

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Tím 22

Dokumentácia k riadeniu

Tímový projekt 2019/2020

Vedúci tímu:

Ing. Miroslav Rác

Členovia tímu:

Bc. Baláž Branislav

Bc. Bokor Peter

Bc. Harnúšek Ondrej

Bc. Karel Max Dávid

Bc. Letanec Richard

Bc. Vejčíková Veronika

Bc. Vrtal Tomáš

Úvod	4
Role členov tímu	4
Aplikácie manažmentov	5
Manažment komunikácie	5
Manažment úloh	5
Manažment verziovania	6
Manažment písania kódu	6
Sumarizácie šprintov	6
Šprint 1:	6
Šprint 2:	7
Globálna retrospektíva ZS/LS	9
Motivačný dokument	9
Tím 22 - Niečo o nás	9
Motivácia pre tému 6 - Predikcia ceny vozidla na základe inzerátu [CarAd]	10
Prečo chceme túto tému?	10
Prečo sme pre tento projekt ten pravý tím?	11
Motivácia pre tému 12 - Inovatívny portál pre boj s antisociálnym správaním s využitím umelej inteligencie [FireAnt]	11
Prečo chceme túto tému?	11
Prečo sme pre tento projekt ten pravý tím?	11
Motivácia pre tému 18 - Dynamické váženie vozidiel počas premávky [WIM]	12
Prečo chceme túto tému?	12
Prečo sme pre tento projekt ten pravý tím?	12
Príloha k motivačnému dokumentu A - Zoradenie tém podľa priority	12
Príloha k motivačnému dokumentu B - Rozvrh tímu	13
Metodiky	13
Metodika komunikácie	13
Dedikácia	13
Osobné stretnutia	14
Slack	14
Email	15
Metodika verziovania kódu	15
Dedikácia	15
Použité technológie	15
Commit	15
Vetva	16
Pull request	17

Metodika písania kódu v jazyku Python	17
Dedikácia	19
Zaužívané štandardy	19
Používanie nápovedí typov (type hints)	19
Dokumentovanie kódu	20
Export evidencie úloh	21

Úvod

Tento dokument popisuje riadenie tímu Predicar (v zoskupení tak ako je popísané v kapitole Role členov tímu) pri práci na projekte CARad v rámci predmetu Tímový projekt. Taktiež obsahuje všetky dokumenty, ktoré súvisia s organizáciou v rámci školského projektu ako sú motivačný dokument.

Role členov tímu

Väčšinu kľúčových povinností sme si v rámci tímu rozdelili tak aby každá pozícia bola obsadená aspoň dvoma členmi. Potom v prípade neprítomnosti vie projekt bezproblémovo napredovať.

Keďže si viacerí z nás chceli vyskúšať pozíciu Scrum mastera striedame sa po jednom šprinte. Číslo šprintu, v ktorom sa daný člen tímu zhostil úlohy je v zátvorke za menom pozície.

Bc. Baláž Branislav

- Machine learning
- Scrum master (3)
- Frontend developer
- Backend developer

Bc. Bokor Peter

- Machine learning
- Project manager
- Databázy
- Backend developer

Bc. Harnúšek Ondrej

- NLP specialist
- Backend developer

- Tester

Bc. Karel Max Dávid

- Administrácia tímového servera
- Git master
- Scrum master (2)

Bc. Letanec Richard

- Backend developer
- Documentation Manager

Bc. Vejčíková Veronika

- NLP specialist
- Správa tímového webu
- Scrum master (1,4)

Bc. Vrtal Tomáš

- Frontend developer
- Backend developer

Aplikácie manažmentov

Na efektívne riadenie projektu sme napísali nižšie popísané metodiky. Tímy by sa mal riadiť každý člen tímu a ich kompletná forma sa nachádza v kapitole metodiky.

Manažment komunikácie

Komunikácia je neoddeliteľnou súčasťou efektívneho riadenia tímu. Na komunikáciu používame výhradne Slack. Tým zabezpečíme, že všetky informácie sú na jednom mieste a spätne dohľadateľné. V Slacku máme vytvorené kanále pričom, každý má vlastný účel a pravidlá popísané v metodike komunikácie. Okrem toho máme integráciu s tromi Slack botmi: Jira, Slack a Standup Alice. Posledný je na zefektívnenie standup meetingov v dňoch keď sa nevieme stretnúť.

