VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Typografie a publikování – 4. projekt Umělá inteligence v testování softwaru

13. dubna 2024 Maksim Kalutski

1 Úvod

Proces vývoje softwaru je v zásadě složitý a náchylný k chybám. Umělá inteligence (UI) se stává silným nástrojem pro zefektivnění testování softwaru, zvyšuje efektivitu, přesnost a nakonec kvalitu softwarových produktů. Techniky založené na UI mohou automatizovat opakující se úkoly, identifikovat složité vzory v testovacích datech a generovat nové testovací případy, které by mohly uniknout lidským testerům.

2 Generování testovacích případů

Jednou z klíčových aplikací UI je generování testovacích případů. Algoritmy UI mohou analyzovat existující testovací případy a strukturu kódu, aby navrhly nové, potenciálně účinnější testovací scénáře. [9] Toto rozšíření pomáhá zvýšit pokrytí testů a odhalit chyby, které mohly být přehlédnuty při tradičním ručním testování. [4] Navíc techniky zpracování přirozeného jazyka (NLP), podmnožina UI, mohou analyzovat dokumenty s požadavky a automaticky vytvářet testovací případy, čímž zajišťují soulad mezi specifikacemi a testovacími postupy. [10]

3 Identifikace anomálií a vzorů

Další významnou výhodou testování řízeného UI je schopnost identifikovat anomálie a vzory ve velkých datech. [8] To může být obzvláště užitečné při výkonnostním testování, kde modely UI mohou analyzovat metriky jako dobu odezvy a využití zdrojů, aby předpověděly úzká místa a optimalizovaly výkon softwaru. [3] Analýza chybových zpráv a sledovačů problémů pomocí UI může také identifikovat opakující se nebo kritické defekty, což poskytuje vhledy pro cílená zlepšení. [6]

4 Výzvy při implementaci

Nicméně implementace UI v testování softwaru přináší výzvy. Dostupnost vysoce kvalitních trénovacích dat je nezbytná pro stavbu účinných modelů UI. Organizace mohou potřebovat investovat do pečlivého sběru a přípravy dat. [7] Navíc interpretace výsledků generovaných algoritmy UI může vyžadovat specializovanou odbornost. Pečlivý návrh modelů vysvětlitelné UI se stává zásadní pro zvýšení transparentnosti a důvěry v testovací proces. [2]

5 Závěr

Pro maximalizaci výhod UI v testování softwaru je nezbytný strategický a integrovaný přístup. Klíčové jsou spolupracující týmy s odborností na testování i UI. [5] Mezi nejlepší postupy patří použití UI k doplnění lidského úsudku spíše než k úplnému nahrazení ručního testování. Kontinuální hodnocení a iterativní zlepšování modelů UI zajišťuje, že se přizpůsobí měnícím se potřebám softwaru a testování. [1]

Reference

- [1] Artificial Intelligence Adoption in Software Development. Qualitative study, Lund University, 2023.
- [2] Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. roèník 16, Aug 24, 2023: s. 45–74, doi:10.1007/s12559-023-10179-81.
- [3] Anomaly detection optimization using big data and deep learning to reduce false-positive1. *Journal of Big Data*, roèník 7, è. 68, Aug 31, 2020, doi:10.1186/s40537-020-00346-1.
- [4] Defect Prediction Guided Search-Based Software Testing. arXiv.org, Sep 21–25, 2020. Dostupné z: https://arxiv.org/pdf/2109.12645.pdf
- [5] Alfadel, K.: How AI Is Changing Software Testing Forever?

 Dostupné z: https://www.valuecoders.com/blog/qa-testing/how-ai-is-changing-software
- [6] Haddad, M.: How AI Can Detect Software Bugs and Root Cause Analysis. March 29, 2023.

 Dostupné z: https://blog.railtown.ai/blog-posts/how-ai-can-detect-software-bugs-ar
- [7] Morozov, N.: Testing and Artificial Intelligence. Litres, 2024, ISBN 5046162199, 22–24 s.
- [8] Mäkelä, M.: *Utilizing Artificial Intelligence in Software Testing*. Qualitative study, Metropolia University of Applied Sciences, 2019.
- [9] Olsthoorn, M.: *More Effective Test Case Generation with Multiple Tribes of AI*. TU Delft Research Portal, 2022, ISBN 9781665495981, 286–290 s.
- [10] Paul, J.: Natural Language Processing (NLP) in Software Testing: Automating Test Case Creation and Documentation. Mar 30, 2023.
 - $Dostupn\'e\ z: \texttt{https://dzone.com/articles/natural-language-processing-nlp-in-software} and the processing and the processing$