

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4



Riadenie projektu

**Dalibor Turay, Kristián Košťál, Patrik Krajča, Patrik
Pernecký, Peter Radványi, Roman Kopšo, Vladimír
Čápka**

Študijný program: Softvérové inžinierstvo

Ročník: 1, Krúžok: Po 16:00, U120

Predmet: Tímový projekt

Vedúci: Ing. Rastislav Bencel

Ak. rok: 2015/16

Obsah

1	BIG PICTURE	1
1.1	Úvod.....	1
1.2	Členovia tímu.....	1
1.3	Motivácia	2
1.4	Náplň projektu	3
1.5	Ciele celého projektu	3
1.6	Aplikácia manažmentov	4
1.7	Sumarizácie šprintov.....	4
1.8	Používané metodiky.....	5
1.9	Podiel práce na jednotlivých dokumentáciách.....	5
1.10	Globálna retrospektíva zimného semestra	6
2	MANAŽMENT KOMUNIKÁCIE	7
2.1	Úvod.....	7
2.2	Použité nástroje.....	7
2.3	Metodika komunikácie	7
3	MANAŽMENT PLÁNOVANIA	9
3.1	Úvod.....	9
3.2	Použité nástroje.....	9
3.3	Metodiky plánovania	9
4	MANAŽMENT KVALITY	16
4.1	Úvod.....	16
4.2	Code review	16
5	MANAŽMENT RIZÍK	17
5.1	Úvod.....	17
5.2	Tabuľka rizík	17
6	MANAŽMENT SOFTVÉRU	20
6.1	Úvod.....	20
6.2	OpenFlow.....	20
7	MANAŽMENT TESTOVANIA	38

7.1	Úvod.....	38
7.2	Použité nástroje.....	38
7.3	Testovanie v mininete.....	38
7.4	Testovanie na fyzickej architektúre	39
8	MANAŽMENT DOKUMENTÁCIE	41
8.1	Úvod a popis	41
8.2	Metodika dokumentácie.....	41
9	OSTATNÉ METODIKY.....	50
9.1	Metodika zápisníc	50
9.2	Metodika k stretnutiam	51
9.3	Metodika nasadzovania.....	52
10	ZÁZNAMY ZO STRETNUTÍ.....	54
10.1	Zápisnica z 1. stretnutia	54
10.2	Zápisnica z 2. stretnutia	56
10.3	Zápisnica z 3. stretnutia	58
10.4	Zápisnica zo 4. stretnutia	60
10.5	Zápisnica z 5. stretnutia	62
10.6	Zápisnica zo 6. stretnutia	64
10.7	Zápisnica zo 7. stretnutia	66
11	EXPORTY ÚLOH	68
11.1	Šprint 1.....	68
11.2	Šprint 2.....	68
11.3	Šprint 3.....	69

1 Big picture

1.1 Úvod

Dokumentácia má v sebe popísané akým spôsobom sme manažovali a riadili chod nášho projektu. Máme tu obsiahnuté rôzne metodiky, spôsoby a pravidlá ktorými sme sa riadili v oblastiach komunikácie, testovania, softvéru, plánovania, kvality, rizík a dokumentácie. Čitateľ sa aj dozvie akí ľudia tvoria náš tím a bude mať aj dobrý prehľad o jednotlivých tímových stretnutiach čo sa dozvie vďaka našim zápisniciam.

Úlohou nášho projektu je vytvoriť funkčnú softvérovo definovanú sieť (SDN), ktorá podporuje plynulý prechod koncového zariadenia medzi dvoma wifi prístupovými bodmi (AP – access point). Táto téma je výskumného charakteru a na jej riešenie sme museli aj prispôbiť náš prístup k riešeniu projektu. Všetky podrobné informácie sa dočítate v nasledujúcich kapitolách.

1.2 Členovia tímu

Bc. Dalibor Turay: Skúsený sieťar, ktorý sa vyzná vo svojej oblasti. Momentálne pracuje ako analytik kde sa už stretol s procesom vývoja produktu. Jeho znalosti z praxe sú veľmi nápomocné a má rád zaznamenávanie všetkých projektových udalostí. V tíme je naším manažérom dokumentovania.

Bc. Kristián Košťál: Kiki je človek, ktorý dokáže spojzdníť a otestovať rôzne typy softvéru či hardvéru. Rád sa hlavne hrá s hardvérom a v našom prípade sú to wifi routre. V nečakaných situáciách dokáže prakticky riešiť problémy, veď koniec koncov je to skaut. V našom tíme zohráva úlohu manažéra testovania.

Bc. Patrik Krajča: Paco má veľkú hlavu na matematické vzorce a úlohy. Ako žiak na ZŠ a gymnáziu chodil na matematické olympiády kde dominoval. Tieto znalosti mu zabezpečili prácu v Accenture ako softvérový vývojár. Má rád keď sú veci matematicky čisté a dokonalé. Je naším manažérom kvality.

Bc. Patrik Pernecký: Paťo má veľmi dobrý prehľad o tej správnej kombinácii technológií, ktoré použijeme v našom projekte. Je to univerzálny znalec, ktorý skôr preferuje hardvérovú časť práce.

Okrem toho pozná ako fungujú rôzne komunikačné wifi protokoly, ktoré budeme testovať a používať. Je hlavným architektom projektu a manažérom komunikácie.

Bc. Peter Radványi: Peťo sa vyzná v programovaní pekného a čitateľného kódu. V našom projekte budeme hlavne pracovať s programovacím jazykom C++ a skriptovacím jazykom Python, ktoré on vie perfektne použiť. Má schopnosti kódového guru a predstavuje softvérový sval tímu. Je naším vývojárom a manažérom softvéru.

Bc. Roman Kopšo: Roman sa snaží do tímu vkladať energiu a motiváciu. Je presvedčený, že každý člen má veľký potenciál, ktorý treba naplniť. Je iniciatívny a chce, aby náš tím vystupoval z radu. Má solídne znalosti sieťových a softvérových technológií. Dáva pozor akým smerom ideme, riadi chod projektu a plánuje do budúcnosti. Jeho úlohou je manažment plánovania a je aj scrum masterom.

Bc. Vladimír Čápka: Vlado je naša cudzinecká légia. Prišiel z Materiálovo-technologickej fakulty, aby nám pomohol pri naplnení nášho cieľa. Napriek tomu, že neštudoval PKSS ako my všetci ostatní, nás dokázal prekvapiť ako sa flexibilne vedel adaptovať na naše prostredie. Vlado sa pri našej práci snaží redukovať riziká a má svoje plány B až po Z. Je manažér rizík.

1.3 Motivácia

V našom projekte sa snažíme dosiahnuť plynulý prechod wifi koncového zariadenia medzi dvoma prístupovými bodmi v softvérovo definovanej sieti. Inými slovami keď niekto pomocou mobilu uskutočňuje Skype hovor prostredníctvom wifi vo veľkej budove, tak hovor môže mať nízku kvalitu. Keď sa tento človek prechádza po budove, tak sa vzd'ahuje a približuje medzi rôznymi wifi anténami, ktoré predstavujú prístupové body do siete (AP). Tento pohyb núti jeho mobil odpojiť sa od vzdialenej AP a pripojiť sa ku AP, ktorá je bližšie. Toto spôsobuje stratu toku dát, ktorá má na svedomí nekvalitný telefónny hovor. Samozrejme telefónny hovor je len jeden príklad tejto problematiky. Všade tam kde je nutná nepretržitá komunikácia sa dá naše riešenie využiť (napr. ovládanie skladových robotov).

Keď sa nám podarí dokončiť projekt, tak tento problém by mal byť eliminovaný. Vedeli by sme totiž zaistiť, aby AP sa vedeli dopredu pripraviť na odpojenie a pripojenie koncového zariadenia tak, aby nenastala strata dát. Okrem iného, toto všetko sa snažíme implementovať v softvérovo definovaných sieťach (SDN). Otázka teraz znie, prečo práve SDN? V sieťovom svete predstavujú novinku, ktorá

má veľké výhody a dokonca samotný Google ich začal používať. Všetko je to totiž o prehľade a cene. V súčasnosti fungujú naše siete ako distribuovaný systém. Je to niečo a graf s uzlami a hranami kde uzol predstavuje komunikačný komponent ako router alebo switch. Tieto zariadenia rozmýšľajú samy za seba a pri výpadku linky (hrana) sa musia reorganizovať, aby bolo znovu nadviazané spojenie. Tieto uzly nemajú dobrú predstavu o celej sieti a okrem toho sú drahé. V cene je totiž špičkový hardvér a aj softvér, ktorý čo najrýchlejšie preposiela správy. V prípade SDN je sieť centralizovaná. Uzle sú tentokrát lacné routre, ktoré „nemajú mozog“ a rozmýšľa za nich centrálny kontrolór. Týmto spôsobom sa ušetrí na zariadeniach a máme prehľad o celej sieti z centrálného bodu. Na záver iba spomeniem, že pri aktualizovaní/konfigurovaní SDN stačí z jedného bodu všetko nastaviť no a na druhej strane v štandardných sieťach treba navštíviť každý jeden bod v sieti.

1.4 Náplň projektu

V našom projekte pracujeme hlavne s virtuálnymi strojmi na simulovanie už skôr spomínaného prechodu. Študujeme wifi komunikačné protokoly, SDN softvérové kontrolóry a spôsob komunikácie v SDN. Pripravujeme zariadenia aj na fyzické testovanie. Testovacie prostredie predstavujú dva wifi routre, na ktorých je nainštalovaný softvér OpenVswitch, vďaka ktorému môžeme uskutočňovať komunikáciu v SDN pomocou protokolu OpenFlow 1.3. Na tieto routre bude pripojený server s nainštalovaným SDN kontrolórom RYU. Pomocou počítača cez SSH tunel budeme pristupovať na tento server a konfigurovať našu malú sieť. Na kontrolór a wifi routre budeme vytvárať Python skripty, ktorými sa budeme snažiť uskutočniť plynulý prechod. Fyzické aj softvérové simulácie budú sledované čo sa týka prenosu dát, aby sme zistili či máme/nemáme stratu dát. Výstup projektu bude úspešné implementovanie softvéru do SDN na zaistenie plynulého prechodu bez modifikácie koncového zariadenia.

1.5 Ciele celého projektu

Poskytnutie wifi pripojenia v rámci SDN

- *Požiadavky:* server, wifi router, SDN kompatibilný softvér
- *Postup:* inštalácia a konfigurácia SDN softvéru na server a následne na router

Zaistiť plynulý prechod medzi AP v rámci SDN

- *Požiadavky:* server, 2 wifi routre, SDN kompatibilný softvér, program na sledovanie toku dát, koncové zariadenie (mobil), Python editor
- *Postup:* inštalácia a konfigurácia SDN softvéru na server a následne na routre, nastavenie sledovania toku dát, skriptovanie v jazyku Python, testovanie kódu a fyzického prechodu

Urobiť prechod bezpečným

- *Požiadavky*: vyššie spomenuté
- *Postup*: pokračovanie predošlého bodu, penetračné testovanie nášho riešenia a zaplátavanie identifikovaných bezpečnostných dier

1.6 Aplikácia manažmentov

Opis jednotlivých činností potrebných pre riadenie projektu, procesu a produktu, tak ako sa realizovali.

Typ manažmentu	Zodpovedná osoba	Stručný popis
Manažment komunikácie	Bc. Patrik Pernecký	Riadenie a použitie rôznych komunikačných kanálov pre koordináciu tímu.
Manažment plánovania	Bc. Roman Kopšo	Plánovanie úloh do budúcnosti pre jednotlivých členov tímu, aby sa spĺňali termíny.
Manažment kvality	Bc. Patrik Krajča	Kontrola kódu a zoznam pravidiel pre zachovanie čitateľnosti a robustnosti softvéru.
Manažment rizík	Bc. Vladimír Čápka	Predpovedanie projektu nebezpečných rizík a alternatívy keby nejaké riziko nastalo.
Manažment softvéru	Bc. Peter Radványi	Použitie nástroje pre tvorbu softvéru, tímová synchronizácia pri vytváraní kódu.
Manažment testovania	Bc. Kristián Košťál	Spôsoby a pravidlá pri testovaní našich riešení.
Manažment dokumentácie	Bc. Dalibor Turay	Pravidlá a spoločný formát pre jednotlivé dokumentácie dôležité pre projekt.

Tab.č.1 – Rozdelenie manažmentov

1.7 Sumarizácie šprintov

Šprint 1

Bol úspešne ukončený. Všetky úlohy boli naplnené a tento šprint bol hlavne analyzačného charakteru.

Šprint 2

Úspešne ukončený s menšími problémami v úprave user stories. Bol hlavne verifikačného typu, spolu s analýzami a s pokusmi dať do prevádzky softvérový kontrolór spolu s firmvérom na wifi router.

Šprint 3

V súčasnosti sa na ňom pracuje. Ide hlavne o testovanie SDN funkcionalít SDN kontrolóra RYU. Pracujeme aj na návrhu virtuálneho AP.

Podrobnejšie informácie o jednotlivých šprintoch sa dajú nájsť v kapitole 10 zo zápisníc, ďalej z kapitoly 11 z exportov úloh. V druhej dokumentácii *Inžinierske dielo* sú jednotlivé šprinty rozobrané do najväčších detailov.

1.8 Používané metodiky

Nasleduje tabuľka so zoznamom použitých metodík, ku ktorým sme napísali aj krátky popis.

Názov metodiky	Popis metodiky
Metodika plánovania	Predstavuje pravidlá pre vytváranie úloh a plánovanie aktivít v rámci tímu.
Metodika dokumentácie	Súhrn pravidiel pre dokumentovanie jednotlivých častí projektu.
Metodika zápisníc	Spôsob a formát zapisovania jednotlivých zápisníc z tímových stretnutí.
Metodika tímových stretnutí	Použitá ako štandard pri predávaní informácií a spôsob brífingov na tímových stretnutiach.
Metodika nasadzovania	Pravidlá pre nasadzovania rôznych častí systému.
Metodika komunikácie	Má v sebe naše dohodnuté štandardy komunikačných kanálov.

Tab.č.2 – Zoznam použitých metodík

1.9 Podiel práce na jednotlivých dokumentáciách

Obsahuje dve tabuľky, na ktorých je vidieť podiel práce na jednotlivých dokumentoch.

Časť dokumentu „Riadenie projektu“	Zodpovední
Big picture	Bc. Roman Kopšo
Manažment komunikácie	Bc. Patrik Pernecký
Manažment plánovania	Bc. Dalibor Turay a Bc. Roman Kopšo
Manažment kvality	Bc. Patrik Krajča
Manažment rizík	Bc. Vladimír Čápka
Manažment softvéru	Bc. Peter Radványi a Bc. Kristián Košťál

Manažment testovania	Bc. Kristián Košťál a Bc. Roman Kopšo
Manažment dokumentácie	Bc. Dalibor Turay
Ostatné metodiky	Bc. Dalibor Turay
Zápisnice zo stretnutí	Bc. Dalibor Turay
Exporty úloh	Bc. Roman Kopšo

Tab.č.3 – Podiel práce na dokumente Riadenia

Za spájanie, kontrolu, úpravy a tvorbu uceleného formátu dokumentu „Riadenie projektu“ zodpovedá Bc. Roman Kopšo.

Časť dokumentu „Inžinierske dielo“	Zodpovední
Úvod	Bc. Roman Kopšo
Ciele	Bc. Dalibor Turay
Šprint 1	Bc. Vladimír Čápka
Šprint 2	Bc. Patrik Krajča
Šprint 3	Bc. Peter Radványi
Použité technológie	Bc. Patrik Pernecký
Architektúra	Bc. Patrik Pernecký

Tab.č.4 – Podiel práce na dokumente Inžinierske dielo

Za spájanie, kontrolu, úpravy a tvorbu uceleného formátu dokumentu „Inžinierske dielo“ zodpovedá Bc. Dalibor Turay.

1.10 Globálna retrospektíva zimného semestra

Táto časť bude dokončená na konci zimného semestra.

2 Manažment komunikácie

2.1 Úvod

V manažmente komunikácie sa zameriavame na komunikáciu medzi jednotlivými členmi tímu, a na postupy akými jednotliví členovia komunikujú v použitých nástrojoch. Je tiež potrebné spomenúť aké nástroje na komunikáciu používame v tíme.

2.2 Použité nástroje

Trello – tento nástroj nepoužívame primárne na komunikáciu, avšak pri jednotlivých úlohách je niekedy potrebné sa na niečom dohodnúť, aké problémy vznikli a podobne.

Skupina na facebooku – skupinu sme používali len zo začiatku projektu. Hlavným dôvodom zmeny bolo rušivé prostredie ako napríklad iné konverzácie, mini hry, články a podobne.

Hipchat – v terajšom čase náš primárny komunikačný nástroj. Spoločne sa na ňom rozoberajú všetky úlohy, vzniknuté problémy a ich následné riešenia a ostatné veci ktoré vznikajú pri projekte.