Výmenu dokumentov riešime cez tímový Google Drive. Popis jednotlivých kanálov a pravidiel komunikácie máme presne opísanú v "[metodike komunikácie](#)".

Manažment úloh

Na sledovanie stavu šprintu a jeho organizáciu používame nástroj Jira Software, ktorý máme nasadený na tímovom serveri. Každý člen tímu môže pridávať do backlogu *Tasky* alebo *Story*. Následne je na product ownerovi aby sa staral o backlog a zaradoval jednotlivé položky v backlogu podľa priority.

Ohodnotenie úloh sa deje priebežne metódou *planning poker*. Tým chceme docieľiť aby sme pri začínaní nového šprintu neminuli príliš veľa času na hodnotenie úloh ale radšej

rozhodli a diskutovali čo má byť cieľom šprintu. Pred samotným hodnotením musí každá úloha obsahovať definition of done pre danú úlohu. Existencia jej podúloh nieje nutná.

Na záver šprintu je každá úloha odprezentovaná a product owner rozhoduje či bude akceptovaná alebo nie.

Manažment verziovania

Pri spolupráci viacerých developerov je nutné efektívne zdieľať kód. Na to používame technológiu git a repozitáre máme hostované na github.com. Rozhodli sme sa pre každú mikroslužbu vytvoriť vlastný repozitár, taktiež máme jeden pre tímovú stránku.

Detaily s špecifikácie ako pracovať s touto technológiou máme v "[metodike verziovania](#)".

Manažment písania kódu

Na písanie kódu zatiaľ používame Python. Riadime sa štandardmi PEP (Python Enhancement Proposal) a používame typový Python. Tým chceme docieľiť efektívnejšie prehliadky kódov a zlepšiť čitateľnosť kódov. Typový Python zas používame na

Detailnejšie použitie spomenutých pravidiel je popísané v [metodike písania kódu](#).

Sumarizácie šprintov

Šprint 1 - BENZ:

V tomto šprinte sme používali angličtinu na označenie úloh. Ukázalo sa, že to spôsobuje komplikácie a zdržania pri písaní definícií úloh. Preto sme neskôr prešli na Jiru v slovenčine.

Cieľ: To implement prototype of the microservice architecture.

Trvanie: 11/Oct/19 11:24 AM - 24/Oct/19 3:20 PM

Názov úlohy	Popis	Zoznam podúloh	Odpracovaný čas	Akceptované
[CAR-6] Integrate Jira	As a team, we want to have integrated Jira software with our team website, in order to work efficiently and track our progress.	CAR-11	4h	Áno
[CAR-4] Create functional	As a product owner, I want a prototype of the system,	CAR-29, CAR-32,	2d2h10m	Áno

system prototype	to validate the proposed system architecture. no need of graphical UI <ul style="list-style-type: none"> just build REST API (json) be aware of GET/POST/PATCH methods meaning 	CAR-33, CAR-34, CAR-35, CAR-36, CAR-37		
[CAR-3] Submit TP Cup application	As a team, we want to attend TP Cup, in order to win.	CAR-7	1h30m	Áno
[CAR-2] Finalize team web site design	As a team, we need to have a website, to present our work.	CAR-8, CAR-9, CAR-10, CAR-28, CAR-30, CAR-31	7h15m	Áno
[CAR-1] Create 1st design of system architecture	As a product owner, I want to know the architecture of the system, to identify the components that will be used.	CAR-12, CAR-13, CAR-14, CAR-15, CAR-16	5h40m	Áno

Šprint 2 - LAURIN & KLEMENT:

Ciel': Chceme zjednodušiť deployment a preskúmať ďalšie možnosti smerovania našej aplikácie z pohľadu používateľa

Trvanie: 24/Oct/19 5:20 PM - 07/Nov/19 3:33 PM

Názov úlohy	Popis	Zoznam podúloh	Odpracovaný čas	Akceptované
[CAR-44] Low fidelity user interface wireframe	As a user I want to see a low fidelity wireframe of a report, so I do know what to expect from the application. DOD: <ul style="list-style-type: none"> visual wireframe <ul style="list-style-type: none"> possible variants well documented <ul style="list-style-type: none"> each component has it's description why is it there what does it tell what kind of data does it 	CAR-54, CAR-55, CAR-56, CAR-57	1d35m	Áno

