

4IT421 Zlepšování procesů budování IS

Správa požiadaviek ako disciplína RUP

Abstrakt

Táto práca si kladie za cieľ poskytnúť manažérske zhrnutie disciplíny správa požiadaviek z metodiky Rational Unified Process. V úvode sa nachádza stručný popis disciplíny. Za ním nasleduje popis významu a vlastností požiadaviek, nasledovaný dopadom tejto disciplíny na usecase modelovanie, ktoré je nasledované popisom životného cyklu správy požiadaviek. Tu sú popísané všetky aktivity a úlohy v tejto disciplíne. V závere sa zameriavam na popis rolí a artefaktov v správe požiadaviek a popis použitých pojmov.

Obsah

1. Základný popis.....	2
2. Správa požiadaviek.....	3
2.1 Požiadavky.....	3
2.2 Vlastnosti požiadaviek.....	4
3. Usecase modelovanie.....	5
4. Životný cyklus správy požiadaviek.....	6
6.3 Aktivita – Analýza problému.....	7
6.4 Aktivita – Pochopenie potrieb stakeholderov	8
6.5 Aktivita – Definícia systému	9
6.6 Aktivita – Riadenie rozsahu systému.....	9
6.7 Aktivita – Rafinácia definície systému	10
6.8 Aktivita – Správa zmenových požiadaviek.....	11
5. Artefakty a role správy požiadaviek	11
6. Slovník	14

1. Základný popis

Správa požiadaviek je zaradená medzi kľúčové procesy v rámci metodiky RUP. Jej cieľom je popísať, čo by mal systém robiť a umožniť vývojárom a zákazníkovi nájsť spoločnú zrodu pri tomto popise. Z toho vyplývajú nasledovné účely:

- Založiť a udržiavať záväznú dohodu medzi realizačným tímom, zákazníkom a stakeholdermi o tom, čo má systém robiť.
- Poskytnúť vývojárom lepšie pochopenie systémových požiadaviek.
- Definovať a vytýčiť hranice systému.
- Poskytnúť základy pre plánovanie technického obsahu jednotlivých iterácií.
- Poskytnúť základy pre odhad finančnej a časovej náročnosti vývoja systému.
- Definovať užívateľské rozhranie systému so zameraním na potreby a ciele jeho užívateľov.

K dosiahnutiu týchto cieľov je kľúčové najmä pochopenie definície a rozsahu problému, ktorý sa snažíme vyriešiť týmto systémom. Prebieha identifikácia stakeholderov a získavanie, kategorizácia a analýza ich požiadaviek. Artefakty spojené s požiadavkami sú v tejto disciplíne plne rozvinuté a presne popisujú, čo systém bude robiť, s cieľom zachytiť pohľad všetkých stakeholderov na systém, vrátane zákazníka a potenciálnych užívateľov. Disciplína správy požiadaviek je tiež spätá s nasledujúcimi procesnými disciplínami.

- Analýza a dizajn
 - Požiadavky predstavujú hlavný vstup pre túto disciplínu.
- Testovanie
 - Táto disciplína validuje systém (nielen) vzhľadom na požiadavky.
- Konfiguračný a zmenový manažment
 - Poskytuje nástroje pre riadenie zmien požiadaviek.
- Projektový manažment
 - Slúži pre prípravu plánov jednotlivých iterácií, kde artefakty správy požiadaviek sú dôležitým vstupným prvkom každého plánovania iterácie.
- Prostredie
 - Vyvíja a udržiava podporné artefakty, ktoré sú využívané počas správy požiadaviek.

Disciplína biznis modelovanie poskytuje pre správu požiadaviek kľúčové artefakty:

- Biznis pravidlá
- Biznis usecase model
- Biznis analytický model
 - Doménový model

2. Správa požiadaviek

Prečo je však správa požiadaviek tak dôležitá? Keď sa pozrieme na výsledky firiem realizujúcich softwarové projekty za posledných 17 rokov, ktoré sú pravidelne zachytávané spoločnosťou Standish Group a publikované v ich CHAOS reporte, tak vidíme istý nepriaznivý trend.

Rok	Úspech (%)	Problémy (%)	Zastavenie (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

Z výsledkov CHAOS reportov uvedených v tabuľke jasne vyplýva, že úspešnosť realizovaných projektov sa pohybuje nad jednou štvrtinou všetkých realizovaných projektov, pričom však toto číslo má rastúcu tendenciu. Alarmujúcim je však nasledujúci údaj, ktorý indikuje, že približne polovica realizovaných projektov má problémy. A na záver zastavené projekty, ktoré sa pohybujú na úrovni jednej štvrtiny realizovaných projektov. Tieto čísla sú veľmi znepokojujúce nielen pre vývojárske spoločnosti, ale aj samotných zákazníkov, ktorí môžu len dúfať, že sa ocitnú v tej šťastnejšej skupine projektov.

Prečo je tomu tak? Firmy si nie vždy uvedomujú kľúčové faktory úspechu softwarových projektov. Podľa výskumov spoločnosti Standish Group medzi najvýznamnejšie faktory úspechu projektu patria (hrubo zvýraznené položky sa týkajú správy požiadaviek):

- Exekutívna/Výkonná podpora (18 %)
- **Zahrnutie užívateľov (16 %)**
- Skúsený projektový manažér (14 %)
- **Jasné biznis ciele (12 %)**
- **Minimalizovaný rozsah (10 %)**
- Štandardná softwarová architektúra (8 %)
- **Stále hlavné požiadavky (6 %)**
- Formálna metodika (6 %)
- Spoľahlivé odhady (5 %)
- Iné (5 %)

Nesmieme zabúdať, že cieľom softwarového projektu je dodať kvalitný produkt v definovanom rozsahu, termíne a rozpočte, ktorý je v súlade s očakávaním zákazníka.