Komunikácia na stretnutiach – na stretnutiach sa vždy najprv referujú naštudované materiály, a potom sa rozoberajú problémy ktoré vznikli pri riešení daných úloh.

2.3 Metodika komunikácie

Metodiku komunikácie môžeme rozdeliť na dve časti a to na komunikáciu na stretnutiach a na elektronickú komunikáciu. Rozdelili sme si túto metodiku preto, lebo na stretnutiach prebieha komunikácia odlišne ako pri elektronickej komunikácii.

Komunikáciu na stretnutiach môžeme rozdeliť na nasledujúce kroky:

1. Postupne každý člen tímu ktorý si naštudoval niečo nové k projektu to zreferuje celému tímu.
2. Nasleduje diskusia na danú tému.
3. Ako ďalšie každý člen tímu povie ako splnil svoje úlohy na daný týždeň.
4. Ak nastal nejaký problém pri plnení úlohy, tak sa tento problém zapíše a uloží do možných rizík.
5. Nasleduje diskusia k úlohám.
6. Ako predposledné si rozdelíme ďalšie úlohy na nasledujúci týždeň.
7. Na záver stretnutia prebieha voľná diskusia k téme, pri ktorej sa rozpráva o možných zlepšeniach projektu a podobne.

Elektronická komunikácia sa trochu líši od komunikácií na stretnutiach a to hlavne z toho pohľadu, že nie je periodická, takže kedykoľvek môže člen tímu napísať na hipchat a problém sa hneď začne riešiť. Z tohto pohľadu je náročné ustanoviť nejakú metodiku k elektronickej komunikácií, avšak je vždy treba zaviesť pravidlá ktoré je treba dodržiavať. Samotnú elektronickú komunikáciu si môžeme rozdeliť na nasledujúce kroky a to:

1. Každý člen tímu je povinný sa zdržiavať na hipchate čo najviac času, aby mohol v prípade problému pomôcť členovi tímu.
2. V prípade že má člen tímu problém pri plnení úlohy, napíše tento problém na hipchat po prípade do danej úlohy na trello.
3. Ak člen tímu vie odpovedať na vzniknutý problém čo najrýchlejšie odpovie na tento problém do hipchatu.
4. Ak člen tímu zistí nejaké podstatné informácie k projektu hneď ich napíše do trello.
5. K jednotlivým novým informáciám k projektu prebieha diskusia na hipchate, kde sa členovia tímu dohadujú, či by sa daná technológia dala použiť v našom projekte.

3 Manažment plánovania

3.1 Úvod

Manažment plánovania sa zaoberá plánovaním úloh v rámci celého životného cyklu projektu. Úlohy sa plánujú na jednotlivé šprinty a každá úloha má priradeného jedného, alebo viacerých riešiteľov. Každá úloha má definovaný predpokladaný čas trvania. V priebehu práce na úlohe si riešiteľ úlohy zaznamenáva čas strávený na danej úlohe. Tým sa dosiahne viditeľná práca jednotlivých členov tímu za danú úlohu. Pomocou manažovacieho nástroja je možné vytvoriť report za šprint a tým určiť prácu jednotlivých členov tímu v rámci jedného šprintu. Plánovanie úloh sa vykonáva na stretnutí, kde sa k jednotlivým úlohám priradí riešiteľ a stanoví sa predpokladaný čas trvania úlohy.

3.2 Použité nástroje

Trello

V rámci projektu bol vybraný členmi tímu nástroj *Trello*. *Trello* plne postačuje funkcionalitou pre náš projekt. Je v ňom možné vytvárať šprinty a v rámci šprintov definovať úlohy, ktoré sa budú nachádzať v rôznych stavoch. Nástroj umožňuje presúvať jednotlivé úlohy v zoznamoch, čo umožňuje jednoduchú prácu.

Plus for Trello

K Trelu bol pridaný modul *Plus for Trello*, pomocou ktorého sa zaznamenávajú časy a vytvárajú reporty. Pridaním modulu bolo možné pridávať predpokladané časy na úlohu, pridávať časy strávené na úlohe riešiteľom, vytvárať burndown grafy a vytvárať reporty. Tým sa výrazne zjednodušil prehľad práce jednotlivých členov a aj zobrazenie ako sa vyvíja projekt.

3.3 Metodiky plánovania

Vytváranie úloh

Štandardne, úlohy venované pre konkrétny šprint si prekonzultujeme na našom tímovom stretnutí. Dohodneme sa, na ktorých veciach treba zapracovať, kto ich bude riešiť a koľko času by mali trvať. Ako už bolo spomenuté v časti *Použité nástroje*, na manažment úloh používame pomôcku *Trello*, ktorá síce neponúka toľko možností ako napríklad *JIRA*, ale pre naše potreby nám stačí. Dajú sa z nej exportovať reporty a dá sa aj sledovať počet človeko-hodín, čo nám vytvára burndown grafy.

Dokonca nedávno *Trello* oznámilo aktualizáciu s tým, že ponúka integráciu viacerých manažment pomôcok ako napríklad komunikačný systém Slack.

Postup tvorenia úloh pre šprint je nasledovný:

1. V Trelle si vytvoríme tzv. Board, čo je akási tabuľka, do ktorej sa dajú pridávať kartičky s úlohami. Každý jeden Board má meno „Šprint x“ kde x predstavuje číslo konkrétneho šprintu.
2. V tabuľke daného šprintu sa vždy vytvorí štyri stĺpcové zoznamy, do ktorých sa pridávajú jednotlivé kartičky. Tieto zoznamy sú:
 - *To do*: obsahuje zoznam naplánovaných úloh, na ktorých sa ešte nezačalo pracovať.
 - *Doing*: obsahuje zoznam úloh, na ktorých sa momentálne pracuje.
 - *Done*: obsahuje zoznam ukončených úloh, ktoré spĺňajú definíciu ukončenia úlohy.
 - *Informácie*: obsahuje ostatné informácie ako napríklad miesto/čas stretnutia tímu, odkazy na zaujímavé stránky a iné veci, ktoré sa netýkajú priamo úloh, ale ktoré chceme dať do pozornosti.
3. Spoločne začneme konzultovať a postupne vytvárame jednotlivé úlohy, ktoré pridávame do zoznamu *To Do*. Zároveň pridelíme jednotlivých ľudí na jednotlivé úlohy. Vieme, že v Scrume by mala mať jedna úloha jedného riešiteľa, ale v našom prípade niekedy dávame aj viacerých ľudí na jednu úlohu. Pokiaľ nastanú problémy pri konkrétnej úlohe, tak sa pridá ďalší človek do počtu (musí byť voľný), aby pomohol ostatným.
4. Odhadneme náročnosť úloh na základe čoho im priradíme určitý počet hodín. Používame plugin *Plus Trello* vďaka ktorému pridávame tieto hodiny. Formát hodín je S/E(R). S predstavuje počet strávených hodín (spent), E predstavuje odhadovaný čas na úlohu (estimated) a R predstavuje zvyšok hodín (remaining). Nakoniec ešte úlohám priradíme ich termín splnenia.

Sledovanie úloh

Keď máme všetky úlohy pripravené, tak tabuľka vyzerá tak, že stĺpec *To Do* obsahuje všetky úlohy, zatiaľ čo ostatné stĺpce sú prázdne. Keď niekto začne pracovať na úlohe, premiestni danú úlohu zo stĺpca *To Do* do *Doing*. Takto vieme, že už niekto sa danej úlohe venuje a postupne si zapisuje koľko hodín pracoval danej úlohe. Príklad časového formátu pre úlohu, ktorá má náročnosť 5 hodín je nasledovná (formát je S/E(R)):

- 0/5(5)

Najprv má úloha 0 strávených hodín, 5 odhadovaných a 5 zostávajúcich hodín. Tento stav je klasický pre všetky úlohy, ktoré sa nachádzajú v *To Do*.

- 2/5(3)

Keď sa úloha presunie do stĺpca *Doing* vieme, že už niekto na nej pracuje a priebežne môže zapisovať strávené hodiny. V tomto prípade daný člen tímu strávil, nad danou úlohou 2 hodiny z 5 hodín a zostávajú mu ešte 3 hodiny.

- 5/5(0)

Keď niekto dokončí úlohu tak dopíše zvyšné hodiny a premiestni úlohu zo stĺpca *Doing* do *Done*. Toto ale platí, keď všetci riešitelia danej úlohy naplnili svoje hodiny. V tomto prípade bolo strávených 5 hodín z 5 a 0 zostáva.

Plugin *Plus Trello* všetky hodiny eviduje a generuje nám pekné grafy a tabuľky vďaka čomu môžeme sledovať akým spôsobom projekt pokračuje.

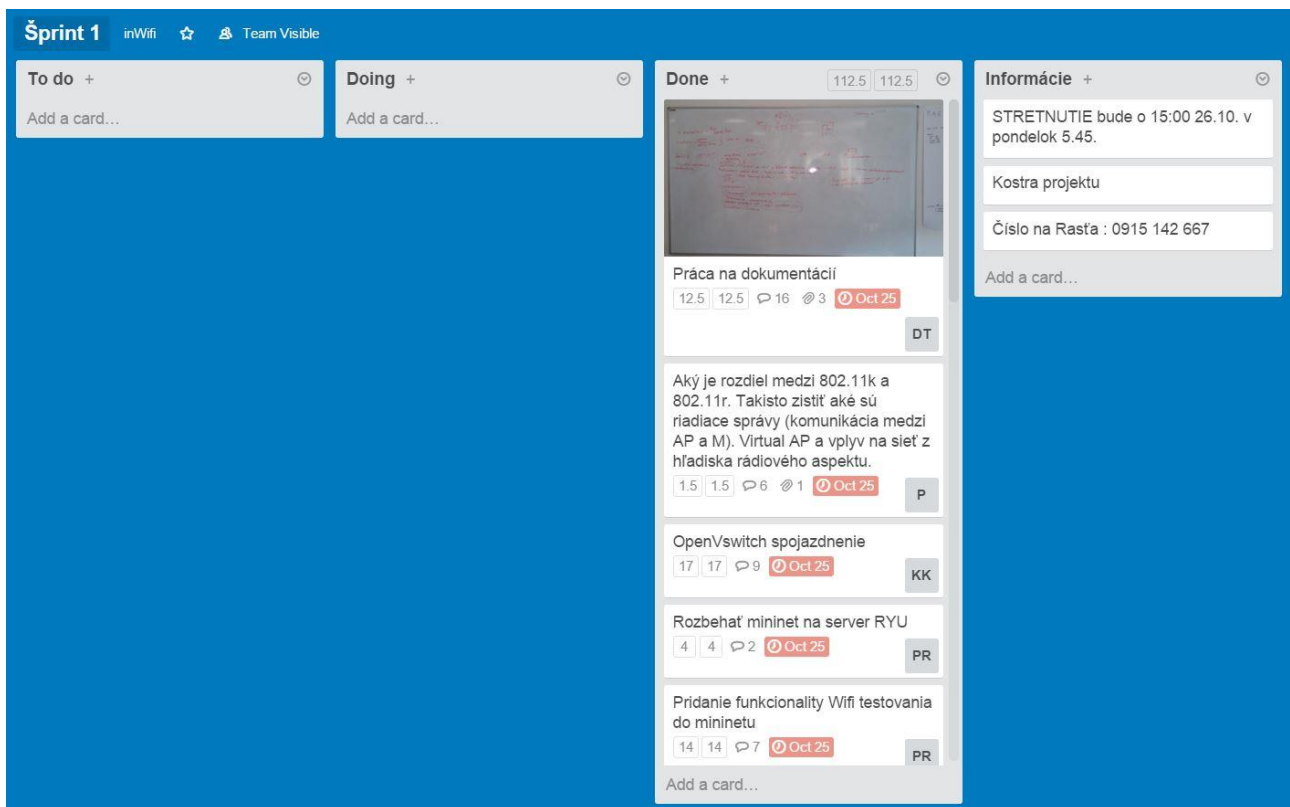
Definícia ukončenej úlohy

Dôležité je dať do pozornosti fakt, že potrebný čas pre riešenie úlohy sa nepriradzuje úlohe, ale jednotlivým riešiteľom. Napríklad máme úlohu na vytvorenie tímovej web stránky a dvoch riešiteľov. Jeden člen tímu sa úlohe bude venovať 3 hodiny (0/3(3)) a druhý zase 5 hodín (0/5(5)). Plugin *Plus Trello* všetkých pekne synchronizuje a danej úlohe dá súčet človeko-hodín, v našom prípade to bude 8 hodín (0/8(8)). Potom pri riešení si postupne budú riešitelia pridávať hodiny až nakoniec stanovený čas pre úlohu bude vyzeráť takto: 8/8(0). Daná úloha sa presunie zo zoznamu *Doing* do *Done* a tím sa považuje za predbežne ukončenú. Na potvrdenie či je úloha skutočne ukončená, podajú riešitelia report na tímovom stretnutí. Ak nie je ukončená, zistíme v čom bol problém a prijmeme určité protiopatrenia.

Môžu nastať aj také problémy, že sme niektorú úlohu zle odhadli a musíme jej priradiť viac času. Dá sa to robiť tak, že pri pridávaní strávených hodín, niekto zadá viac strávených hodín ako odhadovaných a to by vyzeralo napríklad takto: 7/5(-2). Sedem strávených hodín nad úlohou, ktorá mala trvať päť nám dá záporné hodnoty v časti zostávajúcich hodín -2. My ale máme iný prístup a pri zadávaní zmeníme očakávané hodiny z 5 na 7, čiže výsledok bude vyzeráť takto: 7/7(0). Rozhodli sme sa preto takto, lebo záporné hodnoty vo zvyšných hodinách kazia generované burndown diagramy.

Ukážky z aplikácie


Pre lepšiu predstavu sme sem pridali zopár ukážkových obrázkov, aby čitateľ mal lepší prehľad o



Obr.č.1 – Ukážka Trello tabuľky

Na obrázku vyššie vidieť ako napríklad momentálne vyzerá naša tabuľka prvého šprintu. Ako je vidieť tento šprint je ukončený, pretože všetky úlohy, ktoré sa nachádzali v zozname *To Do*, prešli cez stĺpec *Doing* a teraz sa nachádzajú v stĺpci *Done*. V stĺpci *Informácie* máme všeobecné informácie.

Jednotlivé kartičky v stĺpci *Done* predstavujú úlohu. Na kartičke je aj napísaná úloha ako napríklad prvá kartička z hora „Práca na dokumentácii“. Túto úlohu riešil náš manažér dokumentácie Dalo Turay a to je vidieť z malého značenia, ktoré má v sebe Dalove iniciály DT. Ďalej je tam vidieť, nad úlohou strávil 12,5 hodín z 12,5. Termín ukončenia je na 25. októbra.

Práca na dokumentácii in list **Done** Tour 


Recurring add #tags ▾


Members Due Date


DT + Oct 25 at 12:00 PM (past due)

☰ Edit the description...

