

Softvér v automobilovom priemysle*

Peter Zimmermann

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xzimmermann@stuba.sk

11. október 2021

Abstrakt

... Automobilový priemysel stále napriekuje a každým rokom sa dáva čoraz väčší dôraz na softvér automobilov, ako na samotné mechanické vlastnosti alebo diely vozidiel. Pri vývoji automobilových softvérov je kľúčové využívať vopred stanovené postupy, ktoré pozostávajú z budovania systému pre daný softvér, jeho samotný vývoj a následná prevádzka. Realizáciu týchto postupov zabezpečujú modely, s ktorými je realizácia intenzívnejšia, keďže dopredu poznáme všetky fázy (V-Model), alebo model, ktorý nemá vopred stanovené pevné fázy, takže je viac otvorený a kreatívnejší (Agile Model). Jednou z najdôležitejších činností je testovanie už vyvinutého softvéru, ktorým treba zistiť, či je prevádzka daného softvéru plne bezpečná a spoločne. Taktiež je v článku ukázane, ako dokáže moderný softvér v automobiloch modelovať okolité objekty ako autá, budovy, alebo chodcov.

1 Úvod

Softvér v automobilovom priemysle je téma, ktorá začína byť priamoúmerne s každým rokom čoraz viac dôležitejšia, keďže moderné vozidlá v čoraz väčšej miere závisia od ich softvéru.

Mojim hlavným cieľom tohto článku je oboznámiť čitateľa s informáciami ohľadom softvéru v automobilovom priemysle a taktiež si vo väčšom rozmere rozšíriť moje doterajšie poznatky.

Prvá téma, ktorej som sa venoval, bola "Postupy vo vývoji automobilového softvéru". 2 Táto téma slúži na vysvetlenie samotných postupov.

Druhou z tém bola téma s názvom "Modely pri vývoji automobilového softvéru". 3 V tejto téme som bližšie priblížil, čo vo všeobecnosti sú tieto modely a prečo sú dôležité. Táto téma obsahuje viacero podkapitol, ktoré bližšie opisujú jednotlivé modely. 3.1 3.2

Ďalšou problematikou je "Testovanie"4 už vyvinutého softvéru, ktoré patrí medzi jedno z najdôležitejších činností pri vývoji softvéru pre automobily.

Posledná téma, ktorej som sa venoval bola "Modelovanie nebezpečenstva a predikcia možnej autonehody". 5. Táto téma môže byť veľmi zaujímavá aj pre

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Ing. Fedor Lehocki, Phd.

nezainteresovaných ľudí, keďže s touto funkciou sa už mohol ktokoľvek stretnúť pri riadení moderného automobilu.

Na konci článku je samotný záver, ktorý hodnotí celkové časti článku, jeho dôležité fakty a môj osobný pohľad na túto problematiku. Za záverom nasleduje literatúra, z ktorej som čerpal pri písani tohto článku.

2 Postupy vo vývoji automobilového softvéru

V tejto sekcii sa primárne zameriavam na tri hlavné postupy pri vývoji automobilového softvéru.

1. Budovanie systému
2. Vývoj softvéru
3. Prevádzka

Prvý z postupov procesu je budovanie systému. V tomto postupe sa výrobcovia originálnych zariadení obvykle zaoberajú požiadavkami, ktoré sú primárne zamerané na funkčnosť zariadenia, ktoré bude najviac vyhovovať stanoveným požiadavkám (npr. riadiaca jednotka, hardvér palubného počítača). Títo výrobcovia sa vo väčšine prípadov riadia architektúrou automobilového open systému s názvom „AUTOSAR“. AUTOSAR je v dnešnej dobe celosvetové vývojove partnerstvo v oblasti softvéru a elektroniky v automobiloch.

Druhým z postupov je vývoj samotného softvéru, ktorého hlavným cieľom je vytvoriť viaceru programovacích kódov, z ktorých sa vyberajú tie najvhodnejšie, ktoré budú najlepšie pracovať s už zvolenými zariadeniami. Aby sa dokázali rozlísiť rôzne úrovne zložitosti softvéru, možno použiť dva najbežnejšie modely a to V-Model a Agile model.

