

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Metodiky

Tím 16: Radiant

Autor: Bc. Tomáš Sabo

Contents

1	Metodika dokumentácie	2
1.1	Dedikácia	2
1.2	Role	2
1.3	Zápisnica	2
1.4	Retrospektíva	4
1.5	Dokument s exportom úloh	6
2	Metodika agilného riadenia	8
2.1	Dedikácia	8
2.2	Príprava	8
2.3	Začiatok stretnutia	8
2.4	Tímový stand-up	8
2.5	Priebeh stretnutia	8
2.5.1	Product Owner	8
2.5.2	Právomoc Scrum Mastera	9
2.6	Po šprinte	9
2.7	Definitions	9
2.7.1	Definition of Done	9
2.7.2	Definition of Ready	9
3	Metodika komunikácie	10
3.1	Dedikácia	10
3.2	Komunikačné nástroje	10
3.3	Spoločný email:	11
3.4	Komunikácia s product ownerom	11
4	Metodika verziovania	11
4.1	Dedikácia	11
4.2	Zaužívané pojmy	11
4.3	Repozitár	11
4.4	Branch	12
4.5	Commit správy	12
4.6	Mergovanie vetiev	13

1 Metodika dokumentácie

1.1 Dedikácia

Metodika opisuje zásady a postupy písania dokumentácie, ktoré zaužívame v tíme č. 16 Radiant. Je určená všetkým členom tímu.

1.2 Role

- **Člen tímu** - študent, ktorý je súčasťou tímu č. 16 Radiant na predmete Tímový projekt
- **Autor metodiky** - člen tímu, ktorý napísal danú metodiku
- **Odborník v oblasti** - člen tímu, ktorý ma najviac vedomostí v určitej oblasti
- **Scrum Mster** - člen tímu, ktorý riadi tím a ktorý vytvára zápisnice počas stretnutia s product ownerom

1.3 Zápisnica

Zápisnica slúži na spísanie najdôležitejších udalostí zo stretnutia, tento dokument je taktiež veľmi dôležitý pre product ownera ako zdroj informácií. Zápisnicu spravidla spisuje Scrum Master alebo osoba, ktorá ho zastupuje. Avšak prispieť k obsahu zápisnice môže každý člen tímu.

Zápisnica musí obsahovať:

- **Zhrnutie zo stretnutia** - obsahuje hlavné body stretnutia, čo sa riešilo a robilo v očíslovaných odrážkach
- **Poznámky zo stretnutia** - štrukturovaný text, ktorý obsahuje všetky dôležité informácie zo stretnutia

V prílohe 1, je možné vidieť príklad zápisnice.

Príloha 1: Zápisnica

ZÁPISNICA Č. 1

Tím č. 16 - Radiant

12.10.2020

ZHRNUTIE STRETNUTIA

1. Diskusia ohľadom výberu simulátora pre praktické laboratórium
2. Prvotné napíňanie backlogu
3. Tvorba a spustenie prvého šprintu

POZNAMKY ZO STRETNUTIA

- Prvotné zoznamovanie sa so Scrum Poker, diskusia pri odhadovaní zložitosti jednotlivých User Tasks.
- Vypracovanie analýzy nájdených druhov simulácií so zameraním na elektrické, resp. logické obvody, primárne pre C# knižnice. Následne spísanie dokumentu s opísaním daného simulátora, uvedenie jeho výhod, resp. nevýhod.
- Vybranú simuláciu integrovať do Unity, vypracovanie jednoduchého príkladu s príslušným výstupom.
- Používať LTS verziu Unity (2019.4.12f)
- Vytvorenie webového sídla
 - Názov projektu a príslušný opis
 - Zoznam členov a ich roly
 - Počiatočné zameranie projektu
 - Vloženie zápisnice do dokumentového sídla

1.4 Retrospektíva

Retrospektíva slúži na zhodnotenie priebehu celého šprintu. retrospektíva sa vytvára pomocou dvoch otázok - Čo bolo dobré na šprinte a čo bolo zlé na šprinte, prípadne čo by sa mohlo zlepšiť. Retrospektívu vytvára člen tímu počas stretnutia alebo tesne po stretnutí na základe odpovedí jednotlivých členov tímu, následne sa spíše zhodnotenie celého šprintu.