Attachments


20151012_170444.jpg.jpeg
Added Oct 18 at 12:57 PM
[Download](#) [Remove Cover](#) [Delete](#)


20151012_170538.jpg.jpeg
Added Oct 18 at 12:56 PM
[Download](#) [Make Cover](#) [Delete](#)


20151012_175537.jpg.jpeg
Added Oct 18 at 12:56 PM
[Download](#) [Make Cover](#) [Delete](#)






[Add an attachment...](#)

Add Comment

Card Report - Plus

By User	S sum	E sum (1st)	R	
daloturay	12.5	12.5 (20)	0	modify

RK

Add

- Members
- Labels
- Checklist
- Due Date
- Attachment

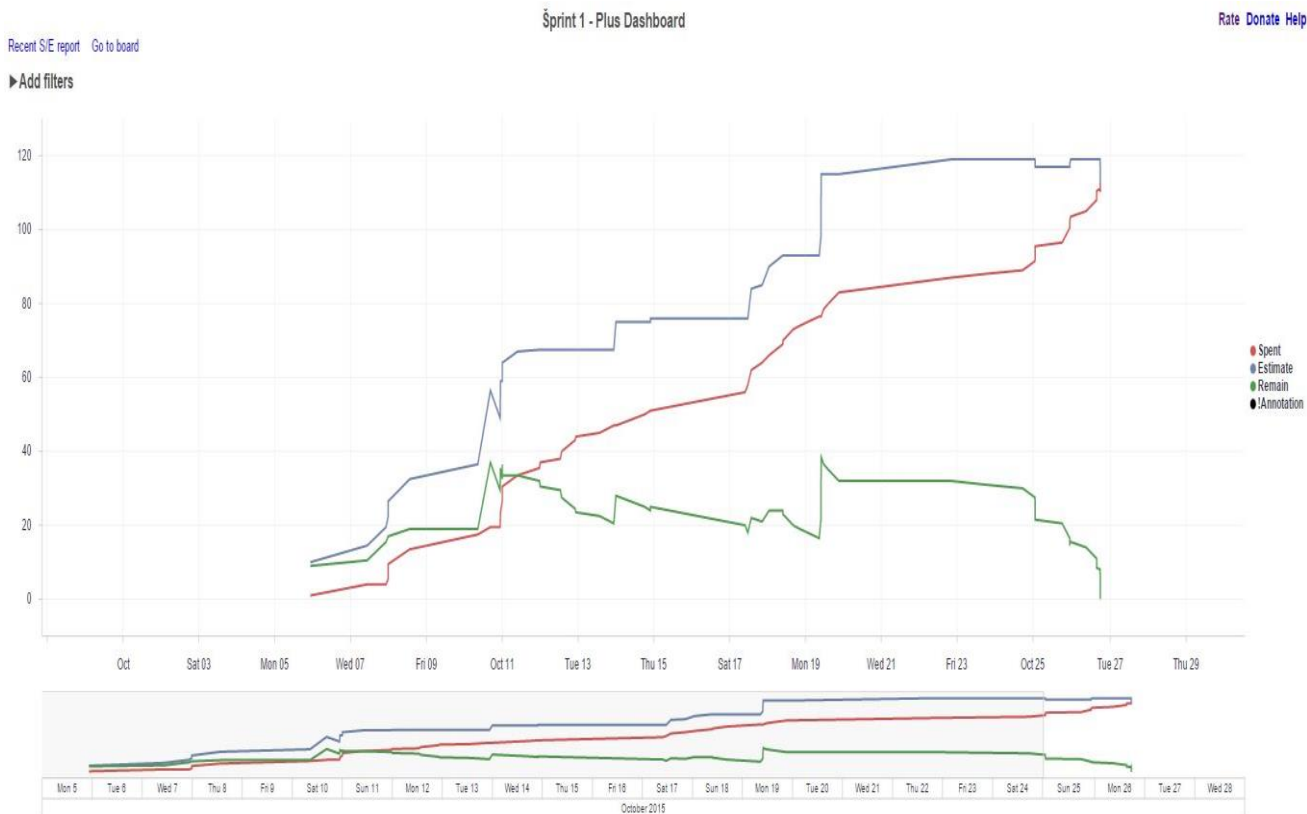
Actions

-
-
-
-
-

[Share and more...](#)

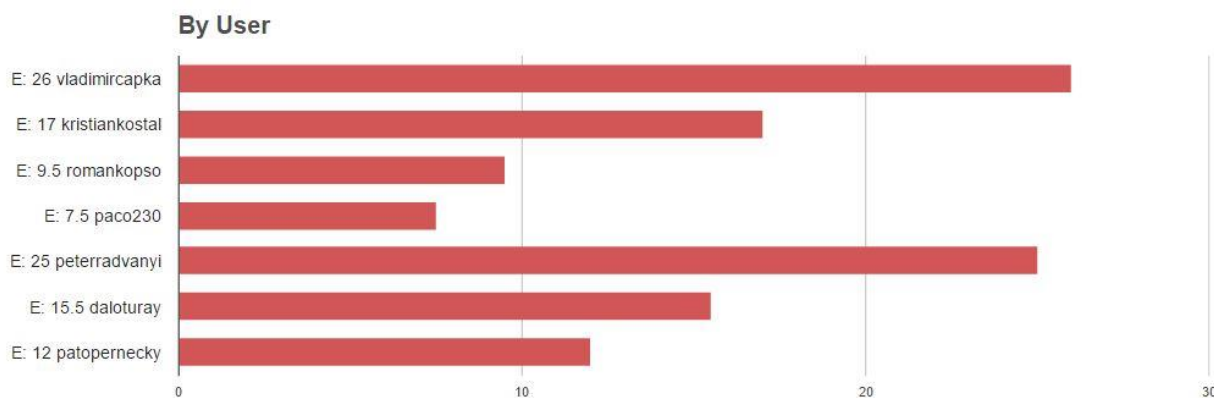
Obr.č.2 – Ukážka editácie jednej kartičky

Výhoda Trella je, že tieto kartičky s úlohami sa dajú bohato editovať ako je vidieť na obrázku vyššie. Dá sa tu komentovať k tejto konkrétnej úlohe, pridávať dôležité súbory potrebné k úlohe ako napríklad fotky zo zápisov z tabule na stretnutiach, pridávať dátumy značenia a iné. My hlavne používame komentovanie úlohy, pridávanie súborov k úlohe a tlačidlo *modify*, ktoré je vidieť nižšie na obrázku. To slúži na pridávanie strávených hodín nad úlohou. K úlohe sa dá pridať termín ukončenia a zároveň sa dajú pridať aj riešitelia.



Obr.č.3 – Generovaný graf práce vynaloženej pre daný šprint

Obrázok číslo 3 ukazuje ako vyzerajú vygenerované diagramy pre daný šprint. Červená čiara predstavuje strávený čas, modrá predpokladaný čas a zelená zostávajúci čas všetkých úloh. Aby bol šprint úspešný, musí červená čiara byť na úrovni modrej, zatiaľ čo zelená musí dosiahnuť úroveň nuly. Okrem toho na stretnutí musia ešte úlohy prejsť overením, že sú naozaj ukončené.



Obr.č.4 – Graf znázorňujúci sumárny počet pracovných hodín každého člena

Plus Trello generuje aj diagramy na ktorých je vidieť koľko každý člen strávil nad úlohami cez jeden týždeň. Na obrázku vyššie sú všetky stĺpce červené čo znamená, že všetci naplnili svoje odhadované hodiny práce. Keby zostali zelené časti stĺpcov, tak tie predstavujú ešte nenaplnené hodiny úloh.

Report	Spent By User	Spent By Board							
20 rows	S: 112.5	E 1 st : 104.5	E: 112.5	R: 0					
Date last	Due date	Week last	Card	Board	List	S	E 1 st	E	R
2015-11-11 23:17	2015-10-25 12:00	2015-W46	Rozdeliť si kompetencie v tíme [R]	Šprint 1	Done				
2015-10-26 17:36	2015-10-25 12:00	2015-W44	Upraviť user stories	Šprint 1	Done	3	3	3	
2015-10-26 17:35	2015-10-25 12:00	2015-W44	Aký je rozdiel medzi 802.11k a 802.11r. Takisto zi...	Šprint 1	Done	1.5	5	1.5	
2015-10-26 17:35	2015-10-25 12:00	2015-W44	Vymyslieť architektúru [R]	Šprint 1	Done				
2015-10-26 17:31	2015-10-25 12:00	2015-W44	Práca na dokumentácii	Šprint 1	Done	12.5	20	12.5	
2015-10-26 15:20	2015-10-25 12:00	2015-W44	Spraviť riziká	Šprint 1	Done	5	5	5	
2015-10-26 15:16	2015-10-25 12:00	2015-W44	Pridanie funkcionality Wifi testovania do mininetu	Šprint 1	Done	14	9	14	
2015-10-26 08:40	2015-10-25 12:00	2015-W44	OpenVswitch spojzdenie	Šprint 1	Done	17	4.5	17	
2015-10-25 22:50	2015-10-25 12:00	2015-W43	Spraviť webovú stránku tímu	Šprint 1	Done	13	10	13	
2015-10-25 22:18	2015-10-25 12:00	2015-W43	Spraviť product backlog	Šprint 1	Done	1	1	1	
2015-10-25 01:24	2015-10-25 12:00	2015-W43	TP cup prihláška	Šprint 1	Done	4	6	4	
2015-10-22 20:07	2015-10-25 12:00	2015-W43	Rozbehať mininet na server RYU	Šprint 1	Done	4	4	4	
2015-10-19 21:10	2015-10-25 12:00	2015-W43	Inštalácia virtuálneho stroja (server)	Šprint 1	Done	15	15	15	
2015-10-19 09:16		2015-W43	STRETNUTIE bude o 15:00 26.10. v pondelok 5.45.	Šprint 1	Informácie				
2015-10-18 20:25	2015-10-25 12:00	2015-W42	IEEE 5 článkov naštudovať	Šprint 1	Done	8	8	8	
2015-10-18 09:38	2015-10-25 12:00	2015-W42	Bezpečnosť na wifi	Šprint 1	Done	3	3	3	
2015-10-13 17:10		2015-W42	Číslo na Rasta : 0915 142 667	Šprint 1	Informácie				
2015-10-12 17:40		2015-W42	Kostra projektu	Šprint 1	Informácie				
2015-10-11 23:36	2015-10-25 12:00	2015-W41	Spojzdenie softvérového kontrolóra RYU	Šprint 1	Done	7.5	7	7.5	
2015-10-07 23:51	2015-10-25 12:00	2015-W41	Analýza dostupných open-source softvérových kontro...	Šprint 1	Done	4	4	4	

Obr.č.5 – Tabuľkový pohľad úloh

Okrem grafov sa nám ešte generujú pekné tabuľky v ktorých sa dá získať pekný prehľad nad procesmi v danom šprinte. Takéto tabuľky sa dajú pekne vyexportovať do excelu a slúžia ako také reporty za daný šprint. Šprinty si takto sledujeme, aby sme si dobre kontrolovali ako sme na tom s časom a učili sa z vlastných chýb, keby sme nejakej úlohe prideliť príliš veľa alebo príliš málo hodín.

Všetky vlastnosti spomenuté v manažmente plánovania sa snažíme efektívne používať, aby sme mohli lepšie si stanoviť do budúcnosti čoho sme schopný dosiahnuť v obmedzenom časovom rozpätí.

4 Manažment kvality

4.1 Úvod

Z hľadiska kvality sme sa rozhodli, že budeme sledovať kvalitu kódu, testovať funkcionality jednotlivých kódov a overovať správnosť ich riešení (code review). Keďže sa nám však ešte nepodarilo úplne spojazdniť smerovač s SDN kontrolórom, zatiaľ žiadny kód nemáme, takže code review plánujeme v blízkej budúcnosti. Pravdepodobne bude kód písaný v programovacom jazyku Python, ak použijeme softvérový kontrolór RYU. Môže sa však stať aj to, že nakoniec sa rozhodneme pre iný kontrolór a budeme používať programovacie jazyky C a C++, prípadne Java.

4.2 Code review

Keďže sa budú programovať rôzne úlohy pre SDN kontrolór, code review sa bude robiť pre každú úlohu implementovanú v konkrétnom programovacom jazyku. Jednotliví členovia tímu teda budú kontrolovať zdrojové kódy napísané ostatnými členmi tímu, pričom budú môcť komentovať zdrojové kódy, prípadne navrhnúť vylepšenia. Cieľom bude dosiahnutie čo najvyššej kvality zdrojového kódu. Nástroj pre code review bude zvolený po definitívnom rozhodnutí, ktorý SDN kontrolór bude použitý. Potom sa budú analyzovať nástroje vhodné pre code review a na základe analýzy sa vyberie konkrétny nástroj, prípadne kombinácia viacerých nástrojov.

5 Manažment rizík

5.1 Úvod

V manažmente rizík sa snažíme analyzovať všetky možné problémové body počas práce na projekte, aby sme mali pripravené alternatívne plány, ak by nastali nejaké ťažkosti. Výhoda je potom rýchla reakcia v takýchto situáciách s použitím „plánu B“.

5.2 Tabuľka rizík

ID	Stupeň rizika	Názov	Stav	Popis	Dopady	Ošetrovanie	Krízový scenár	Riešenie
R01	3,6	Problém s naštudovaním potrebných materialov	Uzatvorené	Z dôvodu problému k prístupu na server diplomových prác nie je možné študovať efektívne	Študovanie sa výrazne predĺži	Vyžiadanie diplomových prác od vedúceho práce		Nastal daný scenár a diplomové práce boli úspešne prevzaté od vedúceho práce
R02	1,2	Vybranie nesprávnej aplikácie pre organizáciu	Uzatvorené	Existuje mnoho aplikácií na organizáciu projektu agilnou metódou, je možné že si vyberieme nevhodnú alebo aplikáciu ktorá nebude úplne splňať naše požiadavky	Spomalí organizáciu a zapríčiní zmätok v organizácii	Vykonať dôkladnú analýzu týchto webových aplikácií a ich možností		Vybratie inej webovej aplikácie
R03	2,4	Inštalácia mininetu	Uzatvorené	Riziko komplikácií pri inštalácii a nekompatibility mininetu				Nájdenie iného virtuálneho networku
R04	2	Nekompatibilita hardvéru	Uzatvorené	Nekompatibilita routra				

R05	2,45	Problém so spozajdnením softvérového kontrolóra	Uzatvorené	ASUS RT - N16 Vybratie nesprávneho softvérového kontrolóra. Existuje mnoho softvérových kontrolórov a nie všetky podporujú OpenFlow 1.3 niektoré su moc komplikované alebo obsahujú slabú dokumentáciu.	Spôsobí predĺženie projektu	Vykonať dôkladnú analýzu softvérových kontrolórov a vybratie správneho	Vyberieme iný vhodnejší softvérový kontrolór	NOX podporuje iba openflow 1.0 = nesplňa kritéria POX príliš nový a nemá takú komunitu= nesplňa kritéria OpenDaylight zlé skúsenosti mal s tým jeden bakalár = nesplňa kritéria Floodvisor používa sa iba na špeciálne účely= nesplňa kritéria OpenContrail slabá dokumentácia = nesplňa kritéria Floodlight ťažký na naučenie = nesplňa kritéria Beacon lebo súvisí s Floodlightom = nesplňa kritéria RYU sme vybrali lebo má veľkú komunitu, super dokumentáciu a podporuje openflow 1.3 stabilný a 1.4 cutting edge
-----	------	---	------------	--	-----------------------------	--	--	---

R0 6	6,3	Spojzdenie DD-WRT firmwaru pre router	Uzatvorené	Je potrebné rozchodiť firmware podporujúci Openflow 1.3 môžu nastať problémy pri inštalácii alebo kompatibilitate	Strata času hľadáním ďalšieho firmwaru	Vybratie iného firmware, ktorý bude fungovať správne	V prípade ak nebude fungovať DD-WRT a budú s ním problémy skúsime OpenWRT podporujúci OpenFlow 1.3 a ak nebude fungovať správne ani ten tak použijeme OpenWRT s OpenFlow 1.0	Vybrali sme OpenWRT s OpenFlow 1.3
R0 7	2	Problém pri spojzdení virtuálneho stroja	Uzatvorené	Môžu nastať komplikácie v nastavení alebo inštalácii Linuxu, Apache serveru, SMTP serveru	Strata času hľadáním riešenia vzniknutých chýb pri inštalácii alebo pri hľadaní riešenia potrebných nastavení	Hľadanie riešení na fórach alebo v dokumentácii Linuxu, Apache, SMTP		

Tab.č.5 – Zoznam projektových rizík

6 Manažment softvéru

6.1 Úvod

Keďže náš projekt je hlavne sieťového charakteru, tak pracujeme so softvérom potrebným na riadenie rôznych wifi routroch, kontrolórov a iných komponentov v sieti. Používame teda hlavne firmvéry a softvérové kontrolóry na funkciu a manažment siete. Okrem toho budeme tvoriť programy/skripty pre náš SDN kontrolór, ktorý bude ovládať našu sieť. Dôležitý aspekt našej SDN siete je inštalácia a spojzadenie OpenFlow prepínača, ktorá je popísaná v tejto kapitole.

6.2 OpenFlow

Táto časť sa venuje problematike smerovačov a OpenFlow. Pre potreby tímového projektu je nutné, aby naše zariadenie podporovalo centralizované ovládanie a dokázalo fungovať ako SDN (softvérovo riadený) prepínač. Bežne takúto funkcionality nemá žiadny smerovač, ale po analýze sa nám podarilo zistiť, že je možnosť prerobiť takmer hocijaký smerovač na prepínač s podporou OpenFlow a centralizovaným prístupom. Ako základ slúži niektorý z dvojice open-source firmvérou OpenWrt alebo DD-Wrt. Spočiatku sme sa uberali cestou DD-Wrt, ale po množstve komplikácií a nevyriešených problémov, čiže v dôsledku neúspešnej do implementácie sme sa rozhodli zmeniť naše orientovanie na OpenWrt. Týmto dokumentom ilustrujeme ako zmeniť bežný komerčný smerovač prepálený s OpenWrt firmvérom na OpenFlow prepínač.

OpenWrt je opisovaný ako distribúcia Linux pre vnorené systémy. Namiesto snahy vytvoriť jeden, statický firmvér, OpenWrt poskytuje plne zapisovateľný systém súborov s manažmentom modulov a balíkov. To oslobodzuje od výberu aplikácií a konfigurácií poskytnutých výrobcom a umožňuje upraviť zariadenie prostredníctvom použitia akýchkoľvek balíkov pre danú aplikáciu. Pre vývojára, OpenWrt je vývojové prostredie na vytvorenie aplikácie bez potreby vytvoriť celý nový firmvér; pre používateľa to znamená schopnosť plnej úpravy zariadenia v spôsoboch, o ktorých nikto nevedel.

