.org/10.3390/make7030097>