Retrospektíva musí obsahovať:

- **Čo bolo dobré na šprinte** - spíše sa pozitíva šprintu
- **Čo bolo zlé na šprinte** - spíše sa negatíva šprintu, prípadne pripomienky
- **Zhodnotenie šprintu** - opíše sa celý výsledok šprintu, aké pravidlá chce tím dodržiavať alebo zmeniť v nasledujúcich šprintoch

V prílohe 2, je možné vidieť príklad retrospektívy.

Príloha 2: Retrospektíva

RETROSKEPTÍVA Č. 1

Tím č. 16 - Radiant

26.10.2020

POZITÍVA ŠPRINTU

- Hneď po zadelení úloha vytvorení šprintu product ownerom, sme si tieto úlohy medzi sebou rozdelili
- Pracovali sme priebežne na úlohách, čiže už v polovici šprintu každý člen tímu prezentoval jeho progres na jemu pridelenú úlohu
- Pravidelná komunikácia počas celého šprintu, nielen počas bloku určeného na spoločnú prácu alebo stretnutie s product ownerom
- Vzájomné pomáhajúce si počas práce na úlohách
- Všetky úlohy boli načas dokončené

NEGATÍVA ŠPRINTU

Žiadne

ZHODNOTENIE

- Pracovné nasadenie na ďalšom šprinte ponecháme zatiaľ nezmenené po prvom šprinte

1.5 Dokument s exportom úloh

Na začiatku každého nového šprintu, po plánovaní úloh, člen tímu vyexportuje dokument, ktorý obsahuje aktuálny stav úloh z nástroja *Azure DevOps*. Tento export musí obsahovať všetky evidované úlohy, ktoré vzišli z našich stretnutí. Ku každej User Story sa uvádzajú dôležité informácie ako názov, popis, stav úlohy, zodpovedný riešiteľ a podobne.

V prílohe 3, je možné vidieť príklad dokumentu s exportom úloh.

Príloha 3: Dokument s exportom úloh

D	Title	Work Item Type	State	Description	Story Points	Iteration Path	Original Estimate	Assigned To
15604	Vytvorenie webového sidla	User Story	Closed	Vytvorenie webového sidla * zapisnica * nazov projektu + popis * zoznam clenov timu + roly * na co sme sa na zaciatok projektu zamerali	13	Radiant\Apple		Bc. Ľubomír Kurčák
15609	Spojzdenie serveru cez virtualne stroje	Task	Closed			Radiant\Apple	4	Bc. Ľubomír Kurčák
15610	Nasadenie sablony	Task	Closed			Radiant\Apple	2	Tomas Sabo
15611	Naplnenie obsahu stranky	Task	Closed	Na stranke by mali byt aj obrázky		Radiant\Apple	4	Bc. Patrik Tománek
15603	Analyza simulatorov pre logicke a elektricke obvody	User Story	Closed	prioritne csharp libraries ked sa nenajdu tak aj ine jazyky Treba spisat aj dokument, vzdy linka, popis, sample, vyhody/newyhody. Vsetko pojde do vysledneho dokumentu. Idealne nuget package * circuit blox	8	Radiant\Apple		Bc. Ľubomír Kurčák
15605	Analyza simulatorov pre logicke obvody	Task	Closed			Radiant\Apple	4	Bc. Viktor Beňo
15606	Analyza simulatorov pre elektricke obvody	Task	Closed			Radiant\Apple	4	Bc. Erik Pafa
15607	Prva integracia v Unity	User Story	Closed	Integrovanie vybranej kniznice na logicke/elektricke obvody do Unity Rozchodenie sample Napríklad jednoduchy obvod, a program vypise voltage v bode ("meram napatie medzi X Y) Verzia LTS Release 2019.4.12f1 Released: 7 October 2020 https://unity3d.com/unity/qa/lts-releases	5	Radiant\Apple		Bc. Ľubomír Kurčák
15608	Prva integracia v Unity	Task	Closed			Radiant\Apple	8	Bc. Ľubomír Kurčák