Počas práce bola využitá zatiaľ najnovšia stabilná verzia OpenWrt, ktorá je 15.05 a je známa pod názvom *Chaos Calmer* a je dostupná z webovej stránky OpenWrt. Ako kompatibilný hardvér pre tento firmvér sme mali k dispozícii SOHO smerovač *Asus RT-N16*. Hardvérové parametre tohto smerovača sú:

- CPU: Broadcom BCM4718 SoC 480 MHz (architektúra MIPS 74K)

- Pamäť RAM: 128 MB
- Vnútorná pamäť Flash: 32 MB
- Rozhrania: 4+1 portový gigabitový prepínač, bezdrôtové (WiFi) rozhranie 802.11 b/g/n s max. rýchlosťou 300MB/s, 2x USB 2.0 rozhranie, sériový výstup, JTag
- Napájanie: 12V 1,25A (externý zdroj)

Možnosti doimplementovania OpenFlow do OpenWrt sú hneď dve. Prvá je vziať čistý firmvér OpenWrt vo forme zdrojových kódov, stiahnuť implementáciu čisto protokolu OpenFlow pre OpenWrt, pridať súbory do zdrojových adresárov OpenWrt a potom skompilovať nový firmvér. Druhou voľbou je pridať softvérový Open vSwitch ako aplikáciu do implementácie OpenWrt. V tomto dokumente sú popísané návody pre obe vyššie spomenuté možnosti.

1. Riešenie pomocou OpenFlow implementácie

V tejto kapitole sa venujeme doimplementovaniu OpenFlow protokolu do zdrojových kódov. Spočiatku sme sa rozhodli doimplementovať do OpenWrt verziu OpenFlow 1.0. K tejto verzii bola dobrá dokumentácia s názvom projektu Pantou vytvorená univerzitou v americkom Stanfords. Po úspešnom doimplementovaní sme ale zistili, že verzia OpenFlow 1.0 nie je dostačujúca pre potreby nášho tímového projektu a tak sme museli postúpiť na verziu 1.3. Tú sa nám podarilo nájsť samotný modul OpenFlow 1.3 pre OpenWrt bol vytvorený brazílskou nadáciou pre telekomunikácie CPqD. Openflow je implementovaný ako aplikácia na vrchu OpenWrt. Tento návod je rozdelený na tri časti:

- získať prislúchajúci firmvér pre smerovač
- vložiť firmvér do zariadenia
- pridanie OpenFlow rozšírenia
- konfigurácia

Pre tento proces je potrebné splňať nasledovné požiadavky:

- Operačný systém: Linux distribúcia (otestované s Ubuntu 14.04)
- Internetové pripojenie
- Voľné miesto na disku minimálne 10GB
- Minimálne 1GB dostupnej pamäti RAM

1.1 Získanie firmvéru

Rozhodli sme sa vytvoriť firmvér zo zdrojových kódov.

Poznámka: V nasledujúcich krokoch považujeme za pracovný adresár ~/openwrt.

Inštalácia závislostí potrebných pre OpenWrt.

```
apt-get install build-essential binutils flex bison autoconf gettext git \
sharutils subversion libncurses5-dev ncurses-term zlib1g-dev gawk libssl-dev
```

Pre potreby OpenWrt potrebujeme aj program Texinfo, ale v staršej verzii ako je šírený dnes, takže ten musíme nainštalovať ručne.

```
wget http://ftp.gnu.org/gnu/texinfo/texinfo-4.13.tar.gz
gzip -dc < texinfo-4.13.tar.gz | tar -xf -
cd texinfo-4.13
./configure
make
make install
cd ..
```

Stiahneme a pripravíme si zdrojové súbory Chaos Calmer Openwrt.

```
cd ~/openwrt
git clone git://git.openwrt.org/15.05/openwrt.git
cd openwrt
./scripts/feeds update -a
./scripts/feeds install -a
```

Vytvoríme konfiguračný súbor.

```
make menuconfig
```

Tu je dôležité nastaviť target system = Broadcom BCM47xx (MIPS), subtarget = MIPS 74K.

Potom môžeme zatlačiť ESC a potvrdiť uloženie Y.

Skontrolujeme, či máme všetko potrebné pre firmvér.

```
make prereq
```

A napokon spustíme build Chaos Calmer. Prepínač -j2 slúži na využitie viac jadier.

```
make -j2
```

Nahrание firmvéru do zariadenia

Teraz pre overenie je potrebné nahráť firmvér do smerovača. Všetky skompilované firmvéry sú pod zložkou ~/openwrt/bin/brcm. Ten náš sa volá openwrt-brcm47xx-mips74k-asus-rt-n16-squashfs.trx.

Overenie sa robí spôsobom:

- Pripojíme kábel do hociktorého "LAN" portu smerovača, nie "WAN"
- PC zmeníme IP adresu 192.168.1.10, maska 255.255.255.0
- Smerovač dáme do recovery režimu - stlačené tlačidlo reset pokým zapájam zdroj
- Recovery režim spoznáme neustálym blikaním symbolu "power" na smerovači
- Akýmkoľvek TFTP klientom pošleme firmvér na IP adresu 192.168.1.1
- Počkáme dve minúty, potom power resetneme smerovač
- Znova počkáme 2 minúty a vyskúšame telnet na 192.168.1.1
- Mala by nás uvítať OpenWrt úvodná obrazovka.

Pridanie OpenFlow rozšírenia

Presunieme sa do pracovného adresára a stiahneme OpenFlow rozšírenie.

```
cd ~/openwrt/  
git clone https://github.com/CPqD/openflow-openwrt.git
```

Pridáme symbolickú linku na OpenFlow.

```
cd ~/openwrt/package/
```

```
ln -s ~/openwrt/openflow-openwrt/openflow-1.3/
```

Pridáme základné konfiguračné súbory.

```
cd ~/openwrt/  
ln -s ~/openwrt/openflow-openwrt/openflow-1.3/files
```

Znova vytvoríme konfiguračný súbor, tento krát už s OpenFlow.

```
make menuconfig
```

Zvolíme nasledovné:

- Target system = Broadcom BCM47xx (MIPS)
 - Subtarget = MIPS 74k
 - Pod network zvolíme balík tc, aby sa nainštaloval
 - Pod Kernel Modules -> Network Support zvolíme kmod-tun na inštaláciu
- ukončíme a uložíme

Pridáme podporu pre frontu.

```
make kernel_menuconfig
```

Pod Networking Support -> Networking options -> QoS zvolíme Hierarchical Token Bucket (HTB) na inštaláciu. Ukončíme a uložíme.

Spustíme build.

```
make
```

Nainštalujeme firmvér do smerovača, napríklad vyššie spomenutým spôsobom cez recovery.

Overenie

Základne, firmvér vytvorený zo zdrojových súborov bude mať port označený ako "internet" (WAN) nastavený ako manažovací port, so statickou IP 192.168.1.1. Mali by sme byť schopný pripojiť sa cez tento port ak máme IP adresu PC v podsieti 192.168.1.0/24. Keď sme nakonfigurovali PC, môžeme sa pokúsiť pripojiť.

```
telnet 192.168.1.1
```

Po pripojení overím, či bežia potrebné OpenFlow procesy.

```
ps aux | grep ofprotocol  
ps aux | grep ofdatapath
```

1.2 Konfigurácia

Pre OpenFlow sa nachádzajú v smerovači tri potrebné konfiguračné súbory. Pre sieť (/etc/config/network) a wifi (/etc/config/wireless) a konfiguráciu (/etc/config/openflow).

Primárne je wifi vypnuté. Tento smerovač vie byť použitý ako 5 portový prepínač. Najskôr musíme nastaviť /etc/config/network.

```
config 'switch'  
    option 'name' 'eth0'  
    option 'reset' '1'  
    option 'enable_vlan' '1'  
  
config 'switch_vlan'  
    option 'device' 'eth0'  
    option 'vlan' '1'  
    option 'ports' '4 8t'  
  
config 'switch_vlan'  
    option 'device' 'eth0'  
    option 'vlan' '2'  
    option 'ports' '3 8t'  
  
config 'switch_vlan'  
    option 'device' 'eth0'
```

```

        option 'vlan' '3'
        option 'ports' '2 8t'

config 'switch_vlan'
    option 'device' 'eth0'
    option 'vlan' '4'
    option 'ports' '1 8t'

config 'switch_vlan'
    option 'device' 'eth0'
    option 'vlan' '0'
    option 'ports' '0 8t'

config 'interface' 'loopback'
    option 'ifname' 'lo'
    option 'proto' 'static'
    option 'ipaddr' '127.0.0.1'
    option 'netmask' '255.0.0.0'

config 'interface'
    option 'ifname' 'eth0.1'
    option 'proto' 'static'

config 'interface'
    option 'ifname' 'eth0.2'
    option 'proto' 'static'

config 'interface'
    option 'ifname' 'eth0.3'
    option 'proto' 'static'

config 'interface'
    option 'ifname' 'eth0.4'
    option 'proto' 'static'

config 'interface'
    option 'ifname' 'eth0.0'
    option 'proto' 'static'
    option type 'bridge'
    option 'ipaddr' '192.168.1.1'
    option 'netmask' '255.255.255.0'

```

Manažovací port je WAN. Tu bude pripojený aj kontrolór. Zvyšné porty LAN1-4 sú použiteľné pre prepínač.

Ešte potrebujeme nastaviť wifi, ktoré je stále vypnuté. Základne je povolené maximálne 802-11g, ale smerovač je schopný fungovať aj v štandarde n. Pre využitie sú tieto príkazy.

```
opkg remove kmod-b43 kmod-b43legacy
opkg update
opkg install kmod-brcm-smac
rm -f /etc/config/wireless
wifi detect > /etc/config/wireless
wifi
```

Nastavíme konfiguračný súbor pre wifi `/etc/config/wireless`.

```
config wifi-device wlan0
    option type mac80211
    option channel 5
    option macaddr 00:25:9c:30:2c:f4
    option hwmode 11n

    # REMOVE THIS LINE TO ENABLE WIFI:
    # option disabled 0

config wifi-iface
    option device wlan0
#    option network lan
    option mode ap
    option ssid OpenFlow-OpenWrt
    option encryption none
```

Ďalej potrebujeme nastaviť `/etc/config/openflow`.

```
config 'ofswitch'
    option 'dp' 'dp0'
    option 'ofports' 'eth0.0 eth0.1 eth0.2 eth0.3 eth0.4 wlan0 '
    option 'ofctl' 'tcp:192.168.1.10:6633' #ip adresa controllera
    option 'mode' 'outofband'
```

Nakoniec spravíme reštart, nech sa všetky zmeny prejaví. Ale pre istotu aj fyzický reštart.

```
/etc/init.d/openflow restart
/etc/init.d/network restart
```

2. Riešenie pomocou Open vSwitch

Open vSwitch je softvérový viacvrstvový prepínač licencovaný pod Open Source Apache 2.0 licencií. Je navrhnutý tak, aby umožňoval masívnu automatizáciu sietí pomocou programových rozšírení. Väčšina zdrojového kódu je napísaná v natívnom jazyku C a je jednoducho prenositeľný do rôznych prostredí, kam patria predovšetkým aj vnorené systémy.

Aktuálna verzia Open vSwitch (v2.4.0) podporuje nasledovné vlastnosti:

- Monitorovanie komunikácie medzi virtuálnych systémov (inter-VM) cez protokolov NetFlow, sFlow(R), IPFIX, SPAN, RSPAN a GRE tunelov.
- LACP (IEEE 802.1AX-2008)
- Štandard 802.1Q – podpora VLAN pomocou trunk liniek
- Multicast snooping
- IETF Auto-Attach SPBM a podpora LLDP
- Štandard 802.1ag pre správu a údržbu sietí
- STP (IEEE 802.1D-1998) a RSTP (IEEE 802.1D-2004)
- Konfigurácia QoS a riadenie premávky
- Podpora pre HFSC qdisc
- Riadenie premávky medzi VM rozhraniami
- NIC bonding with source-MAC load balancing, active backup, and L4 hashing
- Podpora protokolu OpenFlow (s mnohými rozšíreniami pre virtualizáciu)

- Podpora IPv6
- Tunelovacie protokoly (GRE, VXLAN, STT, a Geneve, s IPsec podporou)
- Protokol na vzdialenú konfiguráciu pomocou C a Python väzieb
- Prepínanie (forwarding) v rámci jadra (kernel) a používateľského priestoru (user-space)
- Abstraktná vrstva prepínania (forwarding) umožňuje jednoduchú prenositeľnosť do nových softvérových a hardvérových platforiem

Hlavnou výhodou *Open vSwitch* je to, že podporuje naraz niekoľko verzií protokolu OpenFlow a je kompatibilný s firmvérom *OpenWrt* pre SOHO smerovačov. Súčasná podpora jednotlivých verzií OpenFlow vyzerá nasledovne:

Verzie <i>Open vSwitch</i>	Verzie <i>OpenFlow</i>					
	1.0	1.1	1.2	1.3	1.4	1.5
1.9 a staršie	áno	nie	nie	nie	nie	nie
1.10	áno	nie	čiastočne	čiastočne	nie	nie
1.11	áno	nie	čiastočne	čiastočne	nie	nie
2.0	áno	čiastočne	čiastočne	čiastočne	nie	nie
2.1	áno	čiastočne	čiastočne	čiastočne	nie	nie
2.2	áno	čiastočne	čiastočne	čiastočne	čiastočne	čiastočne
2.3	áno	áno	áno	áno	čiastočne	čiastočne
2.4	áno	áno	áno	áno	čiastočne	čiastočne

Tab.č.6 – Stav podpory jednotlivých verzií *OpenFlow*

Posledné verzie už majú plnú podporu pre OpenFlow v1.3, ktorá je pre náš projekt postačujúca. Preto vznikol nápad využiť *Open vSwitch* v prepínačoch SDN v rámci riešenia projektu.

2.1 Pridanie modulu do firmvéru OpenWrt

Na pridanie balíka Open vSwitch do firmvéru máme k dispozícii 2 možnosti:

1. Pridanie OVS do zdrojového kódu a kompilácia firmvéru

Inštalácia hotového balíka OVS pomocou manažéra balíkov na bežiacom systéme

Počas riešenia boli otestované obe metódy. Postup je podrobne opísaný v nasledujúcich častiach tejto podkapitoly.

Metóda 1: Pridanie OVS do zdrojového kódu a kompilácia

Pre tento proces je potrebné spĺňať nasledovné požiadavky:

- Operačný systém: Linux distribúcia (otestované s Ubuntu 14.04)
- Internetové pripojenie
- Voľné miesto na disku minimálne 10GB
- Minimálne 1GB dostupnej pamäti RAM

V prvom kroku je potrebné skontrolovať, či sú nainštalované všetky potrebné balíky pre kompiláciu použitím tohto príkazu:

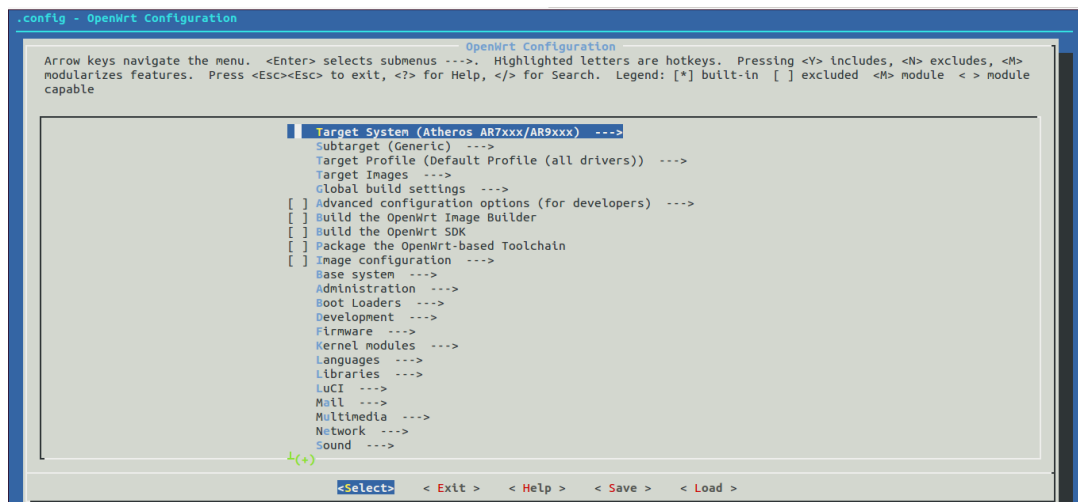
```
$ apt-get install build-essential binutils flex bison autoconf  
gettext texinfo sharutils subversion git libncurses5-dev ncurses-  
term zlib1g-dev gawk
```

Teraz máme na možnosť vybrať si verziu OpenWrt, ktorú chceme použiť. V tomto prípade to bude verzia *Chaos Calmer*. Najnovšia, ale nestabilná sa nazýva vždy ako *trunk*.