2.1 Požiadavky

Správa požiadaviek je systematický prístup pre pochopenie problému, nájdenie, usporiadanie a popis požiadaviek a tiež zmenové riadenie v rámci vývoja software. Základom je, aby sme vedeli, že riešime ten správny problém. Správa požiadaviek však nie je jednoduchá, pretože požiadavky nie sú vždy zrejmé prichádzajú z rôznych zdrojov. Tiež nemusia byť vždy jasne vyjadriteľné slovne. Takisto majú jedinečné vlastnosti alebo hodnoty. Problematické je tiež, že sa v čase menia a je zložitých ich riadiť vo väčšom počte. Pri správe požiadaviek je kľúčové rozlišovať medzi potrebami, vlastnosťami a požiadavkami. Jednak je dôležité pochopiť, rozdiely medzi jednotlivými pojmi, a tiež, do ktorej domény projektu patria. Pred tým, než popíšem jednotlivé pojmy by som ešte rád zadefinoval pojem stakeholder. Jedná sa o akúkoľvek osobu alebo organizáciu, ktorá je s projektom akokoľvek spriaznená. Teraz môžeme prejsť k popisu jednotlivých pojmov. Potreba (stakeholdera), ktorá patrí do problémovej domény, budeme rozumieť popis toho, čo predstavuje u stakeholdera úspešný projekt. Inými slovami, potreby popisujú, čo by mal systém vedieť a robiť, aby pomohol znížiť náklady na biznis procesy, zvýšiť obrát alebo splniť regulačné alebo iné obmedzenia.

Vlastnosťou budeme rozumieť službu, ktorú systém poskytne, aby naplnil jednu alebo viac potrieb stakeholderov. Nakoniec sa zameriam na požiadavku, ktorú budeme chápať ako popis schopnosti, ktorú musí systém alebo nejaká jeho komponenta mať alebo ovládať, aby naplnila kontrakt, štandard alebo nejakú vlastnosť. Vlastnosti a požiadavky sú už súčasťou domény riešenia. Požiadavky majú zásadné dopady na rozhodovanie o použitej architektúre, voľbe dizajnu, návrhu testov, či príprave dokumentácie.



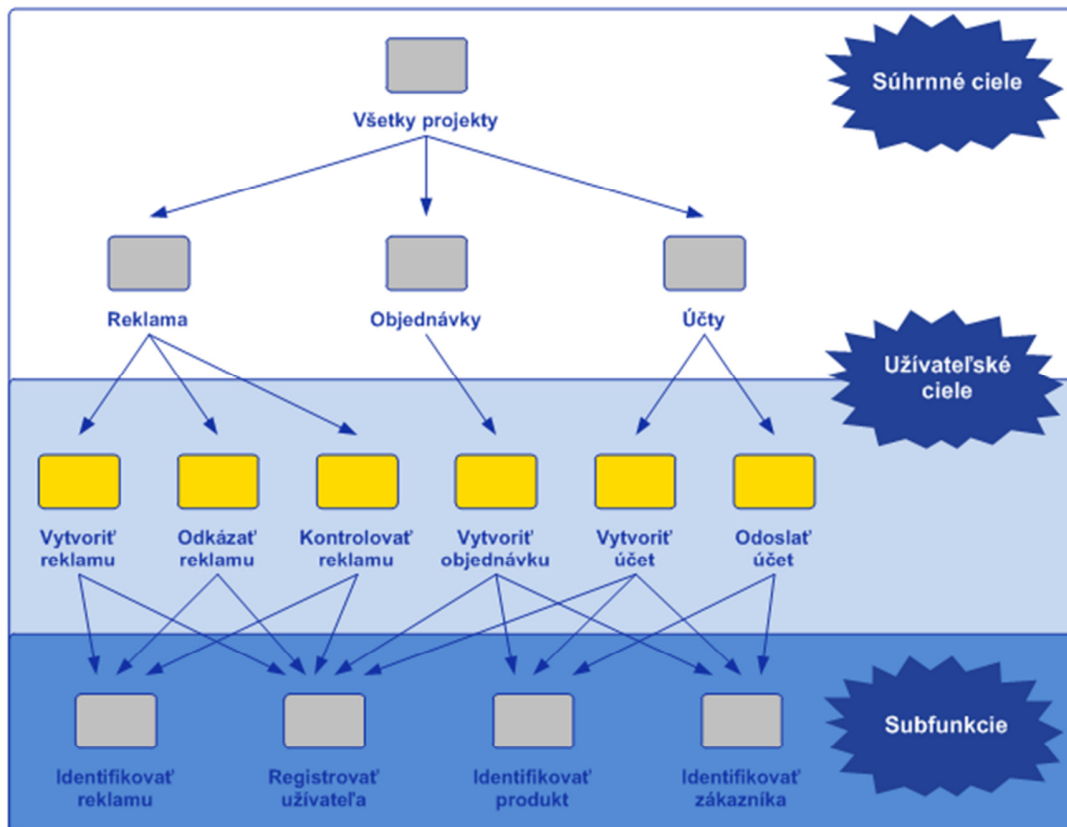
2.2 Vlastnosti požiadaviek

Kvalitná požiadavka musí spĺňať niekoľko dôležitých vlastností:

- **Korektnosť**
 - Požiadavka musí byť pravdivá ohľadom toho, čo má systém robiť.
- **Kompletnosť**
 - Požiadavka musí popisovať všetky požadované aspekty systému a zároveň všetky požiadavky musia popisovať celkový systém.
- **Konzistencia**
 - Požiadavka nesmie byť v konflikte s inou požiadavkou.
- **Jednoznačnosť**
 - Požiadavka musí mať práve jednu interpretáciu.
- **Overiteľnosť**
 - Požiadavka musí byť overiteľná, či merateľná.
- **Pochopiteľnosť**
 - Požiadavka musí byť pochopiteľná nie len užívateľmi ale aj vývojármi
- **Dohľadateľnosť**
 - Pôvod alebo dôvod vzniku požiadavky musí byť dohľadateľný.
- **Potrebnosť**
 - Pokiaľ systém zodpovedá požiadavkám užívateľov bez toho, aby bola nejaká požiadavka realizovaná, tak nebola potrebná.
- **Uskutočniteľnosť**
 - Požiadavka musí byť realizovateľná v rámci dohodnutého rozpočtu a termínu.
- **Neredundantnosť**
 - Medzi požiadavkami nesmú existovať duplicity.
- **Raditeľnosť podľa dôležitosti a stability**
 - Požiadavky musia byť raditeľné podľa klientskej, či vývojárskej dôležitosti alebo stability.

