**Párové programování s AI v porovnání s lidským párovým programováním**

**Softwarové inženýrství**

Marie Kleckerová

13. 5. 2025

# Úvod do tématu

Párové programování je technika vývoje softwaru, při níž dva programátoři spolupracují na jednom počítači. Jeden z nich, označovaný jako ,,řidič“ (driver), píše kód, zatímco druhý, ,,navigátor“ (navigator), sleduje každý řádek kódu, který je psán, a současně přemýšlí o strategickém směru práce. Díky této spolupráci si programátoři předávají vlastnosti a dochází ke zlepšení kvality kódu a rychlejšímu odhalování chyb. Párové programování je klíčovou součástí extrémního programování (XP) a dalších agilních metodik vývoje softwaru [1].

Mezi hlavní výhody párového programování patří vyšší kvalita kódu, rychlejší odhalení chyb, efektivní sdílení know–how a lepší týmová komunikace. Na druhou stranu může být tato metoda náročnější na čas a vyžaduje dobrou spolupráci mezi partnery – v praxi často narážíme na rozdíly v dovednostech dvou programátorů nebo na organizační obtíže.

S nástupem umělé inteligence se objevily nástroje, které umožňují podobnou spolupráci s AI asistentem. GitHub Copilot, Amazon CodeWhisperer nebo Chat GPT využívají velké jazykové modely (LLM), trénované na velkém množství veřejně dostupného kódu, a poskytují návrhy kódu v reálném čase přímo v integrovaném vývojovém prostředí (IDE) [3]. AI asistenti mají schopnost zrychlit psaní kódu, snížit kognitivní zátěž programátora a pomoct hlavně s rutinními úkoly. To může významně zvýšit produktivitu vývojářů. AI nástroje však nejsou náhradou za lidskou spolupráci ani know–how, ale spíše doplňkem, který může podpořit efektivitu a kreativitu vývojářů.

Párové programování s AI je aktuální téma pro softwarové inženýrství, protože reflektuje aktuální snahu o automatizaci vývoje a využívání jazykových modelů v programování. Porovnání tradičního párového programování s lidským partnerem a párového programování s AI asistentem nám může poskytnout cenné poznatky o výhodách a nevýhodách obou přístupů a pomoci vývojářům lépe porozumět, kdy a jak využívat AI nástroje ve své práci.

Motivací pro zkoumání tohoto tématu je osobní zkušenost s párovým programováním a zájem zjistit, jak efektivně využít AI asistenty při práci na školních i osobních projektech. Cílem je porovnat efektivitu a subjektivní vnímání spolupráce při párovém programování s lidským partnerem a s AI nástrojem.

# Výzkumná rešerše

Čím dál více se řeší, jestli AI asistenti zvládnou plnohodnotně nahradit lidského parťáka při párovém programování, a jaký má volba vliv na efektivitu a uživatelskou zkušenost.

Tradiční párové programování je v odborné literatuře popisováno jako účinná metoda pro zlepšení kvality kódu, rozvoj týmové spolupráce a snížení chybovosti díky průbežné kontrole práce v reálném čase [1]. Na druhou stranu studie často zmiňují i nevýhody, jako jsou příliš velké rozdíly v dovednostech partnerů nebo organizační složitost domlouvání společných sezení [2].

V posledních letech se do tohoto procesu stále častěji zapojují AI nástroje, jako je GitHub Copilot, Amazon CodeWhisperer nebo ChatGPT. Tito asistenti využívají velké jazykové modely (LLM) k automatickému generování kódu. Vývojářům pomáhají zejména při řešení rutinních úkolů, takže šetří čas a kognitivní zátěž.

Studie Pohanka et al. (2025) přinesla komplexní srovnání mezi třemi formami programování: individuálním, tradičně párovým (člověk–člověk) a párovým s AI asistentem. Ukázalo se, že **AI asistenti významně snižují programátorskou úzkost**, a to především díky své „neodsuzující“ povaze. Studenti se nebojí dělat chyby, neboť nemají strach ze společenského hodnocení. Tento efekt měl přímý vliv na zlepšení výkonu, zejména u méně zkušených účastníků [5].