Predpokladáme, že sa nachádzame v domovskom adresári, kde vykonáme tieto príkazy:

```
$ svn co svn://svn.openwrt.org/openwrt/branches/chaos_calmer  
$ cd chaos_calmer  
$ ./scripts/feeds update -a  
$ ./scripts/feeds install -a  
$ make menuconfig
```

Po vykonaní posledného príkazu by sa malo objaviť automaticky okno s konfiguráciou, ktoré vyzerá podobne ako na obrázku:



Obr.

Obr.č.6 –Konfiguračné okno „menuconfig“

Ako prvé vyberieme si položku *Target system* pomocou <Select>. Následne si vyberieme *Broadcom BCM47xx/53xx (MIPS)* v prípade Asus RT-N16. Pri stlačení kláves je odporúčané sa riadiť pokynmi, ktoré sa vždy objavujú na obrazovke v hornej časti okna *menuconfig*.

Ako *Subtarget* si vyberieme možnosť *MIPS 74K*. (samozrejme v prípade RT-N16)

V časti *Target Profile* máme možnosť vybrať si ovládač pre WiFi. Dôležité je že predvolený ovládač *b43* podporuje maximálne režim 802.11g.

V časti *Kernel modules* a *Network support* potrebujeme ešte *kmod-tun*.

V časti *Network* potrebujeme ešte modul *openvswitch*.

V prípade potreby používateľského rozhrania, máme na možnosť vybrať si a prispôbiť balík *LuCI*, napríklad takto:

v časti *LuCI* -> *Collections* si vyberieme *luCI*.

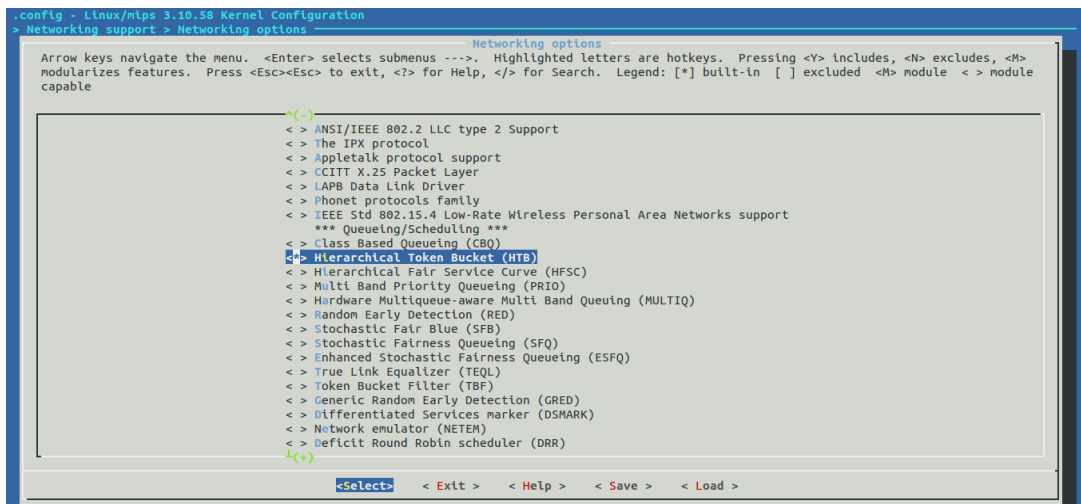
- v časti *LuCI* -> *Modules* si vyberieme *luCI-base*, *luCI-mod-admin-full*.
- v časti *LuCI* -> *Applications* si vyberieme *luCI-app-firewall*, *luCI-app-ntpc*.
- v časti *LuCI* -> *Themes* si zvolíme vzhľad webového rozhrania. Predvolený je *luCI-theme-bootstrap*.

- v časti Luci -> Protocols si vyberieme ipv6 a ppp.
- v časti Luci -> Libraries si vyberieme httpclient, ip, json a nixio.

Hotovú konfiguráciu si uložíme pomocou tlačidla <Exit> a následne Yes v dialógovom okne. Po návrate do konzoly spustíme príkaz:

```
$ make kernel_menuconfig
```

Objaví sa podobné okno ako *menuconfig* ale tento krát s inými položkami. Na ceste *Networking Support* -> *Networking Options* si pridáme balík *Hierarchical Token Bucket (HTB)* do kompilácie.



Obr.č.7 – Konfiguračné okno „Kernel menuconfig“ a položka HTB

Nasleduje posledný krok tejto metódy a to je spustenie kompilácie, ktorá sa spustí vydaním príkazu:

```
$ make
```

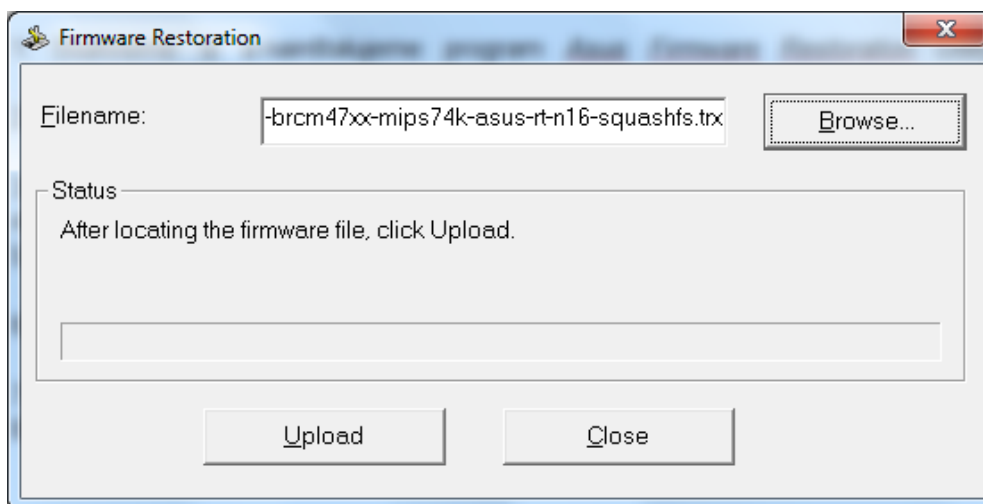
Proces kompilácie môže trvať aj niekoľko hodín. Pre zrýchlenie procesu sa dá zapnúť *multithreading* a to tak že pomocou prepínača *-j* pre *make* pridáme do procesu vykonanie viac úloh naraz. Napríklad v prípade *-j2* sa budú vykonávať dve úlohy naraz, čo je odporúčané množstvo v prípade dvojjadrového procesora a 1GB RAM.

Hotový binárny súbor sa bude nachádzať v adresári *./bin/<SoC_type>* . V prípade RT-N16 to bude: *openwrt-brcm47xx-mips74k-asus-rt-n16-squashfs.trx*.

Inštalácia firmvéru na router RT-N16

Predpokladáme, že máme k dispozícii binárny súbor s príponou *.trx*, ktorý je určený pre dané zariadenie. V nasledujúcich krokoch je opísaný postup inštalácie pri použití operačného systému Windows. Na OS Linux je to možné tiež, napr. pomocou nástroja *ftp*.

1. Stiahneme si a nainštalujeme program Asus Firmware Restoration Utility zo stránkach výrobcu.
2. Smerovača si prepne do tzv. Recovery režimu pomocou tlačidla RESET. Po úspešnom prepnutí do tohto režimu sa bude LED PWR neprerušene blikať.
3. IP adresu počítača si nastavíme na 192.168.1.10 a masku na 255.255.255.0. Pripojíme si zariadenie k počítaču použitím niektorého portu LAN. (napr. LAN1)
4. Spustíme program Firmware Restoration a pomocou Browse si vyberieme súbor s firmvérom (prípona *.trx*).
Klikneme na *Upload* a čakáme kým sa neobjaví informácia o úspešnom dokončení procesu inštalácie. (obrázok nižšie)
5. Zariadenie sa po inštalácii firmvéru reštartuje.
6. O úspešnosti inštalácie sa presvedčíme pomocou nástroja telnet. Pripojíme sa na adresu 192.168.1.1. Mala by sa objaviť konzola s nadpisom OpenWrt.



Obr.č.8 –Okno aplikácie *Firmware Restoration*

Metóda 2: Inštalácia hotového balíka OVS pomocou manažéra balíkov

Táto metóda je jednoduchšia a rýchlejšia ako predchádzajúca, keďže je to bez dlhotrvajúcej kompilácie firmvéru. Nepotrebujeme tu ani prostredie pre kompiláciu. Nevýhodou však je že inštalácia na zariadení trvá dlhšie, keďže potrebné balíky sú nainštalované manuálne pomocou

manažéra balíkov *opkg*. Taktiež je nevyhnutná dostupnosť internetového pripojenia na smerovači cez WAN port.

V prvom kroku si stiahneme už kompilovaný firmvér zo stránky OpenWrt pre príslušné zariadenie. V našom prípade je to verzia *Chaos Calmer (15.05)* pre router *Asus RT-N16*. Názov súboru s firmvérom vyzerá nasledovne:

```
openwrt-15.05-brcm47xx-mips74k-asus-rt-n16-squashfs.trx
```

V ďalšom kroku si nainštalujeme tento stiahnutý firmvér na koncové zariadenie podľa návodu vyššie *Inštalácia firmvéru na router RT-N16* (v prípade Asus). Po úspešnej inštalácii by sme mali mať k dispozícii prístup do zariadenia cez *telnet*.

Pomocou príkazu *passwd* je potrebné nastaviť si heslo. Po úspešnom nastavení hesla budeme mať k dispozícii aj prístup cez SSH a cez webového rozhrania *Luci*. Používateľské meno bude vždy „root“.

Potrebné balíky si nainštalujeme do vnútornej pamäti zariadenia pomocou nasledovných príkazov:

```
$ opkg update # aktualizácia databázy o dostupných balíkoch
$ opkg --force-depends install kmod-tun openvswitch
```

V prípade potreby používateľského rozhrania, máme tu možnosť nainštalovať si a prispôbiť balík *Luci*. Minimálna konfigurácia sa inštaluje takto:

```
$ opkg --force-depends install luci
```

Inštaláciu dokončíme reštartovaním zariadenia a pokračujeme s konfiguráciou SDN prepínača.

2.2 Konfigurácia prepínača

Predpokladáme, že už máme pripravené zariadenie s firmvérom *OpenWrt*, ktorý obsahuje funkčné používateľské prostredie *Open vSwitch*. Musia bežať procesy *ovsdb-server* a *ovs-vswitchd*.

Nasledovné kroky konfigurácie a obsahy konfiguračných súborov sú kompatibilné predovšetkým s verziou OpenWrt 15.05, Open vSwitch 2.3.9 a zariadením Asus RT-N16. V prípade iných

softvérových verzií alebo iného hardvéru obsah niektorých konfiguračných súborov môže vyzerat' inak. Postup:

1. Pripojíme sa na IP adresu prepínača (predvolene 192.168.1.1) cez protokol SSH a zadáme používateľské meno (root) a heslo.
2. Pomocou obľúbeného textového editora (napríklad vi alebo nano) zmeníme obsah niektorých konfiguračných súborov.
 - a. Súbor /etc/config/network :

```
config switch
    option name 'eth0'
    option reset '1'
    option enable_vlan '1'

config switch_vlan
    option device 'eth0'
    option vlan '0'
    option ports '1 8t'

config switch_vlan
    option device 'eth0'
    option vlan '1'
    option ports '4 8t'

config switch_vlan
    option device 'eth0'
    option vlan '2'
    option ports '3 8t'

config switch_vlan
    option device 'eth0'
    option vlan '3'
    option ports '2 8t'

config interface 'loopback'
    option ifname 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd1a:8ff4:8d69::/48'

config interface 'lan'
```

```

option ifname 'eth0.0'
option force_link '1'
option type 'bridge'
option proto 'static'
option ipaddr '192.168.1.1'    # zmenit podla potreby
option netmask '255.255.255.0'
option ip6assign '60'

config interface
    option ifname 'eth0.1'
    option proto 'static'

config interface
    option ifname 'eth0.2'
    option proto 'static'

config interface
    option ifname 'eth0.3'
    option proto 'static'

```

Kontrolované porty prepínača budú LAN1-3 a port pre pripojenie kontrolóra bude LAN4. Port WAN je vypnutý.

b. Súbor /etc/config/wireless :

Odstránime riadky „*option disabled 1*“, „*option network lan*“ a pridáme alebo zmeníme tieto riadky v časti *config wifi-iface*:

```

option ssid 'inWifi'
option encryption 'psk2'    ← podľa potreby
option key 'my_password'    ← podľa potreby, je to heslo typu WPA2-PSK

```

c. Pre správnu funkčnosť WLAN LED je potrebné rozšíriť obsah súboru /etc/config/system pridaním nasledovných riadkov:

```

config led wlan_led
    option name 'WLAN'
    option sysfs 'bcm47xx:blue:wlan'
    option trigger 'netdev'
    option dev 'wlan0'

```

```
option mode 'link tx rx'
```

3. Po dokončení zmien v konfiguračných súboroch je potrebné reštartovať prepínač.
4. Počítač kde robíme konfiguráciu už musí byť pripojený výhradne do portu LAN4.
5. Spojíme sa so zariadením cez SSH, rovnako ako v predchádzajúcich krokoch.
6. Vytvoríme si most (bridge) pomocou príkazu:

```
$ ovs-vsctl add-br br0
```

Parameter *br0* je názov vytvoreného rozhrania.

7. Do vytvoreného virtuálneho rozhrania *br0* pridáme rozhrania portov. Predpokladáme, že sme využili obsah súboru *network* opísaného vyššie.

```
$ ovs-vsctl add-port br0 eth0.1
$ ovs-vsctl add-port br0 eth0.2
$ ovs-vsctl add-port br0 eth0.3
$ ovs-vsctl add-port br0 wlan0 # ak chceme pridať do switchu aj WiFi
```

8. Nastavíme si IP adresu a port kontrolóra. V našom prípade to bude 192.168.1.10:6633.

```
$ ovs-vsctl set-controller br0 tcp:192.168.1.10:6633
```

9. Nastavíme si požadovanú/é verziu/e OpenFlow. Prvým príkazom sa povolí iba OpenFlow 1.3, druhým sa povolia verzie 1.0 a 1.3.

```
$ ovs-vsctl set bridge br0 protocols=OpenFlow13
$ ovs-vsctl set bridge br0 protocols=OpenFlow10,OpenFlow13
```

10. Konfiguráciu si overíme pomocou príkazu:

```
$ ovs-vsctl show
```


7 Manažment testovania

7.1 Úvod

Prudký rozvoj informačných technológií v súčasnosti prináša čoraz vyššie nároky na funkčnosť softvérových aplikácií, čím narastajú aj požiadavky na ich kvalitu. Preto musí mať manažment testovania svoje pevné miesto v procese vývoja akéhokoľvek riešenia. Testovanie je systematický proces - pozorovanie správania sa systému v špecifických podmienkach simulujúcich reálne prostredie - zameraný na odhalenie chýb, nedostatkov a odchýlok od požiadaviek zákazníka a tiež na overenie správania sa v hraničných situáciách z pohľadu vstupných dát, záťaže alebo bezpečnosti. Jednotlivé zistenia sa podrobne zaznamenávajú a vyhodnocujú. Cieľom testovania je vyhľadávanie chýb v čo najkratšom čase, na čo najnižšej úrovni vývoja riešenia a zaistenie ich nápravy.