	<p>need</p> <ul style="list-style-type: none"> it provides a base for UX survey questions 			
[CAR-43] Dockerise the application	<p>As a programmer, I want to easily run up-to-date application with all components, so I can easily add new fancy features.</p> <p>DOD:</p> <ul style="list-style-type: none"> each member can run the application by typing some simple command each repository gets configuration from some environment file (or compose.yaml) 	CAR-47, CAR-48, CAR-49, CAR-67	6h30m	Áno
[CAR-42] Create and connect with system database prototype	<p>As a user, I want to be provided with reports based on long-term history, so reports are as precise as possible.</p> <p>DOD:</p> <ul style="list-style-type: none"> data is persisted each service which do need to persist data uses mongoDB 	CAR-50, CAR-51, CAR-52, CAR-53	2h30m	Áno
[CAR-40] Create writing code in Python methodology	I want to know how write and testing code in Python	CAR-60	4h	Áno
[CAR-39] Create code versioning methodology	I want to know how create and name: commit, branch, pull request		1h	Áno
[CAR-24] Integrate Jira and Slack	As a team we need to have Jira and Slack integrated together to be up to date with changes and progress in Jira.	CAR-27, CAR-45, CAR-46	1h	Áno

Šprint 3 - FIAT:

Cieľ: Získavanie, predspracovanie a analýza dát.

Trvanie: 07/Nov/19 5:23 PM - 07/Nov/19 3:33 PM

Názov úlohy	Popis	Zoznam podúloh	Odpracovaný čas	Akceptované
CAR-5	Analyze NLP tools for Slovak Language	CAR-63, CAR-64, CAR-73		Áno
CAR-68	Pridat github bota do slacku	----	15m	Áno
CAR-62	Vytvorenie parsera pre inzeráty od Miroslava Ráca	CAR-76, CAR-82, CAR-99		Áno
CAR-59	Extrakcia údajov z textu inzerátu - word2vec	CAR-81, CAR-83	4h	Áno
CAR-75	Tool pre rychle anotovanie a anotovanie dát	CAR-80, CAR-84		Nie - *
CAR-71	Analyzovať portály inzerátov	CAR-77, CAR-78, CAR-88		Áno
CAR-66	Skompletizovať veci na odovzdanie po 9. týždni	CAR-74, CAR-85, CAR-86, CAR-87		Nie - **
CAR-58	Dátová analýza - dĺžka predaja auta	CAR-79		Áno

* - CAR-59 - nebol ukončený z dôvodu zlého nastavenia programu na anotáciu. Vzniknuté dáta boli nepoužiteľné

** - CAR-66 - Vytvorené subtasky nedovoľovali ukončiť úlohu pred skončením šprintu. Konkrétne CAR-87 na odovzdanie dokumentov musí byť vytvorený report z posledného šprintu.

Globálna retrospektíva ZS/LS

// TODO: doplniť po ukončení semestra

Motivačný dokument

Tím 22 - Niečo o nás

Mailový kontakt: fiit.tp.22@gmail.com

Sme tím pripravený s chuťou využívať svoje poznatky z doterajšieho štúdia a praxe, a učiť sa novým veciam. Radi pri svojej práci používame moderné technológie.

Všetci v tíme máme skúsenosti s programovacím jazykom Python. Štyria z nás mali bakalársku prácu v oblasti strojového učenia alebo dátovej analýzy s použitím Pythonu. Z toho v troch prácach išlo o hlboké učenie v oblastiach spracovania textu a vizuálnych dát. Využívali sme technológie ako Numpy, Scikit-Learn, Tensorflow + Keras, Pandas, Matplotlib, Seaborn, StanfordNlp, Gensim, či BeautifulSoup a nadobudli sme znalosti o neurónových sieťach pre NLP, konvolučných neurónových sieťach na klasifikáciu, segmentáciu a tiež autoenkóderoch. Ďalší členovia tímu vo svojich prácach tiež využívali Python pre podporné účely. Máme tiež skúsenosti s tradičnými aplikáciami strojového učenia ako používanie rozhodovacích stromov, jednoduchých klasifikátorov a zhlukovania, ale aj s predspracovaním, úpravou a generovaním dát, selekciou a vytváraním črt - hlavne z predmetu IAU, ktorý piati z nás absolvovali na bakalárskom stupni štúdia.