Posledným postupom je samotná prevádzka. Znamená to, že každý softvér je aj po vyrobení a uvedení do predaju udržiavaný aktualizáciami, aby prevádzka a funkčnosť samotného softvéru bola čo najdlhšia a najefektívnejšia. [3]

3 Modely pre vývoj automobilového softvéru

Táto sekcia slúži na priblíženie dvoch široko známych modelov vo vývoji softvérov pre automobily.

3.1 V-Model

V-Model je v dnešnej dobe najpoužívanejší model v automobilovom priemysle. Je rozdelený do troch fáz. Prvou fázou je definovanie si krokov. Po tejto fáze prichádza na rad samotné programovanie, po ktorom nasleduje overovanie predošlých krokov. Funguje to na tom princípe, že sa po fáze programovania otočí naspať nahor na overovanie a výsledok každej fázy sa overí ešte predtým, ako sa prejde do nasledujúcej fázy V-Modelu.

3.2 Agile Model

Tento model z veľkej časti dáva do popredia evolučný rozvoj a rapídne zmeny. Vo viacerých projektoch sa softvéroví inžinieri rozhodnú pre tento spôsob z dôvodu, že vďaka nemu dokážu spracovať aj nepredpokladané zmeny zo strany zákazníckych požiadaviek. Medzi najväčšie výhody patrí to, že nemá pevné fázy, na rozdiel od V-Modelu. Z tohto dôvodu je Agile Model viac flexibilný, čo má za následok zvýšenie produktivity.

3.3 Porovnávanie V-Modelu a Agile Modelu

Obidva modely sú známe najmä z toho dôvodu, že sú jednoduché. Agile Model je navrhnutý tak, aby bol čo najmenej zložitý vo vývojových procesoch. Ale z dôvodu, že softvér v automobiloch vyžaduje komplexnú celistvosť sa v praxi častejšie využíva V-Model. Okrem toho, Agile Model ma viacero nevýhod. [2]

V-Model		Agile-Model	
Výhody	Nevýhody	Výhody	Nevýhody
Jednoduchý	Flexibilita	Jednoduchý	Zlá štruktúra
Silná integrita	Časovo náročné	Flexibilné zmeny	Recyklácia kódu
Skorá verifikácia		Časovo prijateľné	Práca v skupine
			Slabá integrácia

Obr. 1: Tabuľka, ktorá znázorňuje výhody a nevýhody V-Modelu, rovnako ako aj Agile-Modelu.

4 Testovanie

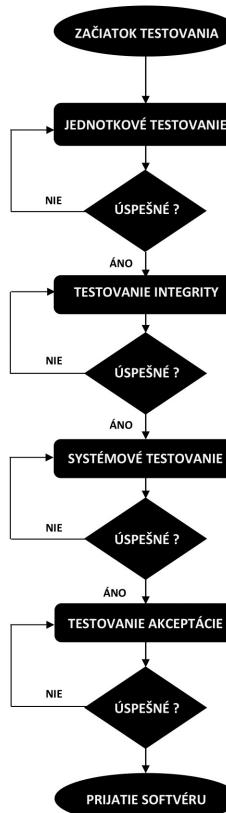
Testovanie softvéru je v automobilovom priemysle jednou z najdôležitejších časti, ktorá by nemohla ostať nikdy zanedbaná. Testovaním každého softvéru sa okrem správnej funkčnosti testuje aj bezpečnosť. Pri vývoji softvéru v automobilovom priemysle sa okrem ohľadu na funkcie a schopnosti softvéru berie ohľad aj na jeho bezpečnosť, keďže zákazníci sa potrebujú v automobiloch cítiť hlavne bezpečne. [1]

4.1 Proces testovania softvéru

Najviac rozšíreným modelom pre testovanie softvéru je V-Model. Testovanie podľa V-Modelu je organizované do viacerých krokov, ktorými sú: jednotkové testovanie, integrácia, systém a jeho akceptovanie.

1. Jednotkové testovanie Účelom jednotkového testovania je zobrať najmenej časť softvéru, ktorú je možné otestovať, izolovať ju od zvyšku kódu a skúšať, či sa správa tak, ako sme očakávali. Každá jednotka sa testuje samostatne ešte predtým, než sa všetky jednotky integrujú do modulov.
2. Testovanie integrity Testovanie integrity spočíva vo vyskúšaní rozhraní medzi jednotlivými modulmi. Za svoj vstup berie jednotlivé testované jednotky, zhromažďí ich do väčších skupín, vykoná sa testovanie a následne sa dodáva výstup vo forme integrovaného systému, ktorý je pripravený na jeho testovanie.

3. Systémové testovanie Tento krok je založený na testovaní kompletne zhôzteného systému, ktorý bol nakonfigurovaný v kontrolovanom prostredí pomocou simulácie reálneho času.
4. Testovanie akceptácie Finálnym krokom tohto procesu je testovanie akceptácie, po ktorom je systém prijatý na prevádzkové použitie podľa zámerov používateľa. [6]



Obr. 2: Vývojový diagram procesu testovania softvéru. Prvotným krokom po začatí testovania je jednotkove testovanie. Ak bolo úspešné, prechádza sa na ďalší krok. Ak nebolo, tak toto testovanie sa bude vykonávať dovtedy, kým nebude úspešné. Rovnaký proces nastáva aj pri testovaní integrity, systémovom testovaní a testovaní akceptácie. Keď sa všetky kroky testovania vykonajú správne, nastáva prijatie samotného softvéru na použíte.

5 Spôsob modelovania automobilky Tesla

Tesla patrí medzi jednu z najlepších, ak nie najlepšiu, automobilku z hľadiska umelej inteligencie, autonomného jazdenia, komplexnosti softvéru vo vozidlách. Vďaka ich progresívnosti a futuristickému zmýšľaniu, Tesla využíva najmodernejšie a najviac inovatívne funkcie a postupy pri vytváraní ich softvérov. [5]

Jedným z najnovších postupov pre modelovanie okolia a objektov je takzvaný Auto Labeling, ktorý je podrobnejšie vysvetlený v nasledujúcej podkapitole.

5.1 Auto Labeling

Automatické modelovania okolia vozidla funguje na spôsobe Clip Labeling.

Clip je objekt, ktorý obsahuje obrovské množstvo dát, ako napríklad viéa, rýchlosť a smer vozidla. Tieto objekty sú odosielané na server, na ktorom beží množstvo neurónových sieti, ktoré sú určené pre umelú inteligenciu, vďaka ktorým je možné získať predbežné informácie, ktorými sú jednotlivé body pre identifikovanie objektov na vozovke.

Prvou úlohou je získanie informácií o vozovke. Tieto informácie sa získavajú pomocou X-ových a Y-ových bodov, vďaka ktorým je možné požiadať neurónové siete o predikciu šírky povrchu vozovky, alebo čiary a zvodidlá. Vďaka tomuto sa získá ďalší bod Z, kvôli ktorému je možné vytvoriť 3D body, ktoré sú následne preprojektované do všetkých kamier z rôznych uhlov. Informácie sa získavajú pomocou viacerých testovacích jázd na jednom alebo viacerých vozidlách, kde sa následne všetky získané informácie spájajú do presnej bodovery zreprodukovej vozovky.

Vďaka tomuto spôsobu je možné modelovať aj ostatné 3D objekty, ktoré sa nachádzajú na, alebo popri vozovke, ako napríklad budovy, prekážky, chodníky, autá, alebo ľudia. [4]

6 Reakcie na témy z prednášok

Grafické vyjadrenie informácií v informatike Táto prednáška bola zameraná na grafické vyjadrenie informácií v informatike, kde sa ako hlavné témy rozoberali vizualizácia, prečo je grafické vyjadrenie informácií dôležité, aké sú ich nástroje a využitia a ich praktická realizácia. Vizualizácia je nejaké znázornenie, alebo formalizácia daného predmetu. Z môjho pohľadu je vizualizácia v akomkoľvek smere veľmi dôležitá, keďže je lepšie raz vidieť, ako viackrát počuť.

