



PROJEKT

Multitenantné operačné centrum kybernetickej bezpečnosti riešené
ako otvorená cloudová služba s prvkami strojového učenia

Previazané s balíkom KPB4 – Architektúra riešenia

Previazané s výstupom

V10 – Zoznam funkčných a nefunkčných požiadaviek
a V11 – Biznis architektúra



Obsah

1.	Úvod.....	3
2.	Podstata softvérových požiadaviek.....	3
3.	Požiadavky z pohľadu zákazníka	5
4.	Dobré praktiky inžinierstva požiadaviek	5
5.	Úloha business analytika.....	5
6.	Stanovenie obchodných požiadaviek.....	6
7.	Hlas používateľa	6
8.	Elicitácia požiadaviek.....	7
9.	Pochopenie používateľských požiadaviek.....	7
10.	Dodržiavanie pravidiel.....	8
11.	Dokumentovanie požiadaviek	8
12.	Písanie kvalitných požiadaviek	10
13.	modelovanie požiadaviek	12
14.	Špecifikovanie dátových požiadaviek	12
15.	Nad rámec funkcionality	13
16.	Znižovanie rizika prototypovaním.....	13
17.	Stanovenie priorít požiadaviek.....	14
18.	Validácia požiadaviek.....	14
19.	Opätovné použitie požiadaviek	16
20.	Nad rámec vývoja požiadaviek.....	16
21.	Agilné projekty	17
22.	Projekty rozvoja a nahradenia.....	17
23.	Projekty s krabicového SW.....	18
24.	Outsourcing projektov	18
25.	Projekty automatizácie obchodných procesov	19
26.	Projekty business analytiky.....	19
27.	Projekty vstavaných a reálnych časových systémov	19
28.	Praktiky riadenia požiadaviek.....	20
29.	Zmeny	20
30.	Väzby v reťazci požiadaviek.....	21
31.	Nástroje na inžinierstvo požiadaviek.....	21
32.	Zlepšovanie procesov požiadaviek.....	22
33.	Požiadavky a riadenie rizík	22
	Zoznam zdrojov	23

1. ÚVOD

Dokument tvorí stručný prehľad metodík zberu požiadaviek a prináša praktické rady, metódy a nástroje, ktoré majú pomôcť pri identifikácii, analýze, špecifikácii, overovaní a riadení požiadaviek.

Význam požiadaviek rastie, keďže zlyhanie v ich definovaní je častou príčinou neúspešných IT projektov. Cieľom je ukázať, ako sa týmto problémom vyhnúť pomocou overených prístupov k požiadavkám.

2. PODSTATA SOFTVÉROVÝCH POŽIADAVIEK

Čo sú softvérové požiadavky?

Softvérové požiadavky predstavujú presný popis toho, čo má systém robiť, *ako to má robiť* a *akým spôsobom* bude komunikovať s používateľom či inými systémami. Zároveň definujú obmedzenia, v rámci ktorých sa vývoj a prevádzka softvéru musí pohybovať (napr. regulačné, technické, bezpečnostné podmienky).

Požiadavky slúžia ako formálna dohoda medzi zákazníkom a vývojovým tímom a zabezpečujú, že výsledok bude zodpovedať očakávaniam používateľov.

Prečo sú požiadavky dôležité?

- Jasná komunikácia – umožňujú zjednotiť pohľad zákazníka, používateľov, vývojárov a testerov
- Predchádzanie nedorozumeniam – ak sa požiadavky definujú vopred a dôkladne, minimalizujú sa riziká nesprávnej interpretácie
- Základ pre testovanie – testerí používajú požiadavky ako referenciu, aby overili, či výsledný produkt spĺňa dohodnuté kritériá
- Podpora plánovania – rozsah projektu, časové odhady, rozpočet a potrebné zdroje sa odvíjajú práve od požiadaviek
- Zníženie nákladov na zmeny – čím skôr sa požiadavky spresnia, tým lacnejšie sa riešia ich úpravy

Typy požiadaviek

Autori rozlišujú niekoľko vrstiev požiadaviek:

1. Obchodné požiadavky (business requirements)
 - najvyššia úroveň
 - vysvetľujú, prečo projekt vzniká, aké má ciele, aké prínosy prinesie
 - bývajú zachytené napr. vo vízii alebo dokumente o rozsahu
2. Používateľské požiadavky (user requirements)
 - popisujú, čo očakávajú jednotlivé typy používateľov od systému
 - sú často vo forme používateľských príbehov, use cases alebo scenárov
3. Funkčné požiadavky (functional requirements)
 - určujú, aké funkcie má systém poskytovať
 - napr. „systém umožní registráciu nového používateľa“
4. Nefunkčné požiadavky (nonfunctional requirements)
 - opisujú kvalitatívne atribúty systému
 - napr. výkonnosť, spoľahlivosť, dostupnosť, bezpečnosť

Úrovne práce s požiadavkami

Kniha vysvetľuje takzvané tri úrovne práce s požiadavkami:

- obchodná úroveň – zachytáva problémy a ciele organizácie
- používateľská úroveň – určuje, čo očakáva používateľ pri interakcii so systémom
- úroveň riešenia – špecifikuje detailné funkcie, obrazovky, pravidlá a technické detaily

Tieto úrovne by mali byť prepojené, aby sa požiadavky od stratégie po konkrétnu funkcionálnosť dali trasovať.

Problémy s požiadavkami

Autori upozorňujú na typické problémy, ktoré vznikajú, ak sa požiadavky nerobia kvalitne:

- neúplné alebo nepresné požiadavky
- nejasnosti a viacvýznamové pojmy
- nedostatočná angažovanosť používateľov
- nekontrolované zmeny počas vývoja
- prílišná orientácia len na funkcie bez kvalitatívnych požiadaviek
- opomenutí zainteresovaných (stakeholderi), ktorí potom s výsledkom nesúhlasia

Tieto riziká často vedú k takzvanému „gold platingu“ (robia sa funkcie navyše, ktoré nikto nepotreboval), alebo k úplnému zlyhaniu projektu.

Prínosy kvalitného procesu požiadaviek

Ak sa proces požiadaviek urobí správne, prináša:

- vyššiu spokojnosť zákazníka
- zníženie počtu chýb
- lepšiu predvídateľnosť rozpočtu a času
- jednoduchšie testovanie
- kvalitnejší výsledok

Kto má požiadavky tvoriť a manažovať?

- business analytici – sprostredkujú komunikáciu medzi biznisom a vývojom
- projektoví manažéri – zabezpečujú, aby požiadavky boli zapracované do plánu
- architekti a dizajnéri – určujú technické riešenie podľa požiadaviek
- tester – overujú, či implementácia vyhovuje požiadavkám

Každý v tíme by mal chápať význam požiadaviek a aktívne sa zapojiť do ich tvorby, validácie a správy.

Zhrnutie

Táto kapitola zdôrazňuje, že dobré požiadavky nie sú len dokument, ale neustály dialóg medzi zainteresovanými stranami, ktorý trvá od začiatku projektu až po jeho ukončenie.



3. POŽIADAVKY Z POHLĎADU ZÁKAZNÍKA

- **Medzera očakávaní**
Často sa stáva, že to, čo zákazník očakáva, a to, čo vývojári dodajú, sa líši. Preto je dôležité aktívne zapojiť zákazníka do definovania požiadaviek.
- **Kto je zákazník?**
Nemusí to byť len koncový používateľ, ale aj manažéri, prevádzka, regulačné orgány.
- **Práva a povinnosti zákazníka**
Kniha uvádza tzv. *Bill of Rights* (práva zákazníka) a *Bill of Responsibilities* (zodpovednosti zákazníka), aby boli očakávania obojstranne jasné.
- **Dohoda o požiadavkách**
Treba sa dohodnúť na požiadavkách a „zmraziť“ ich ako *baseline*, aby počas vývoja nedochádzalo k neustálym nekontrolovaným zmenám.

4. DOBRÉ PRAKTIKY INŽINIERSTVA POŽIADAVIEK

- **Rámec vývoja požiadaviek**
pozostáva z týchto fáz:
 1. elicitácia (zber požiadaviek)
 2. analýza a modelovanie
 3. špecifikácia
 4. validácia (overovanie)
 5. riadenie požiadaviek
- **Kľúčové odporúčania**
 - používať štandardizované šablóny
 - aktívne zapájať všetky zainteresované strany
 - validovať požiadavky včas
 - sledovať ich stav počas celého projektu
- **Začíname s novými praktikami**
Autori odporúčajú začať postupne – nepresadzovať naraz veľké zmeny v procesoch, ale zlepšovať existujúce postupy krok za krokom.

5. ÚLOHA BUSINESS ANALYTIKA

- **Business analytik**
Je kľúčovou osobou pri transformácii potrieb zákazníka na jasné a overiteľné požiadavky.



- **Úlohy analytika**

- sprostredkovanie medzi zákazníkom a vývojovým tímom
- zisťovanie potrieb
- tvorba špecifikácií
- pomoc pri validácii požiadaviek

- **Zručnosti a vedomosti analytika**

- komunikačné schopnosti
- znalosť domény
- modelovanie procesov
- schopnosť facilitácie workshopov

- **Analytik na agilných projektoch**

Aj v agilných tímoch zohráva analytik dôležitú rolu, hoci sa časti jeho úloh prekrývajú s rolou product ownera.

6. STANOVENIE OBCHODNÝCH POŽIADAVIEK

- **Čo sú obchodné požiadavky?**

Ide o požiadavky popisujúce ciele, prínosy a dôvod existencie projektu. Slúžia ako základ pre definovanie rozsahu.

- **Kľúčové prvky**

- *vízia*: dlhodobá predstava, kam projekt smeruje
- *rozsah*: čo sa bude riešiť a čo už nie
- *obmedzenia*: regulačné, technické či organizačné faktory

- **Konfliktné požiadavky**

Obchodné požiadavky môžu byť v konflikte s požiadavkami rôznych skupín používateľov — treba ich zosúladiť a prioritizovať.

- **Dokument vízie a rozsahu**

Autori odporúčajú formalizovať tieto informácie do dokumentu, ktorý bude záväzný a schválený všetkými stranami.

7. HLAS POUŽÍVATEĽA

- **Identifikácia používateľov**

Používateľov je potrebné rozdeliť do tried (*user classes*), napríklad bežní používatelia, správcovia systému, externí zákazníci.

- **Persony**

Používa sa technika *personas*, kde sa vytvoria modelové profily typických používateľov, aby sa lepšie pochopili ich potreby.



- **Produktový šampión**
Autori zdôrazňujú úlohu tzv. *product champion* — človeka, ktorý zastupuje potreby používateľov a pomáha prioritizovať požiadavky.
- **Zvládanie protichodných požiadaviek**
Konflikty medzi rôznymi triedami používateľov treba riešiť dohodou a aktívnym zapojením zástupcov.

8. ELICITÁCIA POŽIADAVIEK

- **Techniky elicítácie**
 - rozhovory
 - workshopy
 - dotazníky
 - pozorovanie
 - analýza existujúcich dokumentov
 - prototypovanie
- **Plánovanie elicítácie**
Pred začiatkom zberu požiadaviek je nutné vypracovať plán — kto sa zúčastní, aké techniky sa použijú, aký je harmonogram.
- **Postup**
 1. príprava (zhromaždenie podkladov)
 2. vedenie rozhovorov, workshopov
 3. zaznamenanie poznámok
 4. overenie výsledkov s účastníkmi
- **Riziká elicítácie**
 - nesprávne interpretované odpovede
 - neúplné informácie
 - predpoklady, ktoré nie sú vyslovené

Autori preto odporúčajú všetky zistené požiadavky priebežne overovať.

9. POCHOPENIE POUŽÍVATEĽSKÝCH POŽIADAVIEK

- **Use cases a user stories**
 - *use case*: popis interakcie používateľa so systémom na dosiahnutie cieľa
 - *user story*: krátke, agilné vyjadrenie požiadavky, napr. „Ako používateľ chcem...“
- **Identifikácia scenárov použitia**
Tieto scenáre pomáhajú objaviť všetky funkcie a situácie, s ktorými používatelia prídu



do kontaktu.

- **Validácia scenárov**
Používateľské príbehy alebo prípadové štúdie treba overovať spoločne so zákazníkom, aby sa predišlo chybám.
- **Výhody orientácie na použitie**
 - lepšie pochopenie, čo používateľ skutočne potrebuje
 - väčšia angažovanosť zainteresovaných strán
 - kvalitnejší návrh systému

10. DODRŽIAVANIE PRAVIDIEL

- **Obchodné pravidlá (business rules)**
Ide o predpisy, fakty, normy alebo obmedzenia, ktoré ovplyvňujú fungovanie systému. Môžu byť:
 - fakty (napr. „evidujeme iba registrovaných zákazníkov“)
 - obmedzenia (napr. „vek zákazníka musí byť minimálne 18 rokov“)
 - pravidlá umožňujúce akciu (napr. „pri prekročení sumy 100 € sa ponúka zľava“)
 - výpočty a odvodeniny
- **Vzťah k požiadavkám**
Obchodné pravidlá často generujú nové funkčné alebo nefunkčné požiadavky.
- **Dokumentovanie pravidiel**
Dôležité je tieto pravidlá evidovať oddelene od funkčných požiadaviek, aby sa dali udržiavať a meniť bez zásahu do základnej špecifikácie systému.
- **Hľadanie obchodných pravidiel**
Autori odporúčajú diskusie s odborníkmi na doménu, analýzu dokumentov a kontrolu regulačných požiadaviek.

11. DOKUMENTOVANIE POŽIADAVIEK

Význam dokumentovania požiadaviek

Dokumentovanie požiadaviek je kľúčovou činnosťou, pretože zabezpečuje, aby všetci členovia tímu, ako aj zákazníci a zainteresované strany mali rovnaké chápanie toho, čo sa má dodať. Požiadavky musia byť zaznamenané zrozumiteľne, konzistentne a v takej forme, aby podporovali komunikáciu a následnú implementáciu.

Správne napísaný dokument požiadaviek slúži ako:

- zmluva medzi zákazníkom a dodávateľom
- základ pre návrh systému
- referenčný bod pre testovanie
- nástroj riadenia zmien počas celého projektu



Špecifikácia softvérových požiadaviek (SRS)

Špecifikácia softvérových požiadaviek (Software Requirements Specification – SRS) predstavuje formálny dokument, ktorý zhromažďuje všetky relevantné informácie o požiadavkách.

Obsahuje:

- funkčné požiadavky
- nefunkčné požiadavky
- požiadavky na rozhrania
- dátové požiadavky
- obmedzenia a predpoklady

Autori odporúčajú využívať osvedčené šablóny pre SRS, aby bol dokument ucelený, prehľadný a ľahko pochopiteľný.

Odporúčaná štruktúra SRS

Bežná šablóna môže obsahovať:

1. Úvod
 - účel dokumentu
 - rozsah systému
 - definície pojmov
 - referencie
2. Celkový opis
 - perspektíva systému
 - funkcie systému
 - používateľské charakteristiky
 - obmedzenia
 - predpoklady
3. Funkčné požiadavky
 - zoznam požadovaných funkcií
 - spracovanie vstupov a výstupov
 - reakcie na udalosti
4. Nefunkčné požiadavky
 - kvalitatívne atribúty (výkon, spoľahlivosť, bezpečnosť, dostupnosť)
 - regulačné požiadavky
5. Požiadavky na rozhrania
 - hardvérové rozhrania
 - softvérové rozhrania
 - komunikačné protokoly
6. Dáta
 - dátové štruktúry
 - popis databáz
 - požiadavky na migráciu údajov
7. Ostatné požiadavky
 - lokalizácia
 - prístupnosť
 - právne a regulačné aspekty

Vlastnosti kvalitného dokumentu požiadaviek

- konzistentný – nemá rozpory v texte
- •
- úplný – obsahuje všetky kľúčové informácie
- jednoznačný – dovoľuje len jeden výklad
- overiteľný – umožňuje definovať testy, ktoré overia splnenie požiadavky
- sledovateľný – dá sa spätne vystopovať, odkiaľ požiadavka vznikla

Dokumentovanie neúplných alebo neistých požiadaviek

- Ak počas analýzy nie je všetko známe, je dôležité tieto medzery v dokumente jasne označiť.
- Neúplné požiadavky by mali byť zapísané s poznámkou, že sú otvorené a budú doplnené v ďalšej fáze.
- Tím musí mať plán, ako tieto neistoty vyriešiť (napr. doplniť prototypovaním alebo rozhovormi).

Zrozumiteľnosť a čitateľnosť

- používať jednoduchý jazyk bez nadbytočných technických pojmov
- využívať odrážky, číslovanie a nadpisy
- pridávať diagramy a príklady, ak pomôžu lepšiemu pochopeniu
- vyhýbať sa nejasným výrazom ako „rýchly“, „intuitívny“ bez jasných metrík

Prehľadnosť a údržba dokumentu

- špecifikáciu treba pravidelne aktualizovať
- je vhodné viesť históriu zmien a verzií
- každý dokument by mal mať zodpovednú osobu, ktorá dohliada na jeho konzistenciu

Spolupráca pri tvorbe dokumentácie

- analytici a architekti – zabezpečujú obsah a presnosť
- projektový manažér – dohliada na súlad s plánom a rozpočtom
- tester – kontroluje, či sa požiadavky dajú overiť
- zákazník – potvrdzuje, že požiadavky vystihujú jeho potreby

Zhrnutie

Dokumentovanie požiadaviek nie je len o písaní textu. Je to proces vytvárania spoločného porozumenia medzi všetkými aktérmi projektu, pričom výsledný dokument sa stáva stabilným základom pre návrh, vývoj aj testovanie systému.

12. PÍSANIE KVALITNÝCH POŽIADAVIEK

Prečo sa zamerať na kvalitu požiadaviek?

Kvalitne napísané požiadavky sú základom úspechu celého projektu. Znižujú riziko nejasností, nedorozumení a chýb pri implementácii. Ak sú požiadavky nepresné alebo nejednoznačné, celý vývoj sa môže uberať nesprávnym smerom, čo vedie k zbytočným nákladom a nespokojnosti zákazníka.

Charakteristiky kvalitnej požiadavky

- úplná – obsahuje všetky potrebné informácie na jej pochopenie
- konzistentná – nie je v rozpore s inými požiadavkami
- jednoznačná – umožňuje len jeden možný výklad

overiteľná – dá sa objektívne skontrolovať, či bola splnená

- realizovateľná – v rámci technických, časových a finančných možností
- sledovateľná – vieme dohľadať jej pôvod a prepojenie na ďalšie časti systému

Tieto vlastnosti sú známe aj pod skratkou „C.U.C.R.O.T.“ (Complete, Unambiguous, Consistent, Realistic, Observable, Traceable).

Jasný štýl písania požiadaviek

- používať jednoduchý, zrozumiteľný jazyk
- vyhýbať sa slangovým alebo príliš technickým pojmom, ak nie sú nevyhnutné
- rozbiť dlhé vety na kratšie a prehľadnejšie
- využívať odrážky a číslovanie tam, kde to pomôže štruktúre
- písať v aktívnom tvare („systém odošle správu“ namiesto „správa bude odoslaná systémom“)

Vyhýbanie sa nejasnostiam

Autori varujú pred vágnymi výrazmi, ktoré každý môže chápať inak. Príklady problémových slov:

- rýchly
- flexibilný
- ľahko použiteľný
- efektívny
- intuitívny

Ak sa takéto pojmy objavajú, je potrebné ich spresniť kvantitatívne (napr. „do 2 sekúnd“, „bez zaškolenia počas 10 minút“).

Používanie štandardizovaných výrazov

- ak sa v organizácii zaužívali štandardné termíny, treba ich používať konzistentne
- glosár pojmov pomáha predchádzať nedorozumeniam
- všetci účastníci projektu by mali mať prístup k rovnakej terminológii

Štruktúra jednej požiadavky

Kvalitná požiadavka by mala obsahovať:

- identifikátor (napr. REQ-23)
- názov alebo stručný popis
- samotný text požiadavky
- zdôvodnenie (prečo je dôležitá)
- akceptačné kritériá (ako overíme, že je splnená)
- prioritu
- stav (napr. koncept, schválená, zamietnutá)

Príklady správnych a nesprávnych požiadaviek

Nesprávna požiadavka:

- „Systém musí byť rýchly.“

Správna požiadavka:

- „Systém musí zobrazíť výsledky vyhľadávania do 2 sekúnd v 95 % prípadov.“

Nesprávna požiadavka:

- „Používateľ bude mať jednoduché rozhranie.“

Správna požiadavka:

„Rozhranie bude obsahovať maximálne 5 ovládacích prvkov na obrazovke a bude prehľadné bez potreby školenia.“

Kontrola kvality požiadaviek

- formálne revízie dokumentu (napr. peer review)
- validácie so zákazníkom alebo budúcimi používateľmi
- použitie kontrolných zoznamov na overenie kvality
- simulácie alebo prototypy na potvrdenie, či požiadavka dáva zmysel

Zhrnutie

Táto kapitola zdôrazňuje, že kvalitné požiadavky nevznikajú náhodou – ich tvorba je disciplinovaný proces, ktorý vyžaduje presné vyjadrovanie, spoločný slovník, kontrolu a validáciu. Iba tak je možné zabezpečiť, že systém splní očakávania a bude vyhovovať svojmu účelu.

13. MODELovanie POŽIADAVIEK

- **Modelovanie požiadaviek**
Grafické modely dopĺňajú text a pomáhajú lepšie pochopiť procesy a dátové toky.
- **Typy modelov**
 - diagramy tokov dát
 - stavové diagramy
 - diagramy aktivít (swimlane diagramy)
 - rozhodovacie stromy a tabuľky
 - dialógové mapy
 - UML diagramy
- **Výber vhodných modelov**
Modelovanie závisí od typu projektu a od toho, čo chceme vysvetliť — napr. na popis tokov dát je vhodný diagram DFD, na správanie systému stavový diagram.
- **Modelovanie v agilnom prostredí**
Aj v agilných projektoch má vizualizácia dôležitú úlohu, ale používa sa v menšom rozsahu a častejšie sa priamo diskutuje v tíme.

14. ŠPECIFIKOVANIE DÁTOVÝCH POŽIADAVIEK

- **Vzťahy v dátach**
Treba pochopiť, ako spolu súvisia jednotlivé dátové prvky (napr. vzťahy 1:N, M:N), čo je dôležité pre návrh databázy.
- **Dátový slovník**
Slúži na presné definovanie názvov, formátov, jednotiek a obmedzení jednotlivých údajov, aby sa predišlo nejasnostiam.



- **Analýza dát**
Zahrňa identifikáciu požadovaných reportov, dotazov, metadát a štruktúr potrebných pre chod systému.
- **Špecifikovanie** reportov
Dôležité je jasne popísať:
 - čo má report zobrazovať
 - aké má mať filtre
 - aké údaje sa budú vypočítavať
- **Dashboardy a vizualizácia dát**
Pri modernejších systémoch je potrebné definovať aj požiadavky na interaktívne dashboardy, aby podporili rozhodovanie.

15. NAD RÁMEC FUNKCIONALITY

- **Kvalitatívne atribúty**
Tzv. nefunkčné požiadavky (napr. výkon, spoľahlivosť, použiteľnosť, bezpečnosť) bývajú často zanedbávané, pritom sú pre spokojnosť zákazníka kľúčové.
- **Externé vs. interné atribúty**
 - externé: vníma ich používateľ (napr. rýchlosť odozvy)
 - interné: týkajú sa kódu či architektúry (napr. modulárnosť, čitateľnosť)
- **Definovanie** kvality
Odporúča sa kvantifikovať kvalitatívne požiadavky, napríklad „systém spracuje transakciu do 2 sekúnd pre 95 % prípadov“.
- **Konflikty medzi atribútmi**
Niektoré kvalitatívne požiadavky sa môžu navzájom ovplyvňovať (napr. bezpečnosť môže znižovať použiteľnosť) a treba ich vyvážiť.

16. ZNIŽOVANIE RIZIKA PROTOTYPOVANÍM

- **Čo je prototypovanie?**
Prototyp je čiastočne funkčný model systému, ktorý umožňuje testovať myšlienky pred finálnym vývojom.
- **Typy prototypov**
 - *vyhoditeľné* (throwaway): rýchly návrh, ktorý sa zahodí
 - *evolučné*: postupne sa vyvíjajú až do hotového riešenia
- **Výhody prototypov**
 - lepšie pochopenie požiadaviek



- odhalenie chýb a nejasností
 - zapojenie používateľov do hodnotenia riešenia
- **Riziká prototypov**
 - ilúzia, že prototyp je „takmer hotový“
 - zbytočné investovanie času do detailov
 - nesprávne očakávania ohľadom výkonu
- **Úspech prototypovania**

Kľúčové je definovať cieľ prototypu a komunikovať, že nejde o finálny produkt.

17. STANOVENIE PRIORÍT POŽIADAVIEK

- **Prečo prioritizovať?**
 - zdroje sú obmedzené
 - nie všetky požiadavky sú rovnako dôležité
 - pomáha plánovať iterácie alebo fázy
- **Techniky prioritizácie**
 - MoSCoW (Must, Should, Could, Won't)
 - párové porovnávanie
 - stupnice dôležitosti
 - „100 dolárová hra“ (rozdeľovanie virtuálneho rozpočtu medzi požiadavky)
- **Hry okolo priorít**

Niekedy sa účastníci snažia všetko označiť ako „musí byť“, preto je dôležité facilitovať diskusiu a trvať na racionálnych kompromisoch.
- **Prioritizácia v agilných projektoch**

V agilnom prístupe sa priorita často prehodnocuje pred každým sprintom na základe aktuálnych potrieb zákazníka.

18. VALIDÁCIA POŽIADAVIEK

Čo znamená validácia požiadaviek?

Validácia požiadaviek znamená overenie, či sú správne, úplné a skutočne vystihujú potreby zákazníka a používateľov. Cieľom validácie je zistiť, či sa buduje *správny* systém (správne pochopené potreby), nie len či sa buduje systém *správne* (to už rieši verifikácia).

Validácia pomáha predísť chybám ešte predtým, než sa začne implementácia, pretože odstrániť chybu v požiadavkách neskôr je oveľa nákladnejšie.

Ciele validácie

- zabezpečiť, že požiadavky sú správne pochopené
- potvrdiť, že sú v súlade s obchodnými cieľmi
- odhaliť nedostatky alebo protirečenia
- získať súhlas všetkých zainteresovaných strán
- minimalizovať riziko, že sa bude budovať systém, ktorý nikto nepotrebuje

Metódy validácie

Autori odporúčajú kombináciu viacerých techník:

- revízie (reviews) – systematické čítanie a posudzovanie požiadaviek s odborníkmi
- inšpekcie – formálnejší druh revízie, často vedený vyškoleným moderátorom
- simulácie – použitie prípadov použitia alebo diagramov na prechádzanie scenárov
- prototypy – vizuálne modely alebo časti funkčného systému, ktoré sa dajú overiť s používateľmi
- testovacie scenáre – navrhovanie testov už počas fázy požiadaviek, aby sa ukázalo, či sú overiteľné

Úloha zainteresovaných strán

- zákazníci a používateľskí zástupcovia musia požiadavky posúdiť a potvrdiť
- analytici a architekti kontrolujú ich konzistenciu a realizovateľnosť
- tester zabezpečí, že požiadavky sa dajú overiť v praxi
- projektový manažér dohliada, aby validácia prebehla včas a riadne

Akceptačné kritériá

Každá požiadavka by mala mať definované akceptačné kritériá, podľa ktorých sa bude merať jej úspešné splnenie.

Príklad:

- „Systém odošle potvrdenie do 2 sekúnd pre 95 % požiadaviek“
- „Používateľ dokáže vytvoriť nový účet bez školenia do 5 minút“

Takéto kritériá sú základom pre plánovanie testov a podporujú spoločné porozumenie.

Proces validácie

Bežný validančný proces môže vyzeráť takto:

1. príprava – zozbierať a distribuovať požiadavky účastníkom validácie
2. individuálne preskúmanie – každý účastník si materiály prečíta
3. stretnutie (review meeting) – spoločná diskusia, zachytenie nedostatkov
4. opravy – zapracovanie zmien
5. schválenie – formálne potvrdenie

Najčastejšie zistené problémy

Počas validácie sa často odhalia:

- duplicity v požiadavkách
- konflikty medzi rôznymi požiadavkami
- nesprávne alebo nereálne predpoklady
- nedostatočné definovanie okrajových podmienok
- príliš všeobecné alebo neoveriteľné formulácie

Prínosy validácie

- zabráni sa budovaniu nesprávneho riešenia
- zvýši sa dôvera medzi zákazníkom a tímom
- podporí sa spolupráca všetkých zainteresovaných strán
- znížia sa náklady na zmeny v neskorších fázach

Zhrnutie

Validácia požiadaviek je kritický krok, ktorý pomáha odhaliť chyby ešte predtým, ako sa začne implementácia. Kvalitná validácia zabezpečí, že systém bude naozaj plniť potreby používateľov a prinesie očakávanú hodnotu.

19. OPÄTOVNÉ POUŽITIE POŽIADAVIEK

- **Prečo znovu použiť požiadavky?**
Ušetrí sa čas, zníži riziko chýb, zvýši sa konzistentnosť medzi projektmi.
- **Formy opätovného použitia**
 - celé funkcie
 - vzory požiadaviek
 - existujúce špecifikácie
- **Prekážky**
 - nedostatočne zdokumentované minulé riešenia
 - zmena technológií
 - odlišné domény
- **Faktory úspechu**
 - kvalitná databáza požiadaviek
 - štandardizované šablóny
 - školenie tímu, ako s požiadavkami pracovať

20. NAD RÁMEC VÝVOJA POŽIADAVIEK

- **Prepojenie požiadaviek s projektovým plánom**
Na základe požiadaviek sa odhaduje náročnosť, tvorí sa harmonogram, pridelujú sa zdroje.
- **Odhadovanie námahy**
Autori odporúčajú použiť metriku (napr. funkčné body) alebo skúsenosti z minulých projektov.
- **Prechod od požiadaviek k návrhu**
Požiadavky sú východiskom pre architektúru a návrh systému.



- **Sledovanie požiadaviek**
Požiadavky by mali byť prepojené na implementačné úlohy, testy aj dokumentáciu.

21. AGILNÉ PROJEKTY

- **Limitácie vodopádového modelu**
Klasický prístup je často príliš rigidný, v agilných tímoch sa preto kladie dôraz na flexibilitu a inkrementálny vývoj.
- **Agilný prístup k požiadavkám**
 - menej dokumentácie, viac komunikácie
 - požiadavky sa rozvíjajú priebežne
 - úzka spolupráca so zákazníkom
- **Backlog a prioritizácia**
Produktový backlog je živý zoznam požiadaviek, ktorý sa pred každým sprintom znovu reviduje.
- **Epiky, user stories, features**
 - *epic*: veľký celok, ktorý sa rozdelí na menšie
 - *user story*: krátky opis potreby používateľa
 - *feature*: skupina user stories
- **Prijímanie zmien**
Zmeny sú očakávané a vítané, systém sa priebežne prispôsobuje.

22. PROJEKTY ROZVOJA A NAHRADENIA

- **Charakteristika**
Pri projektoch, ktoré rozširujú alebo nahrádzajú existujúci systém, je nutné pochopiť aktuálne fungovanie a jeho obmedzenia.
- **Výzvy**
 - dokumentácia pôvodného systému môže chýbať
 - používatelia môžu byť zvyknutí na starý systém
 - zachovanie výkonnosti počas migrácie
- **Techniky**
 - analýza existujúcej funkcionality
 - spätné inžinierstvo
 - prioritizácia podľa obchodných cieľov
- **Podpora prijatia nového riešenia**

Je



potrebné zapojiť používateľov už v úvode, aby akceptovali zmenu a vedeli sa s ňou stotožniť.

23. PROJEKTY S KRABICOVÉHO SW

- **Výber balíkového softvéru**
 - identifikovať požiadavky, ktoré sú kľúčové
 - overiť, či balík podporuje potrebné obchodné pravidlá
 - zvážiť možnosti prispôsobenia
- **Požiadavky na implementáciu**
 - konfigurácia (nastavenia v rámci balíka)
 - integrácia s inými systémami
 - rozšírenia a úpravy
- **Dáta a procesy**

Migračné požiadavky na údaje a zladenie s aktuálnymi procesmi organizácie sú kľúčové.
- **Výzvy**

Komerčné balíky nemusia dokonale zapadnúť do špecifických potrieb, čo vyžaduje kompromisy.

24. OUTSOURCING PROJEKTOV

- **Špecifiká outsourcingu**
 - dodávateľ často sídli v inej krajine
 - kultúrne a jazykové rozdiely
 - vyššie nároky na presnosť požiadaviek
- **Dohoda o úrovni služieb (SLA)**

Potrebné je definovať presné merateľné kritériá, aby sa minimalizovali nejasnosti a spory.
- **Spolupráca obstarávateľ–dodávateľ**

Komunikácia musí byť pravidelná a transparentná, s jasne definovanými rolami a zodpovednosťami.
- **Zmena požiadaviek**

V outsourcingu sa každá zmena môže výrazne predražiť, preto je dôležité čo najpresnejšie špecifikovať rozsah na začiatku.

25. PROJEKTY AUTOMATIZÁCIE OBCHODNÝCH PROCESOV

- **Modelovanie procesov**
Pred automatizáciou je nutné pochopiť aktuálne obchodné procesy, často sa kreslia BPMN diagramy alebo procesné mapy.
- **Odvodenie požiadaviek**
Požiadavky by mali vychádzať z cieľa optimalizácie alebo zrýchlenia procesov, nie len „nakresliť to isté v softvéri“.
- **Definovanie budúceho stavu**
Najprv sa má popísať budúci (ideálny) stav procesu, až potom sa buduje IT riešenie.
- **Meranie výkonnosti**
Zároveň je vhodné nastaviť KPI (kľúčové ukazovatele výkonