

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 predpokladať možnú škodu pri vzniknutom nebezpečenstve a čo najefektívnejšie jej zaísť.

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, 7 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 softvérului

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

Vývoj automobilového softvéru a jeho samotné aplikovanie je v dnešnej dobe jeden z najhlavnejších procesov pri zhodnotení automobilu. Je to proces, ktorý pozostáva z viacerých postupov:

1. Budovanie systému
2. Vývoj softvéru
3. Prevádzka

Prvý z postupov procesu je budovanie systému. V tomto postupe sa výrobcovia originálnych zariadení obvykle zaoberejú požiadavkami, ktoré sú primárne zamerané na funkčnosť zariadenia, ktoré bude najviac vyhovovať stanoveným požiadavkám (npr. riadiaca jednotka, hardvér palubného počítača). Tito 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 viacero 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 predaja udržiavaný aktualizáciami, aby prevádzka a funkčnosť samotného softvéru bola čo najdlhšia a najefektívnejšia. [1]

3 Modely pre vývoj automobilového softvérului

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 častí. Prvou časťou je definovanie si krokov. Po tejto časti prichádza na rad samotné programovanie, po ktorom nasleduje overovanie predošlých krokov. Funguje to na tom princípe, že sa po fáze programovania otočí naspäť 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 zákaznických 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, ktoré sa vo vývoji softvéru pre automobily považujú za kľúčové, ako napríklad: slabá softvérová integrita, obmedzené znova použitie kódu, nepriaznivé efekty na kvalitu práce a zložitá organizácia celej štruktúry.

4 Testovanie

Testovanie softvéru je v automobilovom priemysle jednou z najdôležitejších časti, ktorá by nemohla zostať 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.

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 skúsať, 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 zhotoveného 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. [2]



Obr. 1: Všetky informácie spojené dokopy, ktorým výsledkom je modelovanie podstatných objektov pri jazde vozidlom. Všetky pohyby, ktoré sa reálne vykonávaju, sú aj presne modelované. Automobily, ktoré stoja, majú nastavenú rýchlosť 0, aby softvér dokázal identifikovať, že sú zaparkované a chodci sú stále monitorovaní.

5 Modelovanie

6 Spôsob modelovania automobilky Tesla

6.1 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 získava 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 jazd na jednom alebo viacerých vozidlách, kde sa následne všetky získané informácie spájajú do presnej bodoby 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.

7 Záver

Literatúra

- [1] 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.
- [2] 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.