

Tvorba SW pomocou umelej inteligencie *

Ľubomír Novotný

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xnovotny11@stuba.sk

1. október 2021

Abstrakt

Zaoberáme sa implementáciou umelej inteligencie pri tvorbe a testovaní SW a hlavne GUI. Použitie umelej inteligencie pri vývoji SW a GUI ma viacero výhod. Umelá inteligencia dokáže optimalizovať a zrýchliť, zlacniť, uľahčiť a zautomatizovať testovanie a tvorbu SW a GUI. Použitie umelej inteligencie pri tvorbe SW ma taktiež nevýhody. Nedokáže nahradiť manuálne testovanie, závislé od kvality AI, AI nemá imagináciu a preto nedokáže hľadať chyby ako človek. Umelú inteligenciu môžeme zakomponovať aj do samotných aplikácií a programov pomocou ktorých SW tvoríme. Umelá inteligencia taktiež môže byť použitá na vytvorenie veľkého množstva modelov GUI, ktoré ďalšia umelá inteligencia simulovaného používateľa porovná.

1 Úvod

2 Softvérové inžinierstvo

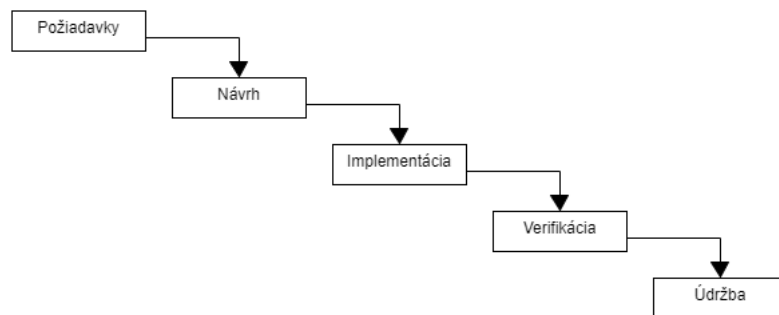
Softvérové inžinierstvo sa nezaobrá iba tvorbou softvérových systémov, ale zahŕňa aj tvorbu softvéru efektívnym spôsobom. Keby sme mali neohraničené zdroje, väčšina problémov s tvorbou softvéru by sa dala vyriešiť. V skutočnosti však treba vytvoriť softvér podľa požiadaviek s ohraničenými zdrojmi a určeným časom ukončenia. Softvérové inžinierstvo sa zaoberá tvorbou väčších softvérových systémov čo najefektívnejším spôsobom. Základným cieľom je:

- zlepšenie akosti softvéru, najmä čo sa týka spoľahlivosti a
- zvýšenie produktivity softvérových inžinierov.

Rozširovanie oblastí použitia počítačov viedlo k riešeniu čoraz zložitejších úloh, pričom návrh a vývoj programov sa zakladal v podstate na ad hoc technikách. V procese tvorby softvéru bolo málo štandardizácie a teda s každým programom sa vymýšľalo už vymyslené. Dôsledkom je zlyhanie celého radu softvérových projektov, predĺženie mnohých z nich. Aj tie softvérové systémy, ktoré sa vytvorili, boli

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Ing. Vladimír Mlynarovič

často nespoľahlivé, vyskytovali sa mnohé nepredvídané poruchy, boli slabo dokumentované, nespĺňali požiadavky, ktoré sa na ne kládli. Na riešenie uvedených problémov bolo treba prehodnotiť všetky oblasti tvorby softvéru. Dôsledkom je vznik softvérového inžinierstva v súvislosti s diskusiami o tzv. softvérovej kríze (konferencie NATO v rokoch 1968- 1969), keď sa prvýkrát otvorene poukázalo na metódy spojené s tvorbou programov.



Obr. 1 Vodopádový model tvorby softvéru.