3. Usecase modelovanie

Na základe zozbieraných požiadaviek v rámci systémovej analýzy dochádza k spísaniu zadania pre vývojový tím. K tomuto v dnešnej dobe slúži popis vo forme usecase. Usecase špecifikuje množinu akcií, ktoré sú vykonávané systémom alebo subjektom, a ktoré vedú k výsledku. Tento výsledok má význam pre aktéra alebo stakeholdera. Aktérom rozumieme definovanú rolu mimo systému alebo subjekt existujúceho usecase, ktorí interagujú so systémom v rámci nejakého usecase.



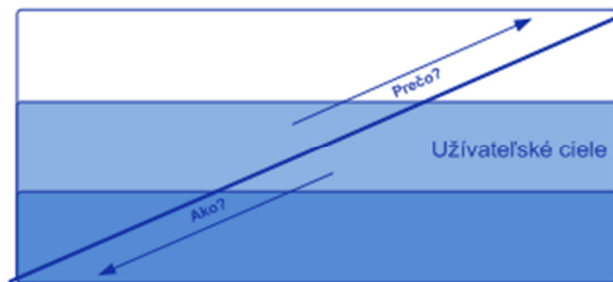
V rámci identifikácie a popisu usecase sa pohybujeme na rôznych úrovniach abstrakcie a podľa A. Cockburna rozoznávame niekoľko úrovní ako ilustruje príklad na obrázku. Z hľadiska cieľov delíme usecase na:

- Usecase zahrňujúce súhrnné ciele
 - Súhrnný usecase zahrňuje niekoľko užívateľských cieľov, tzv. kontext
- Usecase sledujúce užívateľské ciele
 - Užívateľský usecase popisuje hlavný cieľ aktéra
- Podporné usecase
 - Podporné usecase sú vyžadované pre uskutočnenie užívateľských usecase

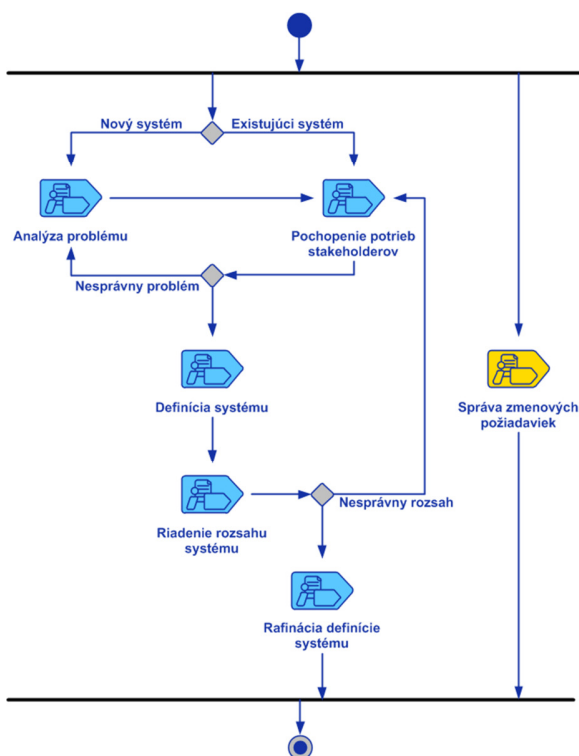
Za účelom oddelenia rôznych vrstiev abstrakcie a prehľadnenia modelu nám slúži väzba <<trace>>. Z hľadiska rozsahu delíme usecase na dve základné skupiny:

- Biznis usecase
 - Jedná sa o súhrnný usecase, ktorý popisuje operácie celej biznis problematiky (aj cez niekoľko systémov)
- Systémový usecase
 - Jedná sa o usecase popisovaný definovanými požiadavkami na systém

Niekedy býva problematické identifikovať úroveň abstrakcie, v ktorej sa identifikovaný usecase nachádza. V takomto prípade môže pomôcť nasledujúca schéma, ktorá bola skonštruovaná s rešpektom k dielu A. Cockburna. V závislosti akým smerom sa po diagonále pohybujeme sa mení aj úroveň vybraného usecase.



4. Životný cyklus správy požiadaviek



Ako každý proces, aj správa požiadaviek má svoj životný cyklus. Ako ukazuje obrázok, správa požiadaviek definuje 6 základných aktivít. Pri vývoji informačných systémov prebiehajú paralelne dva toky, kde okrem štandardných aktivít prebiehajúcich počas vývoja softwaru prebieha aj správa zmenových požiadaviek. V rámci hlavného toku je priebeh jednotlivých aktivít nasledovný. Na začiatku sa identifikuje, či sa jedná o konštrukciu nového systému, alebo o existujúci systém. Ak je to systém nový, tak je cieľom analyzovať problémovú doménu a následne pochopiť potreby stakeholderov. V prípade existujúceho systému sa prvý krok vynechá.

Ak sme v rámci analýzy dospeli k záveru, že riešime správny problém, tak pokračujeme ďalej definíciou systému, inak sa vraciame k analýze problémovej domény. Po úspešnej definícii systému je nutné riadiť a udržiavať rozsah systému v primeranej miere. Ak je správne definovaný rozsah systému, prechádza sa k rafinácii definície systému. Avšak ak je rozsah definovaný nesprávne, je nutný návrat k pochopeniu potrieb stakeholderov.

Úspešný prechod týmto tokom je nutnou podmienkou pre správnu realizáciu správy požiadaviek podľa metodiky RUP.