2 Metodika agilného riadenia

2.1 Dedikácia

Metodika opisuje princípy riadenia prostredníctvom Agile, táto metodika je zaužívaná v tíme č. 16 Radiant a je určená všetkým členom tímu.

2.2 Príprava

Scrum Master by mal ešte pred začiatkom stretnutia si pripraviť osnovu stretnutia, podľa ktorej stretnutie bude prebiehať. V prípade zmeny Scrum Mastera je potrebné sa dohodnúť s členmi tímu na pridelení pozície Scrum Mastera inému členovi ešte niekoľko dní pred začiatkom stretnutia.

2.3 Začiatok stretnutia

Na začiatku stretnutia si členovia pozrú v plánovacom kalendári ich úlohy na dnešné stretnutie. Ak niektoré úlohy nie sú pridelené, členovia sa rozhodnú na ktorej úlohe sa začne pracovať.

2.4 Tímový stand-up

Na začiatku každého stretnutia prebieha tímový stand-up, ktorý slúži na udržanie prehľadu o vývoji produktu a o zistení možných komplikácií.

Každý člen tímu musí odpovedať tri základné otázky:

- Čo som spravil na projekte od posledného stretnutia?
- Čo plánujem spraviť na projekte do budúceho stretnutia?
- Existuje nejaká prekážka, ktorá mi bráni splniť úlohy?

2.5 Priebeh stretnutia

Stretnutie prebieha cez konferenčný hovor, a členovia pracujú na svojich pridelených úlohách. Na konci stretnutia si každý člen poznačí svoj odpracovaný čas na úlohách do plánovacieho kalendára. Počas stretnutia je povinný jeden člen tímu zapisovať poznámky zo stretnutia a taktiež je tento člen zodpovedný za spísanie zápisnice ku danému dňu zo stretnutia. Po skončení stretnutia je potrebné ešte v ten deň spísať zhodnotenie stretnutia, toto zhodnotenie by malo byť odsúhlasené všetkými členmi tímu.

2.5.1 Product Owner

Stretnutie s product ownerom začína s pripomenutím úloh pre aktuálny šprint a skontrolovaním aktuálne rozpracovaných a dokončených úloh. Počas stretnutia s product ownerom sa robí backlog grooming.

User Story sú zoradené v Backlogu podľa priority, túto priority User Story určuje product owner počas stretnutia na základe jej komplexity a užitočnosti. Komplexitu User Story určujú členovia tímu, na určovanie komplexity User Story sa používa planning poker. Pri určovaní komplexity User Story prebieha komunikácia medzi členmi a planning poker sa opakuje kým sa členovia nedohodnú. Ak User Story nemá splnenú Definition of Ready, je možné ju rozdeliť na viaceré User Story. Každá User Story pred zaradením do šprintu musí byť rozdelená na Tasky. Do Taskov je potrebné definovať estimating hours, estimating hours by nemal byť väčší ako 8 hodín. Ak Task je časovo náročnejší ako 8 hodín je potrebné ho rozdeliť na menšie Tasky a rozložiť medzi viacerých členov tímu.

2.5.2 Právomoc Scrum Mastera

Scrum Master má právo zakázať témy diskusie alebo nejaký problém, ktorý aktuálne riešia členovia tímu. Scrum Master má právo vyhlásiť prestávku v rozsahu 10 až 30 minút, prestávky by mali byť odsúhlasené členmi tímu.