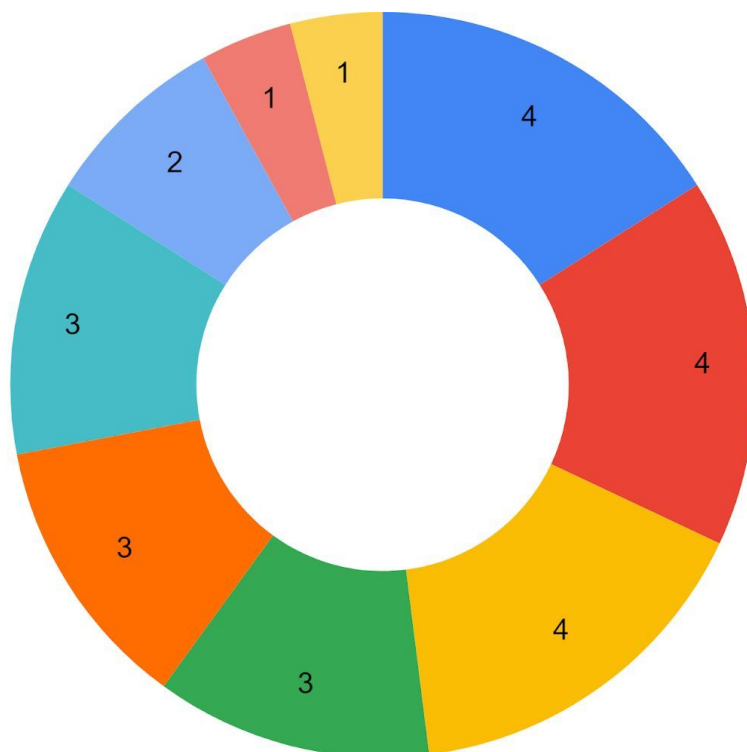
Niektorí z nás sa stretli s jazykom Python v práci, či vo vlastných projektoch, kde sme získali skúsenosti s vývojom backendu webových aplikácií s použitím knižnice Flask alebo grafického používateľského rozhrania s použitím PyQt.

Taktiež sme si všetci prešli základmi z bakalárskeho štúdia: Java, C, C++, SQL, GIT, paralelné a systémové programovanie v C, a niektorí z nás svoje schopnosti využívajú aj v praxi pri práci - MS SQL Server (TSQL).

V praxi sme sa tiež stretli s automatizovaným testovaním, nasadením produktov do prevádzky pomocou technológií Docker, AWS, Jenkins alebo Travis a cloudovými službami.

Skúsenosti máme aj s implementovaním webových aplikácií v prostredí Java s aplikačným serverom Tomcat a využitím webových služieb (REST, SOAP).

- Vyhľadávanie informácií
- Pokročilé databázové technológie
- Štatistické metódy vyhodnocovania experimentov
- Neurónové siete
- Vizualizácia dát
- Objektovo orientovaná analýza a návrh softvéru
- Prírodou inšpirované počítanie
- Aplikáčné architektúry softvérových systémov
- Vnorené systémy



Naše vybrané predmety v prvom roku inžinierskeho štúdia

Motivácia pre tému 6 - Predikcia ceny vozidla na základe inzerátu [CarAd]

Prečo chceme túto tému?

Páči sa nám, že projekt má jasne definované použitie - jednoznačne pomenovaný optimalizačný problém: že predaj vozidla chceme docieľiť za minimalizácie času predaja a maximalizácie ceny vozidla, ktoré sú však negatívne korelované. Zároveň v realizácii detailov v rámci tohto cieľa máme voľnú ruku.

Viacerí v tíme máme záujem ďalej sa rozvíjať v oblasti strojového učenia/hlbokého učenia, zberu, spracovania a analýzy dát. Využitie metód spracovania prirodzeného jazyka na slovenčinu považujeme za veľmi zaujímavú výzvu, keďže väčšina dostupných “hotových” riešení pre NLP vhodných pre napr. Angličtinu, pre slovenský jazyk nie je dostupná. Možné nadobudnuté skúsenosti so spracovaním slovenčiny považujeme za veľkú výhodu do budúcnosti pre nás aj pre výskum v slovenskom jazyku.

Taktiež ide o využitie strojového učenia na reálnom použiteľnom “produkte” čo pri mnohých výskumoch napr. v hlbokom učení nie je samozrejmé. Čiže spojenie strojového učenia a reálnej užitočnej aplikácie vnímame ako veľké plus.

Prečo sme pre tento projekt ten pravý tím?