6.3 Aktivita – Analýza problému

Prvým krokom v každej analýze problému je uistiť sa, že všetky zainteresované strany súhlasia s tým, čo treba riešiť – alebo príležitosťou, ktorá bude napĺňaná – systémom. Aby sme predišli nepochopeniam je dôležité, aby panovala zhoda na všeobecne používanej terminológii, ktorá bude počas projektu používaná. Už spočiatku životného cyklu by malo dôjsť k definícii pojmov projektu v artefakte Slovník, ktorý by mal byť udržiavaný počas celého projektu.

Aby sme plne pochopili problém(y), ktorý riešime, je veľmi dôležité vedieť, kto sú stakeholderi v konceptuálnej vízii projektu. Títo kľúčoví stakeholderi by mali byť zahrnutí v zbere skupiny zvažovaných vlastností, ktoré budú získavané na workshope požiadaviek. Týmto vlastnostiam následne môžeme priradiť rôzne atribúty, ako napríklad racionálnu alebo relatívnu prioritu, či hodnotu, zdroj a tak ďalej, aby sme mohli riadiť ich závislosti a plán práce.

Primárnym artefaktom, v ktorom sú zhromaždené informácie získané z analýzy problému je vízia, ktorá definuje high-level pohľad na budovaný systém od užívateľov alebo zákazníkov. Vo vízii, prvotné high-level požiadavky identifikujú kľúčové vlastnosti, ktoré sú žiadané od vhodného riešenia. Tie sú typicky vyjadrené vo forme množiny high-level vlastností, ktoré by mal mať systém, aby riešil najkritickejšie problémy.

Aby sme určili počiatočný rozsah projektu, tak je nutné zhodnúť sa na hraniciach systému. Systémový analytik identifikuje užívateľov a systémy, ktoré budú s vyvíjaným systémom komunikovať. Užívatelia systému môžu byť vyjadrení ako aktéri v usecase modely.

Plán správy požiadaviek je využívaný k poskytovaniu informácii o tom, aké artefakty zo správy požiadaviek máme vytvárať, typoch požiadaviek, ktoré majú byť spravované v rámci projektu, atribúty požiadaviek, ktoré majú byť zhromažďované a prístup k dohľadateľnosti požiadaviek, ktorý budeme používať pri správe požiadaviek. Je dôležité tento plán vyvinúť dopredu.

V rámci tejto aktivity prebiehajú nasledovné úlohy:

- Identifikovať používané pojmy
 - V správe požiadaviek musí byť definovaný spoločný slovník pojmov, v ktorom sú tie najdôležitejšie pojmy z problémovej domény. Následne musia byť pojmy zo slovníka jednotne používané vo všetkých potrebných textových popisoch systému a požiadaviek na neho. Týmto spôsobom udržíme textovú konzistenciu a predídeme nepochopeniam medzi projektovými členmi o používaní, či význame pojmov. Všetky pojmy sú vedené v dokumente slovník.
- Identifikovať aktérov a usecase
 - V tejto úlohe dochádza k identifikácii aktérov a usecase, aby podporili implementáciu požiadaviek. Definícia aktérov a usecase explicitne vymedzuje hranice systému. Prvým krokom je identifikácia aktérov. Každý externý subjekt, s ktorým systém komunikuje je reprezentovaný ako aktér. Najlepší spôsob identifikácie usecase je zvážiť požiadavky jednotlivých aktérov na systém. Keďže systém existuje len kvôli svojim užívateľom, tak by mal byť postavený na ich potrebách. Následne je nutné popísať komunikáciu medzi usecase a aktérom a v prípade veľkej komplexnosti tohto modelu patrične balíčkovat aktérov a usecase. Nesmieme zabúdať ani na grafickú reprezentáciu pomocou modelov.
- Pripraviť víziu
 - Najjednoduchším spôsobom ako sa dohodnúť na definícii problému je spísať návrh a zistiť, či všetci súhlasia. Niekedy môže byť skupina zameraná na víziu riešenia, a tak pre nich môže byť problematické formulovať definíciu skutočného problému. V takých prípadoch je prínosné preskúmať prínosy riešenia a pokúsiť sa nájsť problémy, ktoré sú takto riešené. Štandardne sa k tomu využívajú techniky ako brainstorming, fishbone diagram alebo Paretove diagramy. Následne je kľúčové identifikovať stakeholderov a definovať systémové hranice. V závislosti na prostredí je možné definovať aj obmedzenia na systém. Nesmieme zabudnúť ani na definíciu vlastností systému.

- Pripraviť plán správy požiadaviek
 - Táto úloha popisuje ako vyvinúť plán dokumentácie požiadaviek, ich atribútov a smernice dohľadateľnosti a správy produktových požiadaviek. Najprv je nutné identifikovať položky, medzi ktorými chceme zaviesť dohľadateľnosť. Výsledkom je dokument obsahujúci matice dohľadateľnosti požiadaviek, ktoré sú súčasťou artefaktu Atribúty požiadaviek. Atribúty požiadaviek sa využívajú k zachytávaniu informácii spojených s dohľadateľným prvkom, typicky pre účely reportovania. Základné atribúty sú risk, benefit, náročnosť, stabilita a dopad na architektúru. Následne je nutné definovať smernice pre prepájanie informácii v rámci rôznych nástrojov.

6.4 Aktivita – Pochopenie potrieb stakeholderov

Táto aktivita pokrýva zber a získavanie informácií od stakeholderov, aby sme pochopili, aké majú potreby. Zozbierané žiadosti stakeholderov môžeme považovať za zoznam prianí, ktorý bude použitý ako primárny vstup do definície high-level vlastností systému, ako je to popísané vo vízii, ktorá riadi softwarové požiadavky, ako je to popísané v špecifikácii softwarových požiadaviek.

Hlavným cieľom je zistenie žiadostí stakeholderov za využitia prostriedkov ako sú rozhovory, rozšírené požiadavky a workshop požiadaviek. Primárnym výstupom je súbor priorizovaných vlastností a ich kritických atribútov, ktoré budú použité pri definícii systému a správe jeho rozsahu.