2.6 Po šprinte

Členovia tímu po šprinte spoločne oddiskutujú čo stihli, čo spôsobovalo problémy a čo by sa dalo zlepšiť.

2.7 Definitions

2.7.1 Definition of Done

Definition of Done je v tíme definovaný:

- Projekt bol odprezentovaný product ownerovi.
- Pre User Story sú splnené akceptačné kritériá product ownera.
- User Story bola schválená product ownerom.
- Kód obsahuje zmysluplné komentáre.
- Kód bol skontrolovaný metódou code review aspoň od jedného ďalšieho člena tímu.

2.7.2 Definition of Ready

Definition of Ready je v tíme definovaný:

- Pre User Story:
 - Musí byť zaradená do šprintu.
 - Musí mať zvolenú zodpovednú osobu.
 - Názov musí byť zrozumiteľný.
 - Komplexita User Story musí byť ohodnotená tímom a zapísaná ako *Story points*.

- Pre Task:
 - Musí byť zaradený do šprintu.
 - Musí mať zvoleného riešiteľa úlohy.
 - Musí obsahovať krátky zrozumiteľný popis.
 - Názov musí byť zrozumiteľný.
 - Náročnosť je odhadovaná počtom hodín v *Original Estimate* a *Remaining*.
 - Úloha musí odkazovať na User Story, pod ktorým sa nachádza.

3 Metodika komunikácie

3.1 Dedikácia

Metodika opisuje dohodnuté spôsoby komunikácie, ktoré zaužívame v tíme č. 16 Radiant.

3.2 Komunikačné nástroje

V našom tíme používame komunikačný nástroj Slack a Discord, každý člen tímu by mal byť pripojený na Slacku alebo Discorde vždy keď to je možné, ideálne mať nainštalovaný Slack alebo Discord aj v mobile, aby členovia tímu boli rýchlo dosiahnuteľní. V komunikačnom nástroji Slack náš tím používa niekoľko komunikačných kanálov pre potreby komunikácie v tíme. Počas stretnutia sme napojení na spoločnom konferenčnom hovore na komunikačnom nástroji Discord.

Aktuálne komunikačné kanály sú:

- **general** - všeobecný kanál na riešenie rôznych problémov ohľadom tímu
- **stretnutie** - tento komunikačný kanál sa používa na komunikáciu tímu počas stretnutia
- **tfs** - tu dostávame všetky notifikácie z TFS
- **status-report** - Každý večer medzi 20:00 a 24:00 do tohto komunikačného kanálu napíšu povinne všetci členovia status report čo za aktuálny deň spravili alebo alternatívne stačí svoj postup za deň oddiskutovať na Discorde s ostatnými členmi tímu. Prostredníctvom tohto reportu dávajú členovia tímu najavo ostatným členom na čom aktuálne pracujú a taktiež to slúži ako spätná väzba pre Scrum Mastera.
- **tp-cup** - komunikačný kanál, ktorý slúži výhradne pre TP Cup.

3.3 Spoločný email:

Okrem Slacku a Discordu používame spoločný e-mail, ku ktorému ma každý člen tímu prístup:

`tp16.radiant [at] gmail.com`

Vo výnimočných situáciach každý člen tímu má k dispozícii telefónny kontakt na každého člena tímu.

3.4 Komunikácia s product ownerom

S product ownerom komunikujeme cez nasledovné spôsoby:

- Počas stretnutia cez Google Meet.
- Slack
- E-mail - V prípade dôležitých e-mailov, ktoré nie sú priamo adresované product ownerovi, tak je zaužívané pridávať product ownera do CC.

4 Metodika verziovania

4.1 Dedikácia

Metodika opisuje zásady a postupy verziovania, ktoré sú zaužívané v tíme č. 16 Radiant.

4.2 Zaužívané pojmy

Zdrojový kód a komentáre musia byť napísané v anglickom jazyku, ostatné jazyky sú zakázané.