Lebo sú nám blízke a známe spomínané technológie vyžadované pre projekt a Vue.js sa rýchlo naučíme. Viacerých nás zaujíma strojové učenie/hlboké učenie – štyria z nás mali takto orientovanú BP z toho dvaja z nás spracovanie jazyka – z toho jeden v slovenčine.

To, že táto téma je pre nás naozaj vhodná svedčí aj naša voľba predmetov - až štyria z nás absolvujeme v tomto semestri Vyhľadávanie informácií a traja Neurónové siete. V našom repertoári sa nachádzajú schopnosti pre scrappovanie a crawlovanie webu. Používali sme Docker, máme skúsenosti s Flask Python frameworkom pre backend, REST API a mikroslužbami. V neposlednom rade viacerí z nás mali BP zameranú na neurónové siete a prácu s dátami. Tensorflow a Keras nám nie je cudzí a tiež sme pracovali s knižnicou scikit-learn.

S úlohou vhodnej vizualizácie výsledkov pre podporu informovaného rozhodnutia používateľa si poradíme.

Motivácia pre tému 12 - Inovatívny portál pre boj s antisociálnym správaním s využitím umelej inteligencie [FireAnt]

Prečo chceme túto tému?

Myslíme si, že ide o užitočnú a veľmi aktuálnu tému. Klamstvá šíriace sa na internete nás trápia a naše riešenie by mohlo pomôcť v tejto oblasti. Takto by sme mohli zlepšiť informovanosť širokej spoločnosti, čo by napr. v medicínskej doméne mohlo znamenať lepšiu zdravotnú úroveň.

Viacerí sa zaujímame o oblasť spracovania prirodzeného jazyka.

Páči sa nám sloboda v možnosti implementovať viaceré nástroje pre boj s antisociálnym správaním. Konkrétne nás zaujala myšlienka “sprievodcu” - rozšírenia prehliadača, ako veľmi praktické riešenie pre neskúseného prehliadača internetu, ktorý by takto mohol byť informovaný o kvalite webu a kvalite informácií, s ktorými sa stretáva.

Chceli by sme tento projekt dotiahnuť do podoby, aby bol vhodný na nasadenie do reálnej prevádzky a prístupný verejnosti.

Prečo sme pre tento projekt ten pravý tím?

Lebo viacerí z nás sa zaujímajú o spracovanie prirodzeného jazyka a ešte viac z nás sa zaoberá hlbokým učením. Lebo nás baví získavanie, analyzovanie dát a strojové učenie, o čom svedčí aj náš výber predmetov - Vyhľadávanie informácií a Neurónové siete.

Používali sme tiež Docker, máme skúsenosti s Flask Python frameworkom pre backend, REST API a mikroslužbami.

Motivácia pre tému 18 - Dynamické váženie vozidiel počas premávky [WIM]

Prečo chceme túto tému?

Táto téma nás zaujala z dôvodu, že sa jedná o už existujúci projekt, ktorý by sme radi skúsili vylepšiť. Vďaka tomu vieme aj porovnávať nami vytvorené riešenie a taktiež dostať spätnú väzbu od skúsenejších, pôvodných autorov tohto systému. Na projekte sa nám páči, že má reálne využitie a veľkú perspektívu do budúcnosti. Tiež nás láka výzva integrácie nami vytvoreného riešenia do reálneho, už existujúceho systému.

Taktiež je to projekt na ktorom si každý z nás vie nájsť časť v ktorej sa cíti viac doma. Či už je to príprava dát, tréningovanie neurónových alebo vnorené systémy.

Prečo sme pre tento projekt ten pravý tím?

Pri spracovaní tejto témy vieme využiť naše skúsenosti so strojovým učením. Viacerí máme skúsenosti so zberom, úpravou a predspracovaním dát a vytváraním číť, či už zo školy z predmetu Inteligentná analýza údajov, kde sme riešili projekt s podobným formátom a postupom ako táto téma, ale aj z bakalárskych prác. Máme skúsenosti s používanými technológiami, hlavne s jazykom Python - z bakalárskych prác a praxe.

S operačným systémom GNU/Linux sme sa všetci stretli na predmete Operačné systémy a niektorí ho používame v praxi/denne. S poslednou technológiou, jazykom C/C++, sme sa v škole stretli všetci, pričom jeden člen tímu mal v C++ bakalársku prácu. Ani kompilovať programy zo zdroja nám nerobí problém.