Tieto informácie povedú k zúšľachteniu vízie, tak ako aj k lepšiemu pochopeniu atribútov požiadaviek. Taktiež, počas trvania tejto aktivity môžete začať diskutovať o funkčných požiadavkách na systém v rovine usecase a aktérov. Tieto požiadavky, ktoré nespádajú do usecase modelu by mali byť zdokumentované v artefakte doplnková špecifikácia.

V rámci definície nových požiadaviek je dôležité dokumentovať akékoľvek závislosti. Ďalším dôležitým výstupom je aktualizovaná verzia slovníka, ktorá štandardizuje formu komunikácie zavádzaním spoločne definovaných pojmov medzi členmi tímu.

V rámci tejto aktivity prebiehajú nasledovné úlohy:

- Identifikovať používané pojmy
- Zistiť žiadosti stakeholderov
 - Táto úloha popisuje ako zisťovať od stakeholderov žiadosti na to, čo by mal systém naplňať a poskytovať. Účelom je pochopiť, kto sú stakeholderi projektu, zozbierať ich žiadosti a priorizovať ich. Na úvod je nutné formulovať otázky, na ktoré potrebujeme nájsť odpovede a následne zistiť a zadokumentovať tieto informácie. Na toto sa často krát využívajú metódy ako storyboards, rozhovory alebo dotazníky. Neraz sa v tejto fázy zorganizuje aj workshop požiadaviek, kde sa stretáva projektový tím a stakeholderi. Pripraví sa zoznam toho, čo by stakeholderi od a v systéme chceli a na záver sú položky tohto zoznamu priorizované.
- Pripraviť víziu
- Identifikovať aktérov a usecase
- Pripraviť doplnkovú špecifikáciu
 - Cieľom tejto úlohy je zachytiť požiadavky, ktoré sa neviažu na usecase model. V doplnkovej špecifikácii zachycujeme ako funkčné, tak aj nefunkčné požiadavky. Pri tvorbe doplnkovej špecifikácie sa snažíme zachytiť funkčné požiadavky, ktoré nie sú vhodné zaradiť do usecase. Následne je nutné zachytiť nefunkčné požiadavky, ktorými sú často URPS z FURPS+. Ďalej tiež zachytávame systémové obmedzenia, či už dizajnové, implementačné, rozhrania alebo fyzické. Poslednou kategóriou požiadaviek v doplnkovej špecifikácii je skupina + z FURPDS+, kde nájdeme okrem iného požiadavky na licencovanie, právne normy alebo aj oborové štandardy.
- Spravovať závislosti

- Táto úloha popisuje ako používať závislosti medzi požiadavkami pre správu, aj rozsahu projektu, aj zmenových požiadaviek. V tejto úlohe definujeme atribúty, ktoré u požiadaviek budeme sledovať a nastavujeme dohľadateľnosť projektových požiadaviek, čím napomáhame udržateľnosti rozsahu projektu a správe zmenových požiadaviek.

6.5 Aktivita – Definícia systému

Hlavným výstupom aktivity definícia systému je dohoda na rozsahu systému a tiež načrtnutie kľúčových požiadaviek na systém. Účelom tejto aktivity je začatie spájania high-level požiadaviek na rozsah systému tým, že definujeme rozsah detailných požiadaviek na systém.

Táto aktivita zahŕňa:

- Zjednotenie projektového tímu v chápaní systému.
- Vykonalie high-level analýzy zozbieraných žiadostí stakeholderov.
- Rafinácia vízie ak, aby zachytávala kľúčové vlastnosti, ktoré charakterizujú systém.
- Rafinácia usecase modelu tak, aby zahŕňal usecase a aktérov nájdených v úlohe Identifikovať aktérov a usecase
- Načrtnúť nie usecase špecifické požiadavky v Doplnkovej špecifikácii.
- Začiatok zachytávania požiadaviek v plnej miere definovaného detailu.

Aktivity, ktoré sú zamerané na analýzu problému a pochopenie potrieb stakeholderov vytvárajú prvé iterácie kľúčových systémových definícií vrátane vlastností definovaných vo vízii a prvých náčrtov detailných požiadaviek. Táto aktivita, definícia systému, sa zameriava viac na identifikáciu aktérov a usecase a na náčrt ich obsahu, tak ako aj na náčrt nie usecase špecifických požiadaviek v doplnkovej špecifikácii.

Slovník je tiež rafinovaný, zatiaľ čo sú identifikované nové projektové pojmy. Počas definície nových požiadaviek je dôležité dokumentovať akékoľvek závislosti medzi požiadavkami.

V rámci tejto aktivity prebiehajú nasledovné úlohy:

- Pripraviť víziu
- Identifikovať používané pojmy
- Identifikovať aktérov a usecase
- Pripraviť doplnkovú špecifikáciu
- Spravovať závislosti

6.6 Aktivita – Riadenie rozsahu systému

Táto aktivita zaisťuje, že požiadavky na systém sú jasné a vytvára udržateľnú skupinu prác na požiadavkách v rámci iterácie. Účelom je, aby bol rozsah systému, ktorý je vyvíjaný tak dobre popísaný, ako to len ide a zameriava sa aj na riadenú skupinu prác na požiadavkách vo vývojovom cykle.

Táto aktivita zahŕňa:

- Priorizovanie a rafinácia vstupov do výberu vlastností a požiadaviek, ktoré budú zahrnuté v súčasnej iterácii.
- Definícia skupiny behaviorálnych scenárov pre jeden alebo viac usecase, ktoré reprezentujú veľkú časť funkcionality.
- Definícia toho, ako bude spravovaná dohľadateľnosť, vrátane toho, ktoré atribúty požiadaviek a vzťahy dohľadateľnosti vôbec budú udržiavané.

- Rafinácia vízie tak, aby popisovala rozsah systému.

V rámci tejto aktivity prebiehajú nasledovné úlohy:

- Pripraviť víziu
- Spravovať závislosti
- Priorizovať usecase
 - Jedná sa o úlohu, kedy sú usecase priradené ich priority, takže poradie ich implementácie môže byť určené. Účelom tejto úlohy je definovať vstup do výberu scenárov a usecase, ktoré budú analyzované v súčasnej iterácii, ako aj definícia scenárov a usecase, ktoré podstatne zasahujú svojím rozsahom do architektonických rozhodnutí, alebo tie, ktoré popisujú špecifický, prípadne delikátny bod architektúry. Takisto je usecase pohľad zadokumentovaný do dokumentu softwarovej architektúry.

6.7 Aktivita – Rafinácia definície systému

Účelom tejto aktivity je pokračovať v rafinácii požiadaviek s cieľom zachytiť konsenzus pochopenia systémovej definície. Počas tejto aktivity dochádza k detailnej špecifikácii požiadaviek, ktoré sú spracovávané v súčasnom vývojovom cykle. Táto aktivita sa zaoberá špecifikáciou požiadaviek. Rozširuje pochopenie rozsahu projektu odzrkadlené v skupine priorizovaných produktových vlastností, ktoré sú často popísané vo vízii, ktorá je pokladaná za dosiahnuteľnú pomocou relatívne pevných rozpočtov a časových hraníc.

Výstupom je hlbšie pochopenie systémových funkčností vyjadrené v rafinovanom a detailnom popise požiadaviek v artefakte špecifikácia softwarových požiadaviek. Takisto sa jedná o náčrt behaviorálneho prototypu. Táto špecifikácia môže byť vo forme usecase modelu a doplnkovej špecifikácie, alebo v niektorých prípadoch môže dôjsť aj k tvorbe samostatného artefaktu špecifikácia softwarových požiadaviek. Tieto práce typicky začínajú revíziou existujúcich aktérov a v prípade potreby popisom aktérov, čo následne pokračuje špecifikáciou usecase, ktoré boli v predošlých aktivitách aktérovi priradené.

Kedykoľvek dôjde k zmene požiadaviek, nastáva revízia a aktualizácia všetkých atribútov dotknutých požiadaviek v zmysle popisu v aktivite správa zmenových požiadaviek.

V rámci tejto aktivity prebiehajú nasledovné úlohy:

- Špecifikovať usecase
 - V tejto úlohe sú usecase priradené detailné popisy. Účelom je popis jedného alebo viacerých tokov udalostí usecase v dostatočnej miere detailu, aby tak bolo možné tento usecase implementovať. Ďalším cieľom je popis usecase na takú úroveň, aby ho pochopili aktéri alebo zákazník a zároveň s ním boli spokojní. V rámci tejto úlohy dochádza k revízii a rafinácii scenárov, detailnému popisu a štrukturalizácii tokov, ilustrácii vzťahov medzi aktérmi a usecase, popisu špeciálnych požiadaviek, definícii komunikačných protokolov, popisu vstupných a výstupných podmienok, ako aj bodov rozšírenia.
- Pripraviť doplnkovú špecifikáciu
- Špecifikovať softwarové požiadavky
 - V tejto úlohe dochádza k detailnej špecifikácii softwarových požiadaviek na systém. Plán správy požiadaviek definuje lokalizáciu a organizáciu požiadaviek a tým priamo ovplyvňuje ako a kde sú požiadavky špecifikované. Účelom je zozbierať, špecifikovať a organizovať balíček artefaktov, ktoré plne popisujú softwarové požiadavky na systém. Špecifikácia softwarových požiadaviek zahŕňa aj špecifikáciu usecase a doplnkovej špecifikácie. Okrem špecifikácie požiadaviek prebieha aj ich balíčkovanie a generovanie reportov pre revízie.

6.8 Aktivita – Správa zmenových požiadaviek

Táto aktivita má za úlohu spravovať zmeny v požiadavkách a priraduje im celkový dopad na súčasný stav. Účelom je odhad dopadu požadovaných zmien na požiadavky a spravovať následky dopadov schválených zmien.

Táto aktivita zahŕňa:

- Vyhodnotenie vplyvu požadovanej zmeny a dopadu na existujúce softwarové požiadavky.
- Štrukturalizácia usecase modelu, ktorá má za následok zlepšenie celkovej správy požiadaviek dokumentovaných usecase modelom.
- Správa zmien v atribútoch požiadaviek a vzťahoch dohľadateľnosti.
- Overenie, že výsledky analýzy súhlasia s užívateľským pohľadom na systém, keď už dôjde k implementácii zmeny.

Zmeny požiadaviek prirodzene vplývajú na artefakty závislé na výstupoch správy požiadaviek (napríklad artefakty analýzy a dizajnu, testovania, nasadenia). Vzťahy dohľadateľnosti identifikované a zdokumentované v úlohe správa závislostí presne definujú vzťahy medzi požiadavkami a inými artefaktmi. Tieto vzťahy sú kľúčom pre pochopenie dopadu požadovanej zmeny.

V rámci tejto aktivity prebiehajú nasledovné úlohy:

- Štrukturalizovať usecase model
 - Toto je úloha, kde usecase dostáva štruktúru, vďaka ktorej sú požiadavky jednoduchšie pochopiteľné a aj udržiavateľné. Účelom je vytiahnuť z usecase správanie, ktoré môžeme považovať za abstraktný usecase. Obdobne sa snažíme identifikovať abstraktných užívateľov, ktorí definujú role zdieľané niekoľkými aktérmi. Začíname identifikáciou spoločných požiadaviek. Následne zavedieme vzťahy include, extend a dedičnosti medzi usecase a vzťahy dedičnosti medzi aktérmi. Na záver postupne organizujeme obsah usecase modelu do balíčkov.
- Spravovať závislosti
- Revidovať požiadavky
 - Táto úloha popisuje spôsob revízie artefaktov správy požiadaviek. Účelom je formálne overenie toho, že výsledky úloh správy požiadaviek sa zhodujú s pohľadom zákazníka na systém. Taktiež dochádza k plánovaniu pravidelných revízných stretnutí.

5. Artefakty a role správy požiadaviek

V disciplíne správa požiadaviek vystupujú nasledovné role (v tabuľke uvedené spolu s úlohami a artefaktmi, za ktoré sú zodpovedné):

Poradie	Rola	Úlohy	Artefakty
1	Špecifikátor požiadaviek (Requirements specifier)	<ul style="list-style-type: none"> • Pripraviť doplnkovú špecifikáciu • Špecifikovať softwarové požiadavky • Špecifikovať usecase 	<ul style="list-style-type: none"> • Usecase • Artefakt • Softwarová požiadavka • Špecifikácia softwarovej požiadavky
2	Softwarový architekt (Software architekt)	<ul style="list-style-type: none"> • Priorizovať usecase 	<ul style="list-style-type: none"> • Dokument softwarovej architektúry
3	Systémový analytik (System analyst)	<ul style="list-style-type: none"> • Identifikovať používané pojmy • Pripraviť plán správy požiadaviek • Pripraviť doplnkovú špecifikáciu • Pripraviť víziu 	<ul style="list-style-type: none"> • Slovník • Atribúty požiadaviek • Plán správy požiadaviek • Žiadosť stakeholdera

		<ul style="list-style-type: none"> • Zistiť žiadosti stakeholderov • Identifikovať aktérov a usecase • Spravovať závislosti • Štrukturalizovať usecase model 	<ul style="list-style-type: none"> • Storyboard • Doplnková špecifikácia • Usecase model • Usecase balíčky • Vízia
4	Technický posudzovateľ (Technical reviewer)	<ul style="list-style-type: none"> • Posúdenie požiadaviek 	<ul style="list-style-type: none"> • Záznam posudku

Nasledovná tabuľka poskytuje základný prehľad artefaktov správy požiadaviek s ich krátkym popisom a zodpovednou rolou.

Artefakt	Slovník
Rola	Systémový analytik
Popis	Definuje dôležité pojmy pre projekt. Existuje len jeden slovník pre celý systém, ktorý prináša konzistentnú množinu definícií, vďaka ktorej predchádzame nedorozumeniam a nepochopeniam. Členovia projektového tímu tento artefakt spočiatku využívajú pre pochopenie projektovo špecifických pojmov. Tento dokument je tiež dôležitý pre vývojárov, analytikov a tých, čo pripravujú školenia a technickú dokumentáciu.
Vstup do	Identifikovať aktérov a usecase
Výstup z	Identifikovať používané pojmy
Artefakt	Iteračný plán
Rola	-
Popis	Tento artefakt je časovou sekvenciou aktivít a úloh, ktoré majú pridelené zdroje, obsahujú závislosti na iných úlohách v danej iterácii. Jedná sa teda o podrobný plán. Iteračný plán slúži projektovému manažérovi k plánovaniu iteračných úloh, zdrojov a k zachyteniu reálnej práce voči plánom. Vďaka iteračnému plánu vedia členovia tímu, čo a kedy majú robiť a aké iné úlohy na tom závisia.
Vstup do	Pripraviť plán správy požiadaviek, Pripraviť víziu, Identifikovať aktérov a usecase
Výstup z	-
Artefakt	Atribúty požiadaviek
Rola	Systémový analytik
Popis	Artefakt popisujúci repozitár projektových požiadaviek, atribútov a závislostí. Vďaka tomu pomáha v správe zmien z pohľadu požiadaviek. Okrem iného tiež poskytuje repozitár textu, atribútov a dohľadateľnosti pre všetky požiadavky. Mal by byť dostupný každému vo vývojovom tíme.
Vstup do	-
Výstup z	Pripraviť víziu, Identifikovať aktérov a usecase
Artefakt	Plán správy požiadaviek
Rola	Systémový analytik
Popis	Artefakt popisuje artefakty správy požiadaviek, typy požiadaviek a ich konkrétne atribúty, čím špecifikuje informácie, ktoré majú byť zozbierané a definuje kontrolný mechanizmus, ktorý bude využívaný na meranie, reportovanie a kontrolovanie zmien v produktovej špecifikácii. Účelom je popísať ako projektu nastaviť a spravovať artefakty správy požiadaviek.
Vstup do	Pripraviť plán správy požiadaviek
Výstup z	Pripraviť plán správy požiadaviek
Artefakt	Záznam posudku
Rola	Technický posudzovateľ
Popis	Artefakt vytvorený, aby zachytil výsledky revízných aktivít nad niekoľkými projektovými artefaktmi.
Vstup do	-
Výstup z	Posúdenie požiadaviek
Artefakt	Zoznam rizík
Rola	-