- *Commit* - zmeny v repozitári
- *Branch* - samostatná vetva projektu, ktorá umožňuje paralelnú prácu v tíme
- *Push* - nahratie zmien z lokálneho repozitára do vzdialeného
- *Pull* - stiahnutie zmien do lokálneho repozitára zo vzdialeného
- *Merge* - spája dve alebo viacero úprav z vetiev do jednej

4.3 Repozitár

Náš tím má vytvorený vlastný repozitár na prácu na tímovom projekte, k tomuto repozitáru majú prístup všetci členovia tímu. Repozitár nášho tímu je súkromný a na jeho prístup je potrebné povolenie od správcu.

4.4 Branch

V našom tíme pri práci rozlišujeme dva hlavné druhy vetiev a to master a development.

Vetva master predstavuje hlavnú vetvu projektu, z tejto vetvy vznikajú ďalšie vetvy, ktoré obsahujú ďalšie funkčné časti projektu. Táto vetva musí byť vždy aktuálna a všetky jej pridané časti musia byť stabilné a funkčné, čiže do master vetvy sa môže pridávať iba funkčný, stabilný a otestovaný kód.

Vetva development predstavuje rozpracovanú vetvu projektu na ktorej sa aktuálne pracuje, táto vetva sa neustále mení, spravidla sa upravuje, spústa a testuje lokálne. V určitých momentoch sa vetva development pridáva do master vetvy, spravidla to je na konci šprintu. Pri takomto pridávaní do master vetvy, musí byť vetva plne funkčná, otestovaná a prejdená code review. Development vetva pri pridávaní do master vetvy nemôže obsahovať bugy a musí byť stabilná.

Členovia nášho tímu si môžu z development vetvy vytvárať ďalšie menšie vetvy, ktoré umožňujú paralelnú prácu na úlohách, tieto menšie vetvy predstavujú prácu na User Story.

Poznámka: pred prácou je potrebné sa uistiť, že sa pracuje s aktuálnou verziou vetvy, v prípade, že nie je aktuálna, je potrebné ju aktualizovať.

Pomenovanie vetiev: náš tím sa riadi podľa konvencie gitflow, čiže názov vetvy dodržiava nasledovný formát: [vetva]/[názov]

Označenie vetiev môže byť:

- *feature* - časť produktu
- *release* - nasadenie otestovanej časti produktu
- *hotfix* - oprava chybnej časti kódu

4.5 Commit správy

Pri písaní správ pre commity je potrebné aby boli efektívne, čiže aby zahŕňali informačnú hodnotu a aby bolo z nej jasné čo daný commit robí. Je potrebné aby každý člen tímu pri písaní commit správ sa snažil aby správa bola stručná a aby vystihovala príslušné zmeny.

Preto je potrebné dodržiavať nasledovné pravidlá:

- Správa musí mať maximálne 50 znakov, v prípade jej prekročenia je potrebné commit rozdeliť.
- Správa začína veľkým písmenom.
- Pre ukončenie správ sa na konci nepíše bodka.
- Správa musí byť napísaná v slovenskom jazyku.

Príklad dobrej správy:

“Oprava chyby pri vkladaní súčiastok do gridu”

Príklad zlej správy:

“opravil som nejaké chyby”

4.6 Mergovanie vetiev

Pri spájaní vetiev je potrebné sa riadiť nasledujúcimi pravidlami:

- Člen tímu, ktorý ide spájať vetvy sa musí najprv uistiť, že sa nachádza vo vetve, ktorú ide spájať.
- Kód v danej vetve sa ešte raz skompiluje a uistí, že funguje správne.
- V prípade, že sa nájdu nejaké problémy alebo chyby počas testovania, je potrebné túto chybu nahlásiť členovi tímu, ktorý na danej vetve pracoval a čo najrýchlejšie ju opraviť, bez opravy nemôžu sa vetvy spojiť.
- Po spojení je potrebné vyriešiť prípadne vzniknuté konflikty.