Načo budem inžinierom (bakalárom) ? Táto prednáška medzi moje najobľúbenejšie, keďže bol na ňu pozvaný dekan Fakulty Informatiky a Informačných Technológií STU, prof. Ing. Ivan Kotuliak, Phd. Prednášal nám rozsiahle informácie, ktoré boli podané veľmi interaktívnu formou, ktoré sa týkali napríklad motivácie pre štúdium, platy v IT na Slovensku, profesie v IT, alebo vedy. Dozvedel som sa mnoho nových informácií, hlavne z psychologického hľadiska a samozrejme z pohľadu pôsobenia v IT.

Udržateľnosť a etika Prednáška na tému Udržateľnosť a etika bola v mojom vnímaní zameraná na vnútornú motiváciu študentov pri študovaní vysokej školy. Dôležitými radami, ktoré odkazovali na to, aby sme mali jasný cieľ, uniesli sme náročnosť posledných dní a v neposlednom rade budovanie udržateľnosti. Okrem zaujímavých diskusií, ktoré boli zamerané na udržateľnosť študentov na prednáškách, zlepšovanie prednášok ma taktiež veľmi zaujala časť, ktorá sa zameriavala na vplyv počítača na zdravie človeka. Myslím si, že zdravie človeka by malo byť aj pri práci v IT na prvom mieste, keďže zdravie je to, čo môže ovplyvniť inšpiráciu a produktivitu pri práci.

7 Záver

Hlavným cieľom tohto článku bolo oboznámiť čitateľa s informáciami ohľadom softvéru v automobilovom priemysle a taktiež si vo väčšom rozmere rozšíriť moje doterajšie poznatky.

V prvej časti článku, som podrobnejšie priblížil postupy vo vývoji automobilového softvéru, kde som sa systematicky rozpísal hlavné ciele daných postupov a načo sú dané postupy využívané.

V druhej časti som rozobral dva najpoužívanejšie modely, ktoré slúžia na definovanie jednotlivých krokov, pomocou ktorých sa bude softvér vyvíjať.

Treťou kapitolou bolo vysvetlenie toho, ako funguje testovanie softvéru v automobilovom priemysle. Jednotlivé kroky testovania softvéru som priblížil aj pomocou vývojového diagramu, z ktorého je postupnosť pri testovaní lepšie čitateľná.

Poslednou a z môjho pohľadu najdôležitejšou a najzaujímavejšou tému tohto článku bol spôsob modelovania automobilky Tesla. V tejto kapitole som podrobnejšie vysvetlil najnovším spôsobom, akým táto automobilová firma modeluje ostatné vozidlá, budovy, chodcov, alebo ostatné objekty.

Dôležitosť softvéru v automobilovom priemysle je v dnešnej dobe jedným z najdôležitejších faktorov pri návrhu a následnej realizácii nových vozidiel. Podľa posledných zistených informácií je dôležitosť samotného softvéru pri vývoji automobilu na úrovni 40%. Z môjho pohľadu sa s pribúdajúcim časom bude na softvér v automobile dávať čoraz väčší dôraz, keďže terajšia softvérová doba stále napriekuje a čo bolo včera aktuálne a najnovšie, dnes už byť nemusí.

Literatúra

- [1] Vard Antinyan and Henrik Sandgren. Software safety analysis to support iso 26262-6 compliance in agile development. *IEEE Software*, 38(3):52–60, 2021.
- [2] Alireza Haghaghkhah, Markku Oivo, Ahmad Banijamali, and Pasi Kuvaja. Improving the state of automotive software engineering. *IEEE Software*, 34(5):82–86, 2017.
- [3] Lama J. Moukahal, Marwa A. Elsayed, and Mohammad Zulkernine. Vehicle software engineering (vse): Research and practice. *IEEE Internet of Things Journal*, 7(10):10137–10149, 2020.
- [4] Tesla. Tesla ai day. <https://www.youtube.com/watch?v=j0z4FweCy4Mt=5593s>.
- [5] R. Thomas, L. Garbuio, L. Gerbaud, and H. Chazal. Modeling and design analysis of the tesla model s induction motor. In *2020 International Conference on Electrical Machines (ICEM)*, volume 1, pages 495–501, 2020.
- [6] Antonio Tierno, Max M. Santos, Benedito A. Arruda, and João N. H. da Rosa. Open issues for the automotive software testing. In *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1–8, 2016.