Návrh a optimalizace technických specifikací pro autonomní vývojové agenty

Motivace

S nástupem pokročilých modelů strojového učení pro zpracování textu získáváme možnost automaticky generovat počítačový kód. Programování se stává dostupnějším a vývoj zdánlivě rychlejším a efektivnějším. Z dlouhodobého hlediska však ztrácíme nad generovaným kódem kontrolu. Vývojáři přicházejí o kontext a hlubokou znalost kódové základny (codebase), zatímco velké jazykové modely (LLM) jsou limitovány omezenou pamětí (context window). Přidáváme nové funkce, zatímco jiné části systému přestávají fungovat, aniž bychom znali příčinu. Z programátorů se stávají systémoví architekti, neboť samotné psaní kódu je rychle automatizováno.

Vyvstává proto klíčová otázka: Jaké podstatné podklady (specifikace) je nutné připravit, aby LLM dokázaly generovat kvalitní a udržitelný kód i v situaci, kdy rozsah projektu narůstá? A je model schopen si tyto specifikace sám vytvořit či domyslet?

Cíle práce

Jaká je taxonomie softwarového inženýrství a které artefakty jsou klíčové pro řízení LLM? Co tvoří podstatný kontext pro LLM a jak s ním model pracuje, jaké metody/způsoby LLM dávají lepší výsledky? V jakých případech LLM selhávají a co je příčinou?

Metodika

1. Rešerše zdrojů k pochopení současných přístupu s práci s LLM v contextu softwarového inženýrství
2. Návrh testovacího prostředí a testovacích případů.
3. Spuštění experimentů (testovacích případů).
4. Analýza a interpretace výsledků.

Vymezení rozsahu

- testování různých LLMs
- tvorba produkčního nástroje/produktu
- testování různých programovacích jazyků
- podrobný research machine learning algoritmů

Literatura

- **GITHUB NEXT.** *Spec Kit: Spec-Driven Development for AI Agents* [online]. 2024 [cit. 2025-11-30]. Dostupné z: https://github.com/github/spec-kit
- **BMAD-CODE-ORG.** *BMAD METHOD: Breakthrough Method for Agile AI-Driven Development* [online]. GitHub, 2025 [cit. 2025-11-30]. Dostupné z: https://github.com/bmad-code-org/BMAD-METHOD
- **ANTHROPIC.** *Effective Harnesses for Long-Running Agents* [online]. Anthropic Engineering Blog, 2024 [cit. 2025-11-30]. Dostupné z: https://www.anthropic.com/engineering/effective-harnesses-for-long-running-agents
- **METR.** *Model Evaluation and Threat Research* [online]. 2024 [cit. 2025-11-30]. Dostupné z: https://metr.org/
- **JIMENEZ, Carlos E. et al.** *SWE-bench: Can Language Models Resolve Real-world Github Issues?* [online]. In: The Twelfth International Conference on Learning Representations (ICLR). 2024 [cit. 2025-11-30]. Dostupné z: https://arxiv.org/abs/2310.06770
- **LIU, Nelson F. et al.** *Lost in the Middle: How Language Models Use Long Contexts* [online]. arXiv preprint arXiv:2307.03172. 2023 [cit. 2025-11-30]. Dostupné z: https://arxiv.org/abs/2307.03172
- **VASWANI, Ashish et al.** *Attention Is All You Need* [online]. Advances in Neural Information Processing Systems, 2017 [cit. 2025-11-30]. Dostupné z: https://arxiv.org/abs/1706.03762
- **WEI, Jason et al.** *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models* [online]. Advances in Neural Information Processing Systems, 2022 [cit. 2025-11-30]. Dostupné z: https://arxiv.org/abs/2201.11903
- **KHONONOV, Vlad.** *Learning Domain-Driven Design: Aligning Software Architecture and Business Strategy*. 1. vyd. O'Reilly Media, 2021. ISBN 978-1098100131.
- **IEEE COMPUTER SOCIETY.** *SWEBOK: Guide to the Software Engineering Body of Knowledge*. Version 3.0. IEEE, 2014. Dostupné z: https://www.swebok.org/
- **BROWN, Simon.** *The C4 model for visualising software architecture* [online]. 2024 [cit. 2025-11-30]. Dostupné z: https://c4model.com/