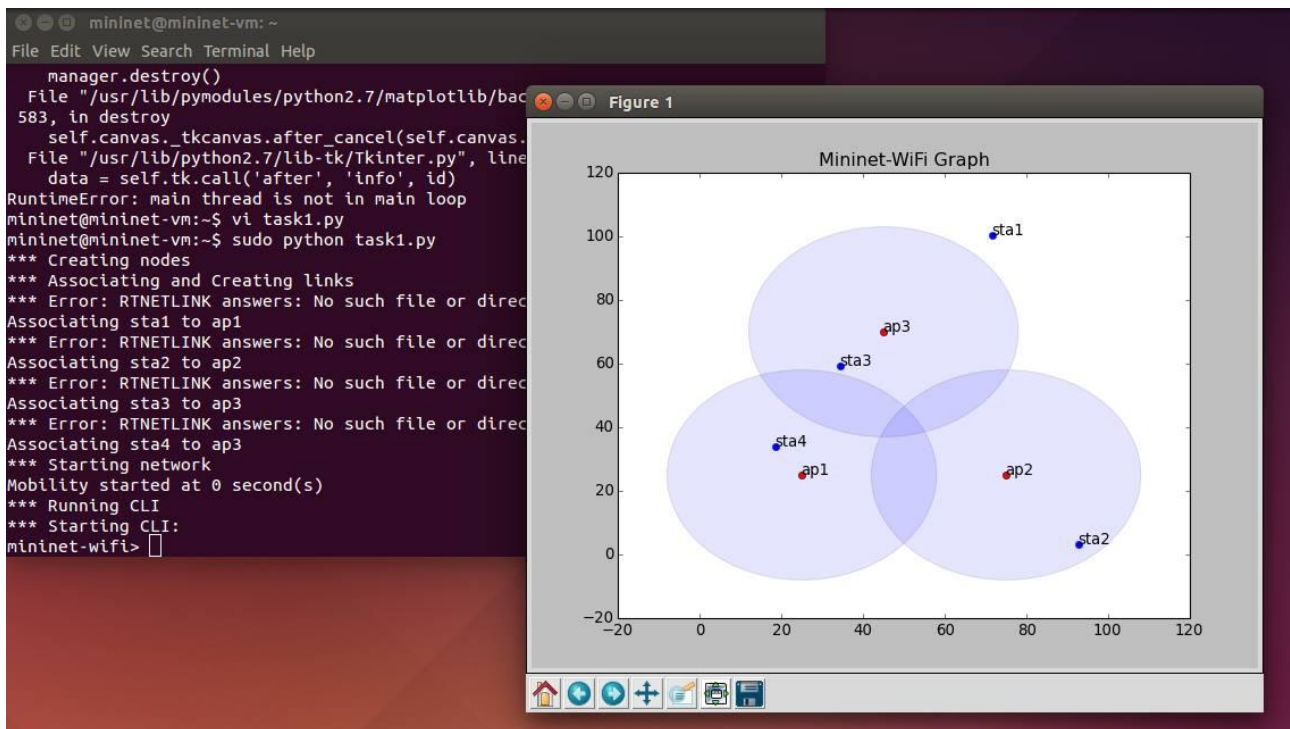
Účelom tohto manažmentu je určenie procesov, nástrojov a metód prebiehajúcich počas celej fázy testovania. K metódam definuje konkrétne nástroje ako aj výstupy, ktoré musia byť výsledkom daného procesu. Táto dokumentácia je určená pre sekciu testovania. Najskôr sa venuje základnej charakteristike nástrojov na testovanie ako takých a potom v jednotlivých kapitolách rozoberá testovacie nástroje podrobnejšie.

7.2 Použité nástroje

Štýl nášho projektu si vyžaduje špeciálne testovanie SDN sieťovej architektúry. Účinnosť nášho softvéru, ktorý budeme vytvárať pre túto sieť otestujeme dvoma spôsobmi: virtuálne a fyzicky. Pre virtuálne testovanie používame virtuálny systém *Mininet* a na fyzické testovanie používame našu normálnu architektúru s použitím softvéru ako *Wireshark*, *iw* a *Ixia IxChariot* na sledovanie prenosu toku dát medzi koncovým zariadeným a jednotlivými prístupovými bodmi.

7.3 Testovanie v mininete

Mininet ako už bolo spomenuté je virtuálny stroj. Má perfektné vlastnosti pre testovanie SDN sietí čo sa týka virtuálnej simulácie. V mininete sa dajú vytvárať rôzne virtuálne SDN architektúry na ktorých sa veľmi rýchlo dajú otestovať viaceré softvérové kontrolóry. Existuje aj viacero pluginov, ktoré rozširujú mininet. Jedným z nich, ktorý používame je plugin na simulovanie wifi signálového pokrytia.



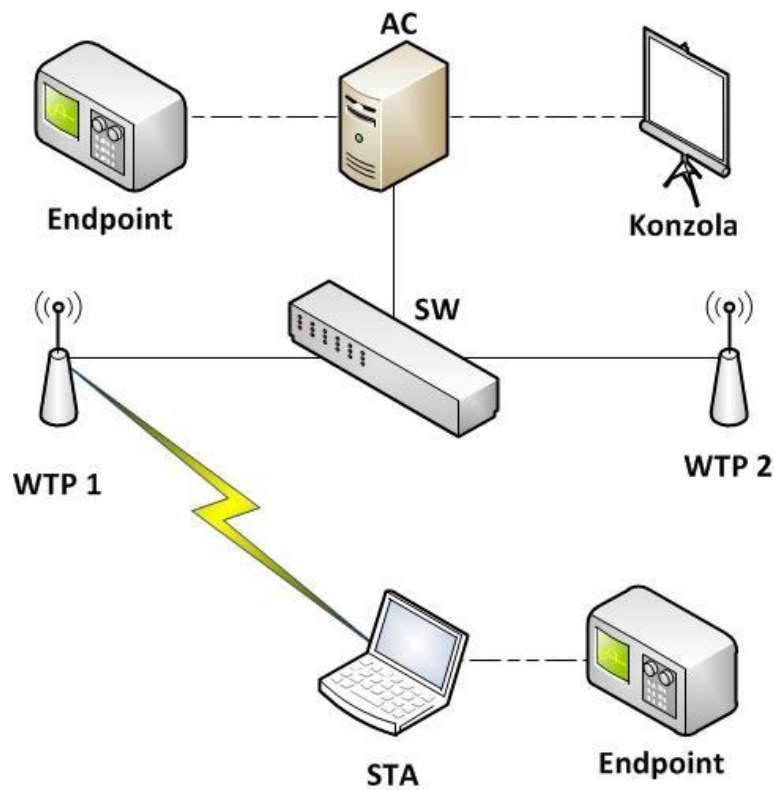
Obr.č.9 – Ukážka Mininet grafu s wifi pokrytím

Simulácia simuluje viacerých používateľov ako chodia po oblasti s troma prístupovými bodmi. Na mininete môžeme sledovať, kto sa ako pripája a odpája pomocou vstavaného *Wireshark* programu. Okrem iného môžeme skúšať ako fungujú flow tabuľky a pohrať sa s ich konfiguráciou. Najčastejšie používame príkaz: `sudo mn`, ktorý nám vytvorí jednoduchú virtuálnu default SDN architektúru, na ktorej sa dá operovať ako na skutočnej. Samozrejme toto nám nestačí na plné pokrytie testovania a tak máme ešte fyzickú architektúru ako testovacie prostredie.

7.4 Testovanie na fyzickej architektúre

Pre účely testovania vlastností nášho riešenia sme použili program IXIA IxChariot. Program IxChariot od spoločnosti IXIA predstavuje sofistikovanejšie riešenie testovania počítačových sietí. Je to profesionálny nástroj pre simuláciu a testovanie výkonu sieťovej komunikácie v reálnych podmienkach zaťaženia. Popri množstve funkcií, ktoré poskytuje, sme použili niektoré scenáre testovania, ktoré sú preddefinované, na vygenerovanie prevádzky s rôznymi vlastnosťami, kde je možné zmeniť dôležité nastavenia podľa používateľa. Ďalej je umožnené vytvoriť výstupy zo zozbieraných dát v podobe grafov a tabuliek a takisto spraviť export do formátu HTML. Program IxChariot pozostáva z dvoch komponentov, a to tzv. konzola programu a koncové body merania (*angl. endpoint*). V konzole sa definujú dvojice koncových bodov merania a priradujú sa a konfigurujú scenáre merania pre tieto dvojice (okrem dvojíc sa dajú vytvárať aj skupiny ak

potrebujeme vykonať merania multikastovej prevádzky). Medzi ďalšie funkcie konzoly programu môžeme zaradiť aj spúšťanie, či zastavovanie testov a tiež zobrazovanie nameraných údajov. Všetky programom poskytované možnosti sú dostupné prostredníctvom GUI. V dokumente inžinierske dielo v kapitole architektúra sa nachádza základný návrh našej architektúry a z toho vyplýva aj základná topológia, ktorá je na obrázku Obr.č 7. V topológii možno vidieť aj spomenuté a použité komponenty programu IxChariot.



Obr.č.10 – Ukážka topológie testovania

(zdroj: JANČIGA, T. Centralizované riadenie prístupových bodov pre neviditeľný prechod klientov v sieťach štandardu IEEE 802.11. *Diplomová práca*. Bratislava, 2013.)

8 Manažment dokumentácie

8.1 Úvod a popis

Manažment dokumentácie sa zaoberá dokumentovaním všetkých častí projektu. Všetky dokumenty sú ukladané na úložisku Google drive, ktorý je dostupný 24/7 všetkým členom tímu. Dokumentácia je rozdelená na niekoľko adresárov zameraných podľa obsahu:

Analýza - dokumenty vytvorené počas analýzy práce

Metodiky - dokumenty opisujúce postup práce

Retrospektívy šprintov - dokumenty popisujúce spokojnosť s riadením a postupmi na jednotlivých šprintoch.

Výstupné dokumenty - dokumenty odovzdávané do AIS

Vzory - šablóny pre dokumentovanie.

Zápisnice - dokumenty zo stretnutí.

Týmto rozdelením zabezpečíme prehľadnosť vo všetkých dokumentoch aj v neskorších fázach projektu. Pre vytváranie dokumentov sú vytvorené a popísané metodiky, uvedené v časti

1.5 Používané metodiky. Pri vytváraní dokumentov je majiteľom dokumentu tvorca. Tvorca má ako jediný právo editovať dokument. Ostatní členovia majú právo iba pridávať komentáre do dokumentu a diskutovať o jednotlivých častiach. V prípade potreby je užitočné vytvárať verzie jednotlivých dokumentov s určenou štruktúrou názvu: *názov_dokumentu_v_číslo_verzie*. Príklad: Riadenie projektu InWifi_v_1.0.docx.

8.2 Metodika dokumentácie

Pole pôsobnosti

Manažment dokumentácie opisuje dokumentovanie všetkých častí životného cyklu projektu. Metodika sa ďalej zaoberá rozdelením rolí a zodpovedností pre dokumentovanie a definovanie procesov.

Cieľ

Cieľom tejto metodiky, je rozdelenie zodpovedností za manažovanie dokumentov, zobrazenie procesov a určenie rolí vykonávajúcich jednotlivé procesy. Vstupné a výstupné podmienky pre dokumentovanie častí.

Role a zodpovednosti

Rola	Zodpovednosť
Manažér plánovania	Vytváranie exportov z plánovaných a vykonaných aktivít.
Manažér kvality	Vytvorenie a manažovanie dokumentu kvality produktu.
Manažér dokumentovania	Vytvorenie zápisníc, vzorov a metodík potrebných pre dokumentovanie.
Manažér rizík	Vytvorenie a manažovanie dokumentu s rizikami.
Manažér testovania	Vytvorenie testovacích scenárov, vytvorenie harmonogramu testovania.
Hlavný architekt	Vytvorenie a manažovanie architektúry..
Vývojár	Vytvorenie dokumentov s implementovanými časťami produktu.
Analytik	Vytvorenie dokumentov súvisiacich s analýzou častí potrebných pre návrh a implementáciu.
Tester	Vytvorenie výstupných dokumentov z testovania.

Proces plánovania

Krok	Názov
1.	Plánovanie
2.	Zaznamenávanie úloh
3.	Export vykonávaných aktivít

Plánovanie

Vstup: Požiadavka na naplánovanie úloh pre šprint

Výstup: Spísaný zoznam úloh s pridelenými osobami

Zodpovedný: Manažér plánovania

Proces: Manažér plánovania vedie stretnutie, zadáva úlohy a po dohode s ostatnými členmi tímu, zdefiniuje zodpovedné osoby za vykonávanie úloh.

Zaznamenávanie úloh

Vstup: Zoznam úloh spolu s pridelenými osobami

Výstup: Zaznamenané úlohy v manažovacom systéme

Zodpovedný: Manažér plánovania

Proces: Manažér plánovania po stretnutí zapíše všetky definované úlohy do manažovacieho systému Trello. K úlohe priradí osobu zodpovednú za vykonanie a nastaví dátum do ktorého je potrebné vykonať úlohu.

Export vykonaných aktivít

Vstup: Zaznamenané úlohy

Výstup: Zaznamenané úlohy v manažovacom systéme

Zodpovedný: Manažér plánovania

Proces: Manažér plánovania vykoná export údajov za daný šprint a uloží ho do priečinku Výstupné dokumenty nachádzajúci sa na google drive. Manažér plánovania zadáva názov dokumentu podľa šprintu.

Proces kvality

Krok	Názov
1.	Výber kódu na prehliadku
2.	Prehliadka kódu a zaznamenanie stavu
3.	Oprava kódu
4.	Nasadenie kódu

Výber kódu na prehliadku

Vstup: Implementovaný kód

Výstup: Výber kódu na prehliadku

Zodpovedný: Vývojár

Proces: Vývojár vytvorí kód na implementovanie. Pred nasadením je potrebné vykonať kontrolu kódu manažérom kvality a preto oboznámi vývojár manažéra.

Prehliadka kódu a zaznamenanie stavu

Vstup: Kód vybraný na prehliadku

Výstup: Skontrolovaný kód

Zodpovedný: Manažér kvality

Proces: Manažér kvality skontroluje vybraný kód. V prípade nesplnenia štandardov, alebo funkcionality manažér vráti vývojárovi kód na opravu. V prípade ak kód spĺňa všetky potrebné štandardy a funkcionality, je kód schválený a pripravený na nasadenie. V oboch prípadoch manažér kvality zaznamená výsledok prehliadky do dokumentu.

Oprava kódu

Vstup: Prehliadnutý kód spolu so zaznamenaným stavom

Výstup: Opravený kód a zaznamenanie zmien

Zodpovedný: Vývojár

Proces: Vývojár obdrží prehliadaný kód spolu s pripomienkami na opravu. Tieto pripomienky zapracuje a zaznamená zmenu. Po vykonaní zmien, znovu odovzdá na kontrolu manažérovi kvality.

Nasadenie kódu

Vstup: Prehliadnutý kód spolu so zaznamenaným stavom

Výstup: Nasadený kód a zaznamenanie zmien

Zodpovedný: Vývojár

Proces: Vývojár obdrží prehliadaný kód spolu so stavom. V prehliadke neboli nájdené žiadne nedostatky, takže vývojár nasadí kód do prostredia a zaznamená nasadenie kódu do dokumentu, v ktorom sa vykonávala prehliadka.

Proces dokumentovania stretnutí

Krok	Názov
1.	Zaznamenanie činností zo stretnutia
2.	Vytvorenie zápisníc

Zaznamenanie činností zo stretnutia

Vstup: Potreba zaznamenávania stretnutí

Výstup: Zoznam preberaných a vykonaných činností na stretnutí

Zodpovedný: Manažér dokumentovania

Proces: Manažér dokumentovania počas stretnutia zapisuje vykonávané a preberané činnosti.

Vytvorenie zápisníc

Vstup: Zaznamenané vykonané a preberané činnosti počas stretnutia

Výstup: Zápisnica zo stretnutia

Zodpovedný: Manažér dokumentovania

Proces: Manažér po stretnutí vytvorí zápisnicu zo stretnutia podľa zozbieraných činností.

Proces vytvárania metodík a vzorov

Krok	Názov
1.	Vytvorenie metodiky
2.	Vytvorenie vzoru
3.	Kontrola vytvorených dokumentov

Vytvorenie metodiky

Vstup: Potreba vytvorenia metodiky

Výstup: Vytvorená metodika

Zodpovedný: Manažér dokumentovania

Proces: Manažér dokumentovania vytvorí na základe potreby metodiku, pre dokumentovanie procesov počas celého behu projektu.

Vytvorenie vzoru

Vstup: Potreba vytvorenia vzoru

Výstup: Vytvorený vzor

Zodpovedný: Manažér dokumentovania

Proces: Manažér dokumentovania vytvorí vzor dokumentu, pre jednoznačnú, jednoduchšiu a konzistentnú prácu s dokumentom.

Kontrola vytvorených dokumentov

Vstup: Vytvorený dokument inou osobou

Výstup: Kontrola formálnej časti dokumentu

Zodpovedný: Manažér dokumentovania

Proces: Manažér dokumentovania skontroluje vytvorený dokument. Tento dokument bol vytvorený inou osobou. Kontrola sa vykonáva na overenie, či tvorca dokumentu dodržal definované štandardy. V prípade potreby manažér dokumentovania okomentuje chybové časti a oboznámi tvorca dokumentu.

Proces zaznamenávania rizík

Krok	Názov
1.	Definovanie rizík a opatrení
2.	Vytvorenie zoznamu rizík
3.	Manažovanie rizík

Definovanie rizík a opatrení

Vstup: Potreba zaznamenávania rizík

Výstup: Definované riziká a opatrenia

Zodpovedný: Manažér rizík

Proces: Manažér rizík zozbiera všetky potrebné podklady pre analyzovanie projektu. Určí riziká vyskytujúce sa na projekte a definuje opatrenia na riziká.

Vytvorenie zoznamu rizík

Vstup: Definované riziká a opatrenia

Výstup: Zoznam rizík

Zodpovedný: Manažér rizík

Proces: Manažér rizík na základe definovaných rizík a opatrení určí dopad, percentuálne hodnotenie a zaznamená všetky atribúty do vytvoreného dokumentu zoznamu rizík.