2.1 Testovanie softvéru

V súčasnosti môžeme konštatovať, že proces testovania je opisovaný v mnohých publikáciách, avšak hlavne vychádza z praktických skúseností veľkých softvérových firiem. Pri samotnom procese testovania sa softvéroví inžinieri sústreďujú na celý životný cyklus testovania, ktorý začína testovaním funkcií a končí preberacím testovaním a zahŕňa [1, 4, 5]:

- Testy funkcií a modulov, ktoré sú vykonávané prevažne softvérovými inžiniermi priamo v etape implementácie.
- Integračné testy, čo je vlastne testovanie viacerých modulov súčasne.
- Regresné testovanie – testuje sa, či nenastal vedľajší efekt pridaním nového modulu alebo funkcie (zavlečenie chyby).
- Nezávislé testy - vykonávané nezávislými externými subjektami.
- Alfa a beta testovanie - čo je vlastne testovanie systému v reálnom prostredí. Pri alfa testovaní sa systém testuje bez živých dát. Testuje ho zákazník u vývojára. Beta testovanie používa reálne dáta so sledovaním výsledkov s možnosťou okamžitej nápravy.
- Systémové testovanie – je to séria rôznych testov, ktorá preveruje celý systém (HW, SW prostredie, databáza, ľudia, ...). Môžeme sem zaradiť aj testovanie obnovy (Recovery), bezpečnostné testovanie, výkonnostné a záťažové testovanie (Stress). Obdobne sem môžeme zaradiť aj tzv. testovanie citlivosti (Sensitivity testing), ktoré sa snaží objaviť kombinácie dát (v rámci platných obmedzení), ktoré môžu spôsobiť nestabilitu systému.
- Inštalčné testy - zahŕňajúce všeobecnú výkonnosť systému, ktorý je prvýkrát nainštalovaný na konkrétnom HW a operačnom systéme.

- Validačné testovanie – slúži k overeniu, že softvér spĺňa „rozumné očakávania“ zákazníka, ktoré sú definované v špecifikovaných požiadavkách. Validačné testovanie sa vykonáva metódami black-box.
- Preberacie testovanie - je to vlastne posledný míľnik pri testovaní projektu. V prípade úspešného zvládnutia nastáva oficiálne prevzatie projektu zákazníkom. Chyby môžu byť do aplikácie vnesené v každom štádiu životného cyklu vývoja aplikácie, vrátane testovania.

3 Umelá inteligencia

Umelá inteligencia označuje disciplínu v rámci počítačovej vedy, v ktorej umelo vytvorená entita môže napodobňovať ľudské kognitívne funkcie tým, že má schopnosť učiť sa a riešiť problémy. Stroj/program sa správa tak, že ak by to robil človek, považovali by sme ho za inteligentného (smart). Okrem rozpoznávania reči, obrazov, prekladu do jazykov dokáže imitovať rozmýšľanie ľudí - učiť sa, uvažovať, riešiť problémy, sám hľadať spôsoby, ako sa dostať k cieľu. Je to aj schopnosť počítačov (agentov) používať algoritmus na učenie sa z dát a získané vedomosti použiť na rozhodovanie podobné ľudskému. Umelá inteligencia ako celok stavia vo veľkej miere na základoch mnohých ďalších vedných odborov, a to predovšetkým na informatike, matematike, štatistike, logike, lingvistike či neurovedách. Podobnosť umelej a ľudskej inteligencie je ale iba povrchná a nie je správne ich stotožňovať. Koncepcia ľudskej inteligencie je stále predmetom výskumu, je to heterogénna entita (logická, emocionálna, sociálna inteligencia...). Vhodné je porovnanie s lietadlom („umelý vták“), ktoré ale prebralo od vtákov iba lietanie. Lietadlo nemá krídla, nebuduje hniezda a neznáša vajcia. Podobne sa treba (zatiaľ) pozeráť na „umelú inteligenciu“, napr. čo do schopnosti chápať kontext situácie, zdôvodňovať, prejavíť empatiu, získať skutočnú múdrosť, byť kreatívny alebo mať vedomie. Vzhľadom na túto iba vzdialenú podobnosť viacerí autori dávajú prednosť pojmu rozšírená inteligencia (Augmented Intelligence), ktorá pomáha pri rozhodovaní ľuďom.

4 Softvér používajúci umelú inteligenciu

Veľmi dôležitá poznámka. Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

5 Používanie AI pri tvorbe SW

kafnisajklmkdfs,

6 Záver