Rozumieme, že pre mnohé úlohy algoritmické riešenia nepokrývajú všetky prípady, ktoré môžu nastať. Tu sa ponúka potenciál iných metód napríklad využitia hlbokého učenia, s ktorým sme sa viacerí stretli a chceme využiť príležitosť skúsiť ho uplatniť v tomto prípade

Príloha k motivačnému dokumentu A - Zoradenie tém podľa priority

1. Téma č. 6. Predikcia ceny vozidla na základe inzerátu [CarAd]
2. Téma č. 12. Inovatívny portál pre boj s antisociálnym správaním s využitím umelej inteligencie [FireAnt]
3. Téma č. 18. Dynamické váženie vozidiel počas premávky [WIM]
4. Téma č. 8. Vyhľadávanie pomocou obrázkov [ImageSearch]
5. Téma č. 2. Prehľadavanie a vizualizácia textov [TextTool]
6. Téma č. 11. We AR city - Inteligentná komunikácia občana s mestom [city4us]
7. Téma č. 5. Pomoc zvieratám v núdzi [AnimalRescue]

8. Téma č. 19. Automatizácia procesu spracovania a klasifikácie textov [TextProcessing]
9. Téma č. 9. Multibanková mobilná aplikácia pre manažment financií [MultiBank]
10. Téma č. 22. Smart pivovar [SmartBrew]
11. Téma č. 20. Použiteľnosť mobilných aplikácií [MobeUX]
12. Téma č. 15. Inteligentný tvorca príbehov z dát [TellStoryAI]
13. Téma č. 1. Event Navigation [EventNav]
14. Téma č. 3. Prostredie na vizualizáciu mikrogridu [GridBox]
15. Téma č. 4. Vizualizácia softvéru vo virtuálnej a rozšírenej realite 2.0 [VizReal]
16. Téma č. 7. Animované architektúry [AnimArch]
17. Téma č. 13. Podpora kvality služieb pre budúci Internet [QoSbySDN]
18. Téma č. 14. Bezpečné manažovanie siete v prostredí Internetu vecí [SecIoT]
19. Téma č. 16. Virtuálna identita [Virtual ID]
20. Téma č. 17. Blockchain platobné brány [BlockPay]
21. Téma č. 21. Kolaboratívne virtuálne prostredie [Coven]
22. Téma č. 10. Databanka otázok a úloh [FIIT-DU]

Príloha k motivačnému dokumentu B - Rozvrh tímu

Deň	8:00 - 8:50	9:00 - 9:50	10:00 - 10:50	11:00 - 11:50	12:00 - 12:50	13:00 - 13:50	14:00 - 14:50	15:00 - 15:50	16:00 - 16:50	17:00 - 17:50	18:00 - 18:50
Pondelok	VINF				NSIETE						
Utorok		VNOS		NSIETE			ASS	VISS	TP1		
Streda	NSIETE	VINF					MTS		MTS		
Štvrtok	VNOS	PPGSO		VINF					SOGAM		
	ASS	ASS		ASS					ASS		
Piatok	PDT										

Max Dávid Karel	Tomáš Vrhel	Stretnutie s vedúcim
Branislav Baláž	Veronika Vejčíková	Stretnutie tímu
Ondrej Harnúšek	Všetci	
Peter Bokor	Prednáška	
Richard Letanec		

Metodiky

Metodika komunikácie

1. Dedikácia

Metodika definuje princípy a pravidlá komunikácie pre tím Predicar. Opisuje komunikáciu medzi členmi tímu, medzi členmi tímu a vedúcim tímu a komunikáciu s verejnosťou.

2. Osobné stretnutia

Najväčšia časť komunikácie prebieha na spoločných stretnutiach tímu. Formálne stretnutie tímu aj s vedúcim prebieha v štvrtok od 14:00 do 17:00, kde sa hlavne diskutuje o projekte a jeho vývoji.

Okrem toho sa tím stretáva aj v pondelok od 14:00 do 18:00, kde okrem diskusie tím spoločne pracuje na projekte.