Manažovanie rizík

Vstup: Zmena stavu rizika

Výstup: zaznamenaná zmena v zozname rizík

Zodpovedný: Manažér rizík

Proces: Manažér rizík udržiava zoznam rizík vždy aktuálny. Na základe zmeny stavu rizika je potrebné zaznamenať túto zmenu aj do zoznamu rizík spolu s dátumom. Pri akejkoľvek zmene, prípadne doplnení je potrebné zaznamenávať aj všetky súvisiace atribúty. Výstupom je aktuálny zoznam rizík.

Proces testovania

Krok	Názov
1.	Definovanie testovacích scenárov
2.	Výber testovacích scenárov

3.	Testovanie aplikácie
4.	Zaznamenanie výsledku testovania
5.	Vytvorenie reportov

Definovanie testovacích scenárov

Vstup: Potreba testovania

Výstup: Definovaná sada testovacích scenárov

Zodpovedný: Manažér testovania

Proces: Manažér testovania na základe potreby definovať štruktúru a rozsah testovania vytvorí testovacie scenáre.

Výber testovacích scenárov

Vstup: Definovaná sada testovacích scenárov

Výstup: Vybrané testovacie scenáre

Zodpovedný: Manažér testovania

Proces: Manažér testovania na základe implementovaných častí vyberie testovacie scenáre, ktoré sa budú testovať.

Testovanie aplikácie

Vstup: Vybrané testovacie scenáre

Výstup: otestovaná aplikácia

Zodpovedný: Tester

Proces: Tester otestuje aplikáciu pomocou manuálnych, alebo automatických testov.

Zaznamenanie výsledku testovania

Vstup: Otestovaná aplikácia

Výstup: Zoznam výsledkov testovania

Zodpovedný: Tester

Proces: Tester zaznamená výsledky testovania do testovacích scenárov. V prípade nájdenej chyby priloží komentár. Vyplnené testovacie scenáre uloží a oboznámi manažéra testovania o ukončení testovania.

Vytvorenie reportov

Vstup: Vyplnené testovacie scenáre

Výstup: Vytvorené reporty

Zodpovedný: Manažér testovania

Proces: Manažér testovania vytvorí reporty z testovania. Reporty s chybami následne zdefinuje ako úlohy pre vývojára. Reporty slúžia aj manažérovi rizík a plánovania.

Proces návrhu architektúry

Krok	Názov
1.	Definovanie architektúry
2.	Schvaľovanie architektúry
3.	Oprava architektúry
4.	Vytvorenie architektúry

Definovanie architektúry

Vstup: Potreba vytvorenia architektúry

Výstup: Definovaný návrh architektúra

Zodpovedný: Hlavný architekt

Proces: Hlavný architekt na základe analýzy definuje návrh architektúry potrebný pre projekt. Túto architektúru zdokumentuje a predloží na schválenie členom tímu.

Schvaľovanie architektúry

Vstup: Definovaná architektúra

Výstup: Schválená architektúra/ Neschválená architektúra

Zodpovedný: Hlavný architekt

Proces: Hlavný architekt predloží definovanú architektúru členom tímu na stretnutí. Na základe rozhodnutia všetkých členov tímu sa architektúra buď schváli a vytvorí, alebo sa neschváli a hlavný architekt ju musí opraviť. V prípade neschválenia je potrebné opäť predložiť upravenú architektúru na schválenie členom tímu. Hlavný architekt si zaznamená pripomienky k architektúre.

Oprava architektúry

Vstup: Neschválená architektúra

Výstup: Upravená architektúra

Zodpovedný: Hlavný architekt

Proces: Hlavný architekt na základe zaznamenaných pripomienok, upraví architektúru a zdokumentuje upravené časti. Po úprave opäť na stretnutí predloží návrh.

Vytvorenie architektúry

Vstup: Schválený návrh architektúry

Výstup: Vytvorenie architektúry

Zodpovedný: Hlavný architekt

Proces: Hlavný architekt po schválení návrhu architektúry vytvorí a spojzdí architektúru prostredia. Po vytvorení zdokumentuje postup vytvorenia.

Proces analýzy

Krok	Názov
1.	Analýza vybraného problému
2.	Vytvorenie dokumentu z analyzovaného problému

Analýza vybraného problému

Vstup: Potreba analýzy vybraného problému

Výstup: Analýza vybraného problému

Zodpovedný: Analytik

Proces: Analytik na základe vybraného problému naštuduje problém a jeho riešenia.

Vytvorenie dokumentu z analyzovaného problému

Vstup: Analýza vybraného problému

Výstup: Dokument analýzy vybraného problému

Zodpovedný: Analytik

Proces: Analytik na základe analyzovaného problému vytvorí dokument analyzovanej časti. Tento dokument slúži pre návrh a implementáciu, prípadne pre oboznámenie zákazníka s problematikou. Analytik po analýze zaznamená koniec vykonávania úlohy a výstupný dokument uloží na google drive v priečinku analýza.

9 Ostatné metodiky

9.1 Metodika zápisníc

Metodika sa ukladá na zdieľanom úložisku google drive.

Názov priečinku: InWifi dokumentácia

Názov zápisnice je v tvare: zápisnica**X** (**X** označuje číslo stretnutia)

Názov zápisnice: Zápisnica z/zo X. stretnutia (X označuje číslo stretnutia, štýl písma Arial, veľkosť písma 14, text bold, centrovane vľavo)

Zápisnica obsahuje oddiely:

Oddiel: A. Účastníci stretnutia (štýl písma Times New Roman, veľkosť písma 11, centrovane stred)

Oddiel obsahuje nasledujúcu štruktúru tabuľky:

Dátum: dd.mm.yy	Čas: hh:mm - hh:mm	Miestnosť: xx
Vedúci tímu:	Meno vedúceho tímu	
Členovia tímu:	Meno zúčastneného člena tímu 1	
	Meno zúčastneného člena tímu 2	
	Meno zúčastneného člena tímu 3	
	Meno zúčastneného člena tímu 4	
	Meno zúčastneného člena tímu 5	
	Meno zúčastneného člena tímu 6	
	Meno zúčastneného člena tímu 7	
Vypracoval:	Meno člena tímu, ktorý vypracoval zápisnicu	

Oddiel: B. Plán stretnutia (štýl písma Times New Roman, veľkosť písma 11, kurzíva, centrovane vľavo)

Oddiel obsahuje krátky popis stretnutia v bodoch

Oddiel: C. Rokovanie (štýl písma Times New Roman, veľkosť písma 11, kurzíva, centrovane vľavo)

Oddiel obsahuje nasledujúcu štruktúru tabuľky:

BOD ROKOVANIA	VÝSLEDOK
Stručný popis bodu rokovania	Výsledok prerokovaného bodu

Oddiel: D. Úlohy do ďalšieho stretnutia (štýl písma Times New Roman, veľkosť písma 11, kurzíva, centrovane vľavo)

Oddiel obsahuje úlohy na vykonanie do ďalšieho stretnutia (tento krok je potrebný v prípade, ak by manažovacie prostredie prestalo fungovať)

9.2 Metodika k stretnutiam

1. Stretnutia sa konajú v pravidelných

- a. Termín stretnutí: Každý pondelok od 8:00 do 11:00
- b. Miesto stretnutí: FIIT STU – Ilkovičová 2, 842 16 Bratislava 4, miestnosť 5.45

2. Rozoberá sa dosiahnutý progres:

- a. Pozrú sa riešené úlohy v nástroji
- b. Do akej miery sa podarilo splniť úlohy
- c. Aké problémy sa vyskytli pri riešení úloh

3. Prezentujú sa naštudované oblasti:

- a. Každý člen tímu sa postaví a prezentuje ostatným, čo si naštudoval
- b. Používa sa tabuľa pre kreslenie
- c. Diskusia k danej oblasti

4. Testujú sa vyriešené úlohy

- a. Test driven development
- b. Testuje sa funkcionálnosť
- c. Overuje sa správnosť riešení

5. Riešia sa vyskytnuté problémy

- a. Analyzuje sa problém

- b. Navrhnu sa možné riešenia
- c. Aplikuje sa konkrétne riešenie

6. Rozdelia sa úlohy na ďalší týždeň

- a. Vedúci tímu zadá úlohy
- b. Členovia tímu si rozdelia úlohy
- c. Úlohy sa nahodia do Trella
- d. Ktoré úlohy sú splnené
- e. Ktoré úlohy sú vo fáze riešenia
- f. Vytvorí sa karty v sekcii „To do“
- g. Členovia tímu si k úlohám priradia očakávaný čas strávený pri ich riešení

7. Zapiše sa priebeh stretnutia – zápisnice stretnutí

9.3 Metodika nasadzovania

Implementované a otestované časti kódu musíme spojiť dokopy, znova ich otestovať a následne nasadiť na zariadenie. Na tomto procese sa zapája viacero ľudí z radov programátorov aj testerov. Samotné nasadzovanie teda môžeme rozdeliť na tri kroky a to celkové spájanie zdrojového kódu, jeho následné testovanie a samotné nasadenie. Proces spájania zdrojového kódu môžeme rozdeliť na nasledovné kroky:

1. Programátori z úložiska GitHub vyberú časti zdrojových kódov.
2. Programátori implementujú prechody medzi týmito časťami.
3. Programátori sú povinný písať kód čitateľne.
4. Programátori sú povinný používať komentáre k podstatným častiam kódu.
5. Programátori sú povinný si ukladať záložné kópie na prenosné médium v prípade výpadku prúdu, poškodenia počítača alebo inej udalosti pri ktorej môže prísť k strate údajov.
6. Po implementácií zdrojový kód otestujú a uložia na úložisko GitHub (ako beta verziu s nasledovným poradovým číslom, pričom sa začína od 1.0) na otestovanie testermi.

Proces testovania spojeného zdrojového kódu robia testeri. Tento proces prebieha nasledovne:

1. Tester si vyberú z úložiska GitHub danú beta verziu.
2. Tester vybranú beta verziu prejdú a následne otestujú.
3. V prípade že je otestovaná beta verzia funkčná uloží sa na hlavné úložisko GitHub už spojený program.
4. V prípade že je otestovaná beta verzia nefunkčná pošle sa späť k doprogramovaniu.

V prípade že by ani tester neodhalil chybu v kóde a odhalil by ju iný člen tímu môže túto chybu opraviť, ale musí vytvoriť novú verziu aby sme sa v prípade novej chyby pri doimplementovaní mohli vrátiť k pôvodnému zdrojovému kódu.

Samotný proces nasadzovania robí developer. Zdrojové kódy ktoré nasadzuje sú už vopred spomenutými krokmi otestované. Proces nasadzovania môžeme rozdeliť na nasledujúce kroky:

1. Developer zálohuje pôvodný zdrojový kód na zariadení.
2. Developer z úložiska GitHub stiahne zdrojové kódy.
3. Developer tieto kódy uloží na zariadenie.
4. Developer pred začiatok pridaného kódu pridá komentár na ľahkú identifikáciu pridaného kódu, ako aj na koniec pridaného kódu.

Takto nasadený program sa po týchto krokoch musí testovať. V prípade že nastane chyba sa hneď môže nasadiť pôvodný funkčný zdrojový kód.

10 Záznamy zo stretnutí

Túto kapitolu tvoria jednotlivé zápisnice zo všetkých doterajších tímových stretnutí.

10.1 Zápisnica z 1. stretnutia

Účastníci stretnutia

Dátum: 21.9.2015	Čas: 18:00 - 19:30	Miestnosť: 5.45
Vedúci tímu:	Ing. Rastislav Bencel	
Členovia tímu:	Bc. Vladimír Čápka	
	Bc. Roman Kopšo	
	Bc. Kristián Košťál	
	Bc. Patrik Krajča	
	Bc. Patrik Pernecký	
	Bc. Peter Radványi	
	Bc. Dalibor Turay	
Vypracoval:	Bc. Dalibor Turay	

Plán stretnutia

Prerokovanie bodov z časti *Rokovanie*.

Stav plnenia úloh z predchádzajúceho stretnutia

Neboli žiadne plánované úlohy

Rokovanie

BOD ROKOVANIA	VÝSLEDOK
Zoznámenie členov tímu spolu s vedúcim tímu	Všetci účastníci sa zoznámili a predstavili svoje silné a slabé stránky.
Výber manažovacieho prostriedku pre správu projektu	Trello (zodpovedný za správu Roman Kopšo)
Zadanie úloh	Súpis úloh na domáce štúdium do nasledujúceho stretnutia
Oboznámenie sa s obsahom projektu	Získanie High level pohľadu

Úlohy do ďalšieho stretnutia

Úloha	Zodpovedná osoba	Termín
Vytvorenie a návrh loga	Všetci členovia tímu	28.9.2015
Štúdium 2 diplomových prác poskytnutých vedúcim tímu	Všetci členovia tímu	28.9.2015
Štúdium SDN sietí	Všetci členovia tímu	28.9.2015
Štúdium WIFI	Všetci členovia tímu	28.9.2015
Štúdium CapWap	Kristián Košťál, Roman Kopšo	28.9.2015
Štúdium Personal AP	Vladimír Čápka, Patrik Krajča	28.9.2015
Štúdium testovacích prostredí pre SDN siete	Peter Radványi, Patrik Pernecký, Dalibor Turay	28.9.2015

10.2 Zápisnica z 2. stretnutia

Účastníci stretnutia

Dátum: 28.9.2015	Čas: 8:00 - 11:00	Miestnosť: 5.45
Vedúci tímu:	Ing. Rastislav Bencel	
Členovia tímu:	Bc. Vladimír Čápka	
	Bc. Roman Kopšo	
	Bc. Kristián Košťál	
	Bc. Patrik Krajča	
	Bc. Patrik Pernecký	
	Bc. Peter Radványi	
	Bc. Dalibor Turay	
Vypracoval:	Bc. Dalibor Turay	

Plán stretnutia

Prerokovanie stavu úloh z predchádzajúceho stretnutia

Výmena znalostí medzi členmi tímu

Prerokovanie bodov z časti *Rokovanie*

Zadanie úloh do ďalšieho stretnutia

Stav plnenia úloh z predchádzajúceho stretnutia

Úloha	Zodpovedná osoba	Stav
Vytvorenie a návrh loga	Všetci členovia tímu	Dokončené
Štúdium 2 diplomových prác poskytnutých vedúcim tímu	Všetci členovia tímu	Preštudované
Štúdium SDN sietí	Všetci členovia tímu	Preštudované
Štúdium WIFI	Všetci členovia tímu	Preštudované

Štúdium CapWap	Kristián Košťál, Roman Kopšo	Preštudované
Štúdium Personal AP	Vladimír Čápka, Patrik Krajča	Preštudované
Štúdium testovacích prostredí pre SDN siete	Peter Radványi, Patrik Pernecký, Dalibor Turay	Preštudované

Rokovanie

BOD ROKOVANIA	VÝSLEDOK
Účasť na TP Cupe	Jednoznačne odsúhlasená účasť
Kontrola času v manažovacom prostredí	Vybraný: Trello plus
Výber metodiky pre ohodnotenie úloh	Veľkosť tričiek (XS,S,M,L,XL)
Vytvorenie spoločného mailu	Vytvorený spoločný email
Rozdelenie úloh medzi členov tímu	Členovia si rozdelili úlohy v Trelle
Plánovanie projektu	Naplánovaný ďalší šprint 0

Úlohy do ďalšieho stretnutia

Úloha	Zodpovedná osoba	Termín
Opakovanie diplomových prác	Všetci členovia tímu	5.10.2015
Štúdium Wifi v SDN sieťach	Kristián Košťál, Roman Kopšo, Vladimír Čápka, Patrik Krajča	5.10.2015
Štúdium a vyhľadávanie možností pre virtualizáciu Wifi v SDN sieťach pre testovacie účely	Peter Radványi, Patrik Pernecký, Dalibor Turay	5.10.2015
Pozrieť si možnosti SDN Wifi pre router ASUS RT - N16	Kristián Košťál, Roman Kopšo, Vladimír Čápka, Patrik Krajča	5.10.2015
inštalácia Mininetu (virtuálne prostredie SDN sietí)	Všetci členovia tímu	5.10.2015

10.3 Zápisnica z 3. stretnutia

Účastníci stretnutia

Dátum: 5.10.2015	Čas: 8:00 - 11:00	Miestnosť: 5.45
Vedúci tímu:	Ing. Rastislav Bencel	
Členovia tímu:	Bc. Vladimír Čápka	
	Bc. Roman Kopšo	
	Bc. Kristián Košťál	
	Bc. Patrik Krajča	
	Bc. Patrik Pernecký	
	Bc. Peter Radványi	
	Bc. Dalibor Turay	
Vypracoval:	Bc. Dalibor Turay	

