**Časovni plan in osnutek diplomskega dela: Hiter inženiring pri razvoju programske opreme**

<https://www.one-tab.com/page/aWWGxjAYTsWsJeeGtf3AMw>

Časovni plan:

31. 1. -> Podrobnejši osnutek poglavij, podrobno proučena literature, načrt za izpeljavo anketiranja učencev

15. 2. -> Narejen uvod, začetna poglavja

31. 2. -> Narejena poglavja, povezana s preučeno literaturo

15. 3. -> začetek dela na praktičnem izdelku

31. 3. -> glavna poglavja končana, prvi osnutek teoretičnega diplomskega dela

15. 4. -> osnutek izdelka

30. 4. -> dopolnitve teoretičnega dela in izdelka

Prvi teden maja (ob soglasju otrok in šefa): eksperiment učenja programiranja pri mlajših otrocih z AI asistenti

21. 5. -> teoretični del pripravljen za pregled in lektoriranje, narejen zaključek

31. 5. : Končan teoretični del, končan izdelek

Osnutek:

**UVOD**

Kaj so AI asistenti, primeri uporab pri programiranju, zakaj se uporabljajo, cilji diplomske naloge.

**AI asistenti splošno**

Nastanek in popularnost AI asistentov, zgodovina, uporaba v različnih industrijah.

**Hitro programiranje**

Uporaba AI asistentov pri programiranju, najbolj pogosti primeri uporabe, največje prednosti in največje pomanjkljivosti+

**ChatGPT podrobno**

Preučevanje prednosti in slabosti, ocena pravilnosti in razlage rešitve, primerjava z ostalimi AI asistenti.

**githubCopilot podrobno**

Preučevanje prednosti in slabosti, ocena pravilnosti in razlage rešitve, primerjava z ostalimi AI asistenti.

**intelliJassistant podrobno**

Preučevanje prednosti in slabosti, ocena pravilnosti in razlage rešitve, primerjava z ostalimi AI asistenti.

**Izgradnja lastnega AI asistenta**

**Načrtovanje**

Cilji lastnega AI asistenta, kako bom dosegel cilje, kako se bom lotil dela.

**Izdelava**

Koraki izdelave lastnega AI asistenta

**Ocena**

Ocena lastnega AI asistenta, ocena doseženih ciljev, primerjava z obstoječimi AI asistenti.

**Učenje programiranja pri mlajših otrocih z AI asistenti \*(če bo čas in če dobim dovoljenja)**

Preučevanje hitrosti, težavnosti in pravilnosti učenja z AI asistenti v primerjavi s tradicionalnim učenjem brez AI pomoči. Primerjava stopnje naučenega.

**Zaključek**

Podrobno o ugotovitvah diplomske naloge, preučeni cilji, ki so bili postavljeni na začetku.

Preglod diplomskega dela: Hiter inženiring pri razvoju programske opreme

AI asistenti so postali zelo pogosto uporabljeno orodje v programskem inženirstvu, ki pomagajo tako začetnikom napisati preprosto kodo, kot tudi izkušenejšim programerjem pri zahtevnejših nalogah olajšajo delo in povečajo produktivnost. Po anketi, ki jo je sestavila znana stran StackOverflow, kar 44% programerjev redno uporablja AI orodja v razvojnem procesu, dodatnih 25% pa to načrtuje v bližnji prihodosti (n= 89 184). Tekom diplomskega dela se bom osredotočil na najbolj uporabljena AI orodja: chatGPT, githubCopilot in intelliJ AI asistant. Raziskoval bom, v katerih področjih so najbolj uporabna, kakšna je hitrost in pravilnost napisanih programov z in brez uporabe AI orodij in kakšne so potencialne slabosti in nevarnosti uporabe AI orodij. Uporabo AI asistentov pri učenju programiranja bom tudi preučil na več skupinah mlajših otrok, ki se učijo programiranja. Otroci so stari 10-14 let in imajo z redkimi izjemami manj kot eno leto izkušenj s programiranjem. Programirajo v programskih jezikih python ter spletno programiranje s stackom html, css, javascript. Preučeval bom hitrost učenja, pravilnost in razumevanje kode, ki jo dobijo z uporabo AI asistentov.