Z hlediska **motivace**a**vnímání autonomie**AI partner podporoval u studentů větší pocit kompetence – poskytoval okamžitou zpětnou vazbu a umožnil jim objevovat řešení vlastním tempem. V tom se ukázal jako vhodný nástroj zejména pro začátečníky [5]. Nicméně v oblastech jako je **sociální přítomnost a vnímání spolupráce**tradiční lidské párování jednoznačně vedlo. Účastníci vnímali lidské partnery jako více podporující a oceňovali možnost verbální interakce, sdílení nápadů či společného ladění strategie [5].

Empirické studie, například Zimmermann et al. (2023), ukazují, že vývojáři při spolupráci s AI pociťují nižší míru stresu díky okamžité zpětné vazbě a absenci hodnotícího postoje ve formě programovacího partnera [3]. Na druhou stranu však postrádají hlubší diskuzi a kreativní brainstorming, které při spolupráci s člověkem vznikají přirozeně. Pohanka et al. (2025) tvrdí, že ačkoliv AI asistenti mají pozitivní vliv na pocit autonomie a kompetence programátorů, nedokážou nahradit sociální dimenzi lidské spolupráce [2]. Amazon CodeWhisperer přináší podobné výsledky – podporuje efektivitu při repetitivních úlohách, avšak chybí mu schopnost skutečné kolaborace [4].

Z toho, co zatím víme, se zdá, že AI asistenti představují cenný nástroj pro podporu produktivity a snížení programátorské úzkosti, avšak z hlediska socializace a sdílení know–how zůstává lidské párové programování stále nenahraditelné.

# Výzkumné otázky

Cílem tohoto výzkumu je porovnat zkušenost a efektivitu práce při párovém programování s lidským partnerem a s AI asistentem. Důraz bude kladen nejen na objektivní měřitelné výsledky, ale také na subjektivní introspektivní zkušenost. Výzkum je zaměřen na tři hlavní otázky:

* VO1: Jaký je rozdíl v efektivitě (čas, počet iterací, kvalita kódu) mezi párovým programováním s AI a s lidským asistentem?
* VO2: Jak se liší subjektivní vnímání programátorské úzkosti, motivace a míry kolaborace při párovém programování s AI oproti lidskému partnerovi?
* VO3: Do jaké míry dokáže AI kompenzovat kolaborativní benefity lidského partnera při párovém programování z pohledu uživatelské zkušenosti?

Pro účely experimentu bude jako AI asistent využit GitHub Copilot. Práce bude probíhat ve vývojovém prostředí Visual Studio Code. Všechny úlohy budou implementovány v programovacím jazyce Python.

Lidským partnerem v experimentu bude David Král, student oboru aplikovaná informatika, se srovnatelnou úrovní programátorských dovedností jako autorka.

Detailní popis úloh, měřených metrik a způsobu provedení je uveden v následující kapitole.

# Použitá literatura

1. HOREJŠEK, Jan. O čem je párové programování [online]. 2023 [cit. 2025-05-13]. Dostupné z: https://blog.horejsek.com/o-cem-je-parove-programovani/
2. POHANKA, Martin, et al. Pair Programming in STEM Education: Benefits and Challenges. STEM Education Journal [online]. 2025, 6(1), 1–14 [cit. 2025-05-13]. Dostupné z: https://stemeducationjournal.springeropen.com/articles/10.1186/s40594-025-00537-3
3. ZIMMERMANN, Thomas, et al. Pair Programming with AI Assistants: How Developers Collaborate with Copilot and ChatGPT. ACM CHI Conference on Human Factors in Computing Systems [online]. 2023 [cit. 2025-05-13]. Dostupné z: https://arxiv.org/abs/2306.05153
4. JAIN, Ananya, et al. Human-AI Pair Programming: Empirical Insights from Industry Projects. IEEE Transactions on Software Engineering [online]. 2024 [cit. 2025-05-13]. Dostupné z: <https://arxiv.org/abs/2402.01820>
5. POHANKA, Martin, et al. The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. STEM Education Journal [online]. 2025, 6(1), 1–14 [cit. 2025-05-13]. Dostupné z: https://stemeducationjournal.springeropen.com/articles/10.1186/s40594-025-00537-3