Plán stretnutia

Prerokovanie stavu úloh z predchádzajúceho stretnutia

Výmena znalostí medzi členmi tímu

Prerokovanie bodov z časti *Rokovanie*

Zadanie úloh do ďalšieho stretnutia

Stav plnenia úloh z predchádzajúceho stretnutia

Úloha	Zodpovedná osoba	Stav
Opakovanie diplomových prác	Všetci členovia tímu	preštudované
Štúdium Wifi v SDN sieťach	Kristián Košťál, Roman Kopšo, Vladimír Čápka, Patrik Krajča	Preštudované
Štúdium a vyhľadávanie možností pre virtualizáciu Wifi v SDN sieťach pre testovacie účely	Peter Radványi, Patrik Pernecký, Dalibor Turay	Preštudované a vyhľadané

Pozrieť si možnosti SDN Wifi pre router ASUS RT - N16	Kristián Košťál, Roman Kopšo, Vladimír Čápka, Patrik Krajča	Preštudované a vyhl'adané
inštalácia Mininetu (virtuálne prostredie SDN sietí)	Všetci členovia tímu	Nainštalované a otestovaná funkčnosť

Rokovanie

BOD ROKOVANIA	VÝSLEDOK
Výber možnosti pre virtualizáciu Wifi v SDN siet'ach	Vybraný modul mininet- Wifi
Výber firmware-u pre router ASUS RT - N16	Vybraný firmware DD-WRT
Prezentácia a zoznámenie sa s prostredím mininet (virtuálne prostredie)	Všetci členovia si osvojili používanie mininetu

Úlohy do ďalšieho stretnutia

Úloha	Zodpovedná osoba	Termín
Vymyslieť user stories	Všetci členovia tímu	12.10.2015
Vytvoriť návrh kostry projektu	Všetci členovia tímu	12.10.2015
Vytvorenie product backlog	Všetci členovia tímu	12.10.2015
Vytvoriť Webovú stránku tímu	Vladimír Čápka, Peter Radványi	12.10.2015
Spojzdnenie OpenVSwitch na Routri	Kristián Košťál	12.10.2015
Pridanie funkcionality Wifi testovania do mininetu	Patrik Pernecký, Peter Radványi, Dalibor Turay	12.10.2015
Rozdelenie kompetencií v tíme	Všetci členovia tímu	12.10.2015
Spojzdnenie softvérového kontrolóra RYU	Roman Kopšo, Patrik Krajča	12.10.2015

10.4 Zápisnica zo 4. stretnutia

Účastníci stretnutia

Dátum: 12.10.2015	Čas: 18:00 - 19:30	Miestnosť: 5.45
Vedúci tímu:	Ing. Rastislav Bencel	
Členovia tímu:	Bc. Vladimír Čápka	
	Bc. Roman Kopšo	
	Bc. Kristián Košťál	
	Bc. Patrik Krajča	
	Bc. Patrik Pernecký	
	Bc. Peter Radványi	
	Bc. Dalibor Turay	
Vypracoval:	Bc. Dalibor Turay	

Plán stretnutia

Prerokovanie stavu úloh z predchádzajúceho stretnutia

Výmena znalostí medzi členmi tímu

Prerokovanie bodov z časti *Rokovanie*

Zadanie úloh do ďalšieho stretnutia

Stav plnenia úloh z predchádzajúceho stretnutia

Úloha	Zodpovedná osoba	Stav
Vymyslieť user stories	Všetci členovia tímu	Hotové
Vytvoriť návrh kostry projektu	Všetci členovia tímu	Hotové
Vytvorenie product backlog	Všetci členovia tímu	Hotové

Vytvoriť Webovú stránku tímu	Vladimír Čápka, Peter Radványi	Hotové
Spojzdenie OpenVSwitch na Routri	Kristián Košťál	Nepodarilo sa spojzdnit'
Pridanie funkcionality Wifi testovania do mininetu	Patrik Pernecký, Peter Radványi, Dalibor Turay	Hotové
Rozdelenie kompetencií v tíme	Všetci členovia tímu	Hotové
Spojzdenie softvérového kontrolóra RYU	Roman Kopšo, Patrik Krajča	Hotové

Rokovanie

BOD ROKOVANIA	VÝSLEDOK
Prerokovanie vymyslených user stories	Vytvorené user stories, potrebné ešte upraviť
Návrh kostry projektu	Predstavený a upravený návrh kostry projektu
Webová stránka	Vytvorená webová stránka, potrebné ešte doladiť niektoré aspekty.
Spojzdenie OpenVSwitch na rootery	Nepodarilo sa spojzdnit', je potrebné odstrániť buggy.
Pridanie funkcionality Wifi testovania do mininetu	Ukážka a oboznámenie sa s prácou wifi-mininetu
Rozdelenie kompetencií v tíme	Kompetencie boli rozdelené na základe výberu členmi tímu
Spojzdenie softvérového kontrolóra RYU	Oboznámenie ostatných členov s kontrolórom
Vytvorenie predbežného plánu šprintov	High level pohľad na beh šprintov
Prerokovanie hardvérových požiadaviek	Hardvérové požiadavky prerokované pre kontrolór RYU
Zadelenie úloh do ďalšieho týždňa	Úlohy boli zadelené a k nim boli priradený členovia.

Úlohy do ďalšieho stretnutia

Príprava 1. šprintu. Úlohy boli zaznamenané do manažovacieho systému Trello.

10.5 Zápisnica z 5. stretnutia

Účastníci stretnutia

Dátum: 19.10.2015	Čas: 8:00 - 11:00	Miestnosť: 5.45
Vedúci tímu:	Ing. Rastislav Bencel	
Členovia tímu:	Bc. Vladimír Čápka	
	Bc. Roman Kopšo	
	Bc. Kristián Košťál	
	Bc. Patrik Krajča	
	Bc. Patrik Pernecký	
	Bc. Peter Radványi	
	Bc. Dalibor Turay	
Vypracoval:	Bc. Dalibor Turay	

Plán stretnutia

Prerokovanie stavu úloh z predchádzajúceho stretnutia

Výmena znalostí medzi členmi tímu

Prerokovanie bodov z časti *Rokovanie*

Zadanie úloh do ďalšieho stretnutia

Stav plnenia úloh z predchádzajúceho stretnutia

Úloha	Zodpovedná osoba	Stav
Práca na dokumentácii	Dalibor Turay	Vytvorené dokumentácie
Pridanie funkcionality Wifi testovania do mininetu	Peter Radványi	Pridané a otestované

Spojzdnenie OpenVSwitch	Kristián Košťál	Spojzdnené s firmwerom Open-WRT 1.0
Spojzdnenie softvérového kontrolóra RYU	Roman Kopšo, Patrik Krajča	Spojzdnené
Analýza dostupných open-source softvérových kontrolórov	Roman Kopšo, Patrik Krajča	Zanalyzované, oboznámení členovia tímu
Inštalácia virtuálneho servera	Peter Radványi, Vladimír Čápka	Spojzdnené
Štúdium IEEE článkov od vedúceho tímu	Patrik Pernecký	Preštudované, oboznámení ostatní členovia tímu
Analýza štandardov 802.11i a 802.1x	Dalibor Turay	Zanalyzované, oboznámení členovia tímu

Rokovanie

BOD ROKOVANIA	VÝSLEDOK
TP cup vypracovanie prihlášky	Roman Kopšo vypracuje prihlášku.
Štúdium a porovnanie rozdielov medzi štandardami 802.11k a 802.11r	Patrik Krajča spíše a oboznámi
Navrhnuť architektúru , user stories	Patrik Pernecký navrhne architektúru
Štúdium OpenNet virtuálnej wifi v SDN sieťach	Peter Radványi naštuduje a otestuje
Návrh rizík a user stories	Vladimír Čápka vypracuje riziká a navrhne user stories
Prerokovanie zmeny DD-WRT na Open WRT	DD-WRT nebolo možné nahodiť na router , boli prehodnotené riziká a zvolila sa zmena na OpenWRT. Nasadzovanie má na starosti Kristián Košťál
Práca na dokumentácii	Dalibor Turay

Úlohy do ďalšieho stretnutia

Úlohy boli zaznamenané do manažovacieho systému Trello.

10.6 Zápisnica zo 6. stretnutia

Účastníci stretnutia

Dátum: 26.10.2015	Čas: 8:00 - 11:00	Miestnosť: 5.45
Vedúci tímu:	Ing. Rastislav Bencel	
Členovia tímu:	Bc. Vladimír Čápka	
	Bc. Roman Kopšo	
	Bc. Kristián Košťál	
	Bc. Patrik Krajča	
	Bc. Patrik Pernecký	
	Bc. Peter Radványi	
	Bc. Dalibor Turay	
Vypracoval:	Bc. Dalibor Turay	

Plán stretnutia

Review šprintu

Výmena znalostí medzi členmi tímu

Prerokovanie bodov z časti *Rokovanie*

Zadanie úloh do ďalšieho stretnutia

Rokovanie

BOD ROKOVANIA	VÝSLEDOK
Editovanie User stories	Vytvorené a pozmenené user stories.
Problém s nasadzovaním firmware-u na router	Ku Kristiánovi Košťálovi sa pridal Peter Radványi, spoločne sa budú snažiť nasadiť firmware na router.

Doplnenie rizík a ich analyzovanie	Vladimír Čápka zapísal nové riziká ku existujúcim. Všetci členovia tímu prešli riziká, ktoré boli vytvorené a diskutovali o nich.
Prerokovanie práce na dokumentácii	Dalibor Turay
Plánovanie 2. šprintu	Roman Kopšo naplánoval druhý šprint.
Oboznámenie s návrhom architektúry	Patrik pernecký oboznámil členov tímu s návrhom architektúry.
Oboznámenie ostatných členov tímu s wifi štandardami	Patrik Krajča oboznámil ostatných členov tímu s wifi štandardami 802.11 k a 802.11 r

Úlohy do ďalšieho stretnutia

Príprava 2. šprintu. Úlohy boli zaznamenané do manažovacieho systému Trello.

10.7 Zápisnica zo 7. stretnutia

Účastníci stretnutia

Dátum: 2.11.2015	Čas: 8:00 - 11:00	Miestnosť: 5.45
Vedúci tímu:	Ing. Rastislav Bencel	
Členovia tímu:	Bc. Vladimír Čápka	
	Bc. Roman Kopšo	
	Bc. Kristián Košťál	
	Bc. Patrik Krajča	
	Bc. Patrik Pernecký	
	Bc. Peter Radványi	
	Bc. Dalibor Turay	
Vypracoval:	Bc. Dalibor Turay	

Plán stretnutia

Review šprintu

Výmena znalostí medzi členmi tímu

Prerokovanie bodov z časti *Rokovanie*

Zadanie úloh do ďalšieho stretnutia

Rokovanie

BOD ROKOVANIA	VÝSLEDOK
Prehodnotenie rizík	Prehodnotenú riziká, doplnené nové riziká.
Ohodnotenie nových taskov pomocou Fibonacciho čísel	Ohodnotenú nové tasky, priradené osoby zodpovedné za tasky

Prechod na Jiru z Trela	Zamietnutý návrh prechodu. Neopodstatnený prechod , momentálne Trello postačuje našim potrebám
Prerokovanie používanej metódy v rámci fungovania tímu	Členovia tímu sa dohodli na metóde Test Driven Development, kvôli overovaniu naimplementovanej funkcionality.
Testovanie v rámci metódy	Dohodnuté automatické testy, ku ktorým sa napíše metodika.
Spísať všetky naštudované materiály do elektronickej formy	Každý člen tímu spíše naštudované materiály, poprípade postupy.
Výber chatu pre komunikáciu v tíme	Rozhodovanie medzi Hipchatom alebo Slackom. Výber chatu je Hipchat pre neskoršie možné integrovanie na JIRU.
Návrh architektúry	Patrik pernecký odprezentoval návrh architektúry, pričom je potrebné zmeniť niektoré body. Patrik dopracuje do nasledujúceho stretnutia. Názov architektúry je ASLAN.
Inštalácia RYU kontrolóra na server	Nainštalovaný RYU kontrolór na server.

Úlohy do ďalšieho stretnutia

Úlohy boli zaznamenané do manažovacieho systému Trello.

11 Exportsy úloh

11.1 Šprint 1

Date last	Due date	Card	List	S	E 1st	E	R
11.11.2015	25.10.2015	Rozdeliť si kompetencie v tíme [R]	Done				
26.10.2015	25.10.2015	Upresniť user stories	Done	3	3	3	
26.10.2015	25.10.2015	Aký je rozdiel medzi 802.11k a 802.11r. Takisto zi...	Done	1.5	5	1.5	
26.10.2015	25.10.2015	Vymyslieť architektúru [R]	Done				
26.10.2015	25.10.2015	Práca na dokumentácií	Done	12.5	20	12.5	
26.10.2015	25.10.2015	Spraviť riziká	Done	5	5	5	
26.10.2015	25.10.2015	Pridanie funkcionality Wifi testovania do mininetu	Done	14	9	14	
26.10.2015	25.10.2015	OpenVswitch spojzdenie	Done	17	4.5	17	
25.10.2015	25.10.2015	Spraviť webovú stránku tímu	Done	13	10	13	
25.10.2015	25.10.2015	Spraviť product backlog	Done	1	1	1	
25.10.2015	25.10.2015	TP cup prihláška	Done	4	6	4	
22.10.2015	25.10.2015	Rozbehať mininet na server RYU	Done	4	4	4	
19.10.2015	25.10.2015	Inštalácia virtuálneho stroja (server)	Done	15	15	15	
19.10.2015		STRETNUTIE bude o 15:00 26.10. v pondelok 5.45.	Informácie				
18.10.2015	25.10.2015	IEEE 5 článkov naštudovať	Done	8	8	8	
18.10.2015	25.10.2015	Bezpečnosť na wifi	Done	3	3	3	
13.10.2015		Číslo na Rasťa : 0915 142 667	Informácie				
12.10.2015		Kostra projektu	Informácie				
11.10.2015	25.10.2015	Spojzdenie softvérového kontrolóra RYU	Done	7.5	7	7.5	
7.10.2015	25.10.2015	Analýza dostupných open-source softvérových kontro...	Done	4	4	4	

11.2 Šprint 2

Date last	Due date	Card	List	S	E 1st	E	R
8.11.2015		Stretnutie v pondelok 8:00 9.11. v 5.45.	Informácie				
8.11.2015	8.11.2015	Naštudovať Floodlight	Done	2	2	2	
8.11.2015	8.11.2015	Naštudovať ako funguje RYU	Done	7	3.5	7	
8.11.2015	8.11.2015	Aktualizácia Webu	Done	1	1	1	
8.11.2015	8.11.2015	Práca s rizikami	Done	4	4	4	
8.11.2015	8.11.2015	Nasadenie a sfunkcnenie controller plus switch	Done	21	12	21	
8.11.2015	8.11.2015	Práca na dokumentácií	Done	17	25	31	14
8.11.2015	8.11.2015	Metodika k meetingom	Done	2	2	2	
7.11.2015	8.11.2015	Aký je rozdiel medzi 802.11k a 802.11r. Takisto zi...	Done	7	5	7	
6.11.2015	8.11.2015	Metodika nasadzovania	Done	2	2	2	
3.11.2015	8.11.2015	Metodika zápisníc	Done	2	1.5	2	
3.11.2015	8.11.2015	Spraviť Hipchat, oboznámit sa s nim a pridať ľudí	Done	1	1	1	
2.11.2015	8.11.2015	Metodika dokumentácie	Done	4	4	4	
2.11.2015	8.11.2015	Firmware na router s openflow 1.3	Done	5	5	5	

1.11.2015		Zaujímavé linky	Informácie			
28.10.2015		Dôležité body od Šimka zo stretnutia MSI	Informácie			
28.10.2015	8.11.2015	NEVYHOVUJE!!!! Simulácia v OpenNet	Done	5	5	5
28.10.2015		Naučiť sa Python [R]	To do			
26.10.2015	8.11.2015	User stories	Done	4	4	4

11.3 Šprint 3

Date last	Due date	Card	List	S	E 1 st	E	R
16.11.2015	18.11.2015	Testovanie RYU kontrolora s OF 1.3	Doing	5	10	10	5
15.11.2015	18.11.2015	Návrh virtuálneho AP	To Do	1	5	5	4
15.11.2015	18.11.2015	Návrh bezpečnosti pre architektúru	To Do	4	5	5	1
15.11.2015	18.11.2015	Testovanie integrácie RYU s wifi	Doing	4	8	8	4
9.11.2015		zadania uloh a odhadovane casy	Informácie				
9.11.2015	18.11.2015	Testovanie OpenFlow tabuliek	Doing		8	8	8
9.11.2015	18.11.2015	Testovanie RYU kontrolora s OF 1.0	Doing		1.5	2	2
9.11.2015	18.11.2015	Návrhy základných blokov architektúry	To Do		1	1	1
9.11.2015	18.11.2015	Metodika testovania	To Do		10	10	10
9.11.2015	18.11.2015	Testovanie python aplikácie	Doing		10	10	10
9.11.2015		Vzor a osnova na dokumentáciu	Informácie				