Popis	Jedná sa o artefakt, ktorý je zoradeným zoznamom známych a neuzatvorených rizík projektu, ktoré sú radené podľa dôležitosti.
Vstup do	Priorizovať usecase
Výstup z	-
Artefakt	Dokument softwarovej architektúry
Rola	Softwarový architekt
Popis	Artefakt poskytujúci zrozumiteľný prehľad architektúry systému, využívajúci mnoho rôznych pohľadov, ktoré popisujú rôzne aspekty systému. Slúži ako komunikačné médium medzi softwarovým architektom a ostatnými členmi tímu v kontexte rozhodnutí, ktoré majú vážny dopad na architektúru systému.
Vstup do	Priorizovať usecase
Výstup z	Priorizovať usecase
Artefakt	Plán vývoja software
Rola	-
Popis	Artefakt je zrozumiteľným zhrnutím všetkých informácií nutných pre správu projektu. Zahŕňa mnoho artefaktov vytvorených počas inepčnej fázy a je spravovaný počas celého projektu. Popisuje prístup k vývoju software a je high-level plánom používaným manažmentom pre riadenie vývoja.
Vstup do	Pripraviť plán správy požiadaviek
Výstup z	-
Artefakt	Softwarová požiadavka
Rola	-
Popis	Artefakt popisujúci podmienku alebo schopnosť, ktorú musí systém spĺňať, či mať. Ide o základný artefakt správy požiadaviek.
Vstup do	Identifikovať používané pojmy
Výstup z	-
Artefakt	Špecifikácia softwarových požiadaviek
Rola	-
Popis	Tento artefakt sa zameriava na zber a organizáciu všetkých požiadaviek v rámci projektu. Tento dokument kontroluje vývin systému cez vývojovú fázu projektu, kedy sú pridávané nové alebo upravované existujúce vlastnosti vo vízii.
Vstup do	Identifikovať používané pojmy
Výstup z	-
Artefakt	Žiadosti stakeholderov
Rola	Systémový analytik
Popis	Artefakt, ktorý obsahuje všetky druhy žiadostí stakeholderov na vyvíjaný systém. Môže tiež obsahovať referencie na rôzne externé zdroje, ktorým má systém zodpovedať.
Vstup do	Pripraviť víziu, Identifikovať aktérov a usecase
Výstup z	-
Artefakt	Storyboard
Rola	Systémový analytik
Popis	Artefakt je logickým, ale aj konceptuálnym popisom toho, aká má byť systémová funkcionálna pre špecifický scenár, ktorá zahŕňa interakciu medzi užívateľmi a systémom. Dôležité je vedieť, že cieľom tohto artefaktu je pochopiť celkové toky a interakcie, nie prototypovať alebo testovať užívateľské rozhranie.
Vstup do	Identifikovať používané pojmy
Výstup z	-
Artefakt	Doplnková špecifikácia
Rola	Systémový analytik
Popis	Artefakt zachytávajúci systémové požiadavky, ktoré nie sú zapísané v behaviorálnych požiadavkách ako je napríklad usecase model. Jedná sa napríklad o právne predpisy alebo normy, atribúty kvality, či požiadavky na prostredie nasadenia.
Vstup do	Identifikovať používané pojmy

Výstup z	-
Artefakt	Usecase model
Rola	Systémový analytik
Popis	Tento artefakt predstavuje model požadovaných funkčností systému a jeho prostredia a slúži ako kontrakt medzi zákazníkom a vývojármi. Slúži ako základný vstup pre aktivity v analýze a dizajne a testovaní.
Vstup do	Identifikovať používané pojmy
Výstup z	Identifikovať aktérov a usecase
Artefakt	Prototyp užívateľského rozhrania
Rola	-
Popis	Artefakt, ktorý je príkladom užívateľského rozhrania. Slúži na tvorbu alebo schválenie užívateľského rozhrania.
Vstup do	Špecifikovať softwarové požiadavky
Výstup z	-
Artefakt	Vízia
Rola	Systémový analytik
Popis	Je artefaktom, ktorý popisuje pohľad stakeholderov na vyvíjaný software, ktorý je špecifikovaný kľúčovými potrebami a vlastnosťami. Obsahuje aj náčrt kľúčových projektových požiadaviek, čím poskytuje bázu pre detailnejší popis.
Vstup do	Identifikovať používané pojmy, Pripraviť víziu, Identifikovať aktérov a usecase
Výstup z	Pripraviť víziu

6. Slovník

Pojem	Vysvetlenie
Aktér	Je definovaná rola mimo systém alebo subjekt nejakého usecase, ktorý so systémom komunikuje v kontexte usecase
Artefakt	V rámci oficiálnej terminológie produkt práce. Jedná sa o hmotné výstupy z rôznych úloh v rámci RUPu. Môže ním byť model, element modelu alebo dokument.
High-level	Popisuje situáciu, kedy sa nejaký subjekt nachádza, alebo je popisovaný na vyššej úrovne abstrakcie, čím sa zbavujeme v danej situácii nepotrebných špecifik konkrétného subjektu.
Stakeholder	Jedná sa o akúkoľvek osobu, ktorá má väzbu na vyvíjaný systém.
Usecase	Špecifikuje množinu akcií, ktoré systém alebo subjekt vykoná a to povedie k očakávanému výsledku. Tento výsledok má zvyčajne význam pre aktéra alebo stakeholdera.

Referencie

VŠE. KITSCM.VSE.CZ [online]. [cit. 2011-11-19]. Classic RUP for SOMA. Dostupné z WWW:

<https://kitscm.vse.cz/RUP/LargeProjects/#core.base_rup/guidances/supportingmaterials/introduction_to_rup_36B63436.html>

AKED, Mark. Risk reduction with the RUP phase plan. *IBM Developer Works* [online]. 25.11.2003, [cit. 2011-10-23]. Dostupný z WWW: <<http://www.ibm.com/developerworks/rational/library/1826.html#N100E4>>.

Best Practices for Software Development Teams. In Rational Software. *Rational Unified Process* [online]. : Rational Software, 2001 [cit. 2011-10-23]. Dostupné z WWW:

<http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf>.

KRUCHTEN, Philippe. What Is the Rational Unified Process?. *The Rational Edge* [online]. 2001, [cit. 2011-10-23]. Dostupný z WWW:

<<http://perso.enstimac.fr/~journeau/GSI/MDP-info/Pr%E9sentation%20RUP/WhatIsTheRationalUnifiedProcessJan01.pdf>>.

MCEWEN, Scott. Requirements: An introduction. *IBM Developer Works* [online]. 16.04.2004, [cit. 2011-10-23]. Dostupný z WWW:

<<http://www.ibm.com/developerworks/rational/library/4166.html>>.

The Standish Group. CHAOS. *THE STANDISH GROUP REPORT* [online]. 1995, [cit. 2011-10-23]. Dostupný z WWW:

<<http://www.projectsmart.co.uk/docs/chaos-report.pdf>>.

PASSING, Johannes. *Requirements Engineering in the Rational Unified Process*. Potsdam : Hasso-Plattner-Institute for Software Engineering, 2007. 16 s.

WEILKIENS, Tim. *UML 2 Certification Guide: Fundamental & Intermediate Exams*. Morgan Kaufmann; 1 edition, 18.12.2006. 320 s.

Skriptá z Unicorn College. *Pod kódou UCL-BT:REQ10S.CZ/LEC01/GL* [online]. 25.02.2010 [cit. 2011-12-09]. Dostupné z WWW:

<<https://unicornuniverse.eu/>>.