Kljub njihovi popularnosti, so današnja AI orodja v zgodnji dobi razvoja in imajo veliko prostora za izboljšave. Pomankljivosti AI orodij, ki so na voljo bom najprej podrobno preučil, nato pa bom napisal svojo razširitev za najbolj popularno razvojno okolje: VisualStudioCode, ki bo področja problemov poskušala izboljšati. Trenutna hipoteza je, da so največje pomanjkljivost trenutnih AI orodij izmišljena koda, v primeru da orodje najde rešitve (npr. izmišljene metode), premalo možnosti za personalizacijo asistentov (npr. nastavitve glede na željen stil, komentiranje kode) in dejstvo, da uporabniki pišejo neoptimalne kodne pozive (prompts). S svojim razširitvijo bom probal te problematike nasloviti in razviti aistenta, ki bo bolj prijazen za uporabnika in si ne bo izmišljeval kode, ki ne obstaja. Svoj model bom ocenil s podobnimi kriteriji, kot bom ocenjeval že obstoječe probleme, torej kako hitro in učinkovito bo rešil programerski problem, bolj pozorno pa bom tudi opazoval lahkoto uporabe ter resničnost pridobnljene kode. Svojo razširitev bom tudi primerjal z že omenjenimi AI asistenti.

Hiter inženiring je relativno nova zadeva, ki se jo še vedno intenzivno preučuje. Večino literature in raziskav o tej temi je zato iz zadnjih nekaj let. Izhajal bom iz že narejenih raziskav o hitrem inženiringu. Najbolj sorodna najdena raziskava je: Burak Yetiştiren (2023): Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT, ki se osredotoča na kvaliteto in pravilnost kode pri različnih AI asistentih. Raziskava preizkuje pravilnost in efektivnost kode, v mojem delu pa bom preučeval tudi druga področja uporabe, npr. učenje ter razlaganje kode z AI orodji. V svojem delu se bom bolj posvetil tudi pomankljivostim, ki jih bom poskusil tudi izboljšati s svojo razširitvijo v VisualStudioCode. Preučil bom tudi : Arghavan Moradi Dakhel (2023): GitHub Copilot AI pair programmer: Asset or Liability? Ki podrobno in natančno analizira github copilot, njegove predostni in slabosti. V tej raziskavi se tudi primerja AI rešitev s človeškimi, ne poda pa rešitve problemov, ki jih delo najde. Moja rešitev bo dodatno razvila svojo verzijo AI asistenta, ki bo upoštevala že znane slabosti github copilota. Uporabil bom tudi: James Finnie-Ansley (2022) The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming, kjer je podrobneje preučeno programiranje s chatGPT, zmožnost reševanja lažjih programerskih problemov in testov, ter implikacije hitrega inženiringa. V svoji nalogi bom dodatno chatGPT generacijo kode primerjal tudi z drugimi AI asistenti, kar omenjeno delo ne naredi. Pregledal bom tudi potencialne nevarnosti, navedene v delu (2022) : Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions.

VIRI: https://survey.stackoverflow.co/2023/#section-sentiment-and-usage-ai-tools-in-the-development-process : StackoverFlow survey https://arxiv.org/abs/2304.10778 : Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT https://arxiv.org/abs/2304.10778: GitHub Copilot AI pair programmer: Asset or Liability? https://dl.acm.org/doi/10.1145/3511861.3511863 : The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming https://nyuscholars.nyu.edu/en/publications/asleep-at-the-keyboard-assessing-the-security-of-github-copilots- : Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions

<https://www.one-tab.com/page/aWWGxjAYTsWsJeeGtf3AMw>