3. Slack

Na spoločnú komunikáciu ak nie je možnosť osobného stretnutia využívame nástroj Slack. Každý člen tímu musí pravidelne kontrolovať Slack, či už na mobile alebo počítači - teda každý deň v prípade potreby komunikovať. V tomto nástroji máme viacero komunikačných kanálov:

- **brainstorming** – Kanál, ktorý slúži na zdieľanie akýchkoľvek nápadov či pripomienok.
- **general** – Kanál, ktorý slúži na všeobecnú diskusiu o projekte a úlohách a taktiež aj v prípade, že člen tímu potrebuje radu alebo pomoc.
- **jira** - Kanál, ktorý slúži na notifikácie z nástroja Jira. Nachádzajú sa tu správy o vytvorení, editovaní, priradení a komentovaní príbehu alebo úlohy a ešte správy o tom ak úloha sa dostane do stavu “Ready for Review” a “Done”, čo nám slúži na lepší prehľad o tom čo kto robí.
- **methodics** – Kanál, ktorý slúži na komunikáciu o upravovaní existujúcich a tvorení nových metodík.
- **random** – Kanál, ktorý slúži na komunikáciu netýkajúcu sa tímového projektu.
- **standup** – Kanál, na ktorom každú stredu a nedeľu večer do 20:00 každý člen tímu napíše, na čom aktuálne pracuje a stav úlohy. Na pripomínanie nám slúži Standup

Alice bot, ktorý každý deň vyzve každého člena tímu aby vyplnil formulár o jeho činnosti, s tým že v ostatné dni okrem stredy a nedele sa nemusí vyjadriť.

- **tim-web** – Kanál, na ktorom sa riešia všetky veci týkajúceho sa webovej stránky tímu.
- **tp_cup** – Kanál, na ktorom sa rieši všetko súvisiace s TP Cup-om.
- **uzitocne_linky** – Kanál, kde sa pridávajú užitočné odkazy aj s popisom aby sa dali jednoducho kedykoľvek dohľadať.

Okrem týchto kanálov, môže člen využiť priamu komunikáciu s iným členom tímu v súkromnom chate.

4. Email

Náš tím používa spoločný email pre komunikáciu s verejnosťou, odkiaľ sa preposielajú emailové správy všetkým členom tímu - fiit.tp.22@gmail.com. Prístup k tomu mailu majú všetci členovia tímu okrem vedúceho tímu.

Metodika verziovania kódu

1. Dedikácia

Metodika definuje princípy a pravidlá uchovávaní a verziovania zdrojového kódu pre tím Predicar. Opisuje použité technológie a spôsob tvorby commitov, vetiev a pull requestov.

2. Použité technológie

Na uchovávanie a verziovanie kódu používame webovú službu Github podporujúcu vývoj softvéru s pomocou verziovacieho nástroja Git.

Na Githube máme vytvorenú organizáciu <https://github.com/predicar>, ktorá obsahuje všetky repozitáre vytvorené našim tímom.

3. Commit

Pomenovaná množina zmien v kóde.

Pravidlá pre vytváranie commitov:

- Commit správa začína vždy slovesom: Add, Fix, Remove, Change, Revert
- Commit správa musí byť pochopiteľná aj bez znalosti daného programovacieho jazyka
- Commit správy sa píšú výlučne v angličtine

- Commitovať treba ucelené, ideálne aj funkčné celky práce
- Jeden commit pokrýva práve jednu zmysluplnú zmenu v kóde
- Zmeny vo formátovaní kódu sú commitované samostatne
- Dĺžka commit správy by nemala presiahnuť viditeľnú dĺžku
- Commit nesmie pokrývať skryté zmeny, ktoré nie sú opísané v správe
- Ak commit súvisí s github issue treba v správe commitu spomenúť vo formáte “#<číslo issue>”
- Ak commit opravuje pripomienky z code review treba v commit správe vo formáte “#<číslo pull requestu>”
- Obsahuje referenciu na Jira task

4. Vetva

Pomenovaná množina commitov určená pre implementáciu konkrétnej funkcionality.

Rozdelenie vetiev:

- master
 - hlavná vetva, ktorá odráža aktuálny stav na produkčnom serveri
 - je zamknutá na priame pushovanie do vetvy, povolené len cez pull requesty z dev vetvy
- dev
 - vývojová vetva, ktorá odráža aktuálny stav na testovacom serveri
 - je zamknutá na priame pushovanie do vetvy, povolené len cez pull requesty z ďalších vetiev
- ďalšie vetvy:
 - vetvy pre implementáciu novej funkcionality:
 - následne sa zlučujú s vetvou dev pomocou pull requestu
 - názov vetvy je Jira issue ID s predponou “feature” (napr. *feature/CAR-1*)
 - pre každú Jira issue je vytvorená maximálne jedna vetva v danom repozitári

- ak robia viacerí na jednej Jira issue, každý má vytvorenú vlastnú a potom sa mergeuje. Názov takejto vetvy je napr feature/CAR-1/meno
- vetvy pre rýchlu opravu chýb:
 - používajú sa len v prípade, že je nevyhnutná okamžitá zmena kódu na produkcii
 - následne sa zlučujú s vetvami master a dev pomocou pull requestu
 - názov vetvy je stručné pomenovanie opravy s predponou “hotfix” (napr. *hotfix/update-db-credentials*)

5. Pull request

Žiadosť o presunutie implementovanej funkcionality z jednej vetvy do inej.

Pravidlá pre vytváranie pull requestov:

- Názov pull requestu je rovnaký ako ID a názov príslušného Jira issue (napr. *CAR-1 Vytvorenie 1. dizajnu systémovej architektúry*)
- Opis pull requestu obsahuje doplňujúce informácie dôležité najmä pri code review
- Názov a opis pull requestu sa píše výlučne v slovenčine
- Každá vetva môže mať maximálne jeden schválený pull request
- Po schválení a zlúčení vetvy sa vetva vymaže (pokiaľ nejde o vetvu dev)

Metodika písania kódu v jazyku Python

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one—and preferably only one—obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea—let's do more of those!

The Zen of Python (PEP 20)

1. Dedikácia

Metodika definuje princípy a pravidlá písania kódu v jazyku Python. Opisuje štýl písania, štandardy ktoré používame a dokumentáciu kódu.

2. Zaužívané štandardy

Pri písaní kódu sa riadime zaužívanými štandardmi [PEP 8](#) (Style Guide for Python Code) pre písanie kódu, [PEP 257](#) (Docstring Conventions) pre dokumentáciu kódu a [PEP 484](#) (Type Hints) pre typovanie kódu.

Kód píšeme v anglickom jazyku, premenným, metódam a triedam dávame výstižný názov.

Názvy premenných a metód píšeme celé malými písmenami, jednotlivé slová oddelené podtržníkom (_).

Názvy tried píšeme štýlom UpperCaseCamelCase.

Funkcie a definície tried sa oddeľujú 2 prázdnyimi riadkami. Definície metód v triedach sa oddeľujú 1 prázdnyim riadkom. Na konci súboru musí byť 1 prázdny riadok.

```
class ExampleClass():

    def example_method():

        example_var1 = 1
        example_var2 = 2

        return example_var1 + example_var2
```

3. Používanie nápovedí typov (type hints)

Pri písaní kódu používame nápovede typov premenných dovážaných do funkcií a typov vracaných funkciami.

```
def example_method(number: int, letter: String) -> String:

    return letter + str(number)
```

4. Dokumentovanie kódu

Komentáre, podobne ako kód, píšeme v **anglickom** jazyku. Riadime sa pri tom štandardom PEP8. Blokové komentáre platia pre kód pod nimi a sú odsadené na rovnakej úrovni ako kód ktorý komentujú. Každý riadok blokového komentáru začína znakom # a jednou medzerou.

Pri písaní docstringu pre metódy a triedy sa riadime štandardom PEP257 pre docstring konvencie a používame Google štýl písania docstringov. Každý docstring obsahuje krátky a výstižný opis metódy, alebo triedy. Každý docstring opisuje privezené argumenty ako Args, typ návratu ako Returns a vyvolané výnimky ako Raises.

```
def connect_to_next_port(self, minimum):
    """Connects to the next available port.

    Args:
        Minimum(int): A port value greater or equal to 1024.

    Returns:
        int: The new minimum port.

    Raises:
        ConnectionError: If no available port is found.
    """
    if minimum < 1024:
        # Note that this raising of ValueError is not mentioned in the doc
        # string's "Raises:" section because it is not appropriate to
        # guarantee this specific behavioral reaction to API misuse.
        raise ValueError('Minimum port must be at least 1024, not %d.' % (minimum,))
    port = self._find_next_open_port(minimum)
    if not port:
        raise ConnectionError('Could not connect to service on %d or higher.' % (minimum,))
    assert port >= minimum, 'Unexpected port %d when minimum was %d.' % (port, minimum)
    return port
```

Viac o Google štýle písania docstringov:

https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html

<http://google.github.io/styleguide/pyguide.html>

Export evidencie úloh

Kompletné exporty evidencie úloh z Jira Software sú priložené k dokumentu alebo sa dajú nájsť na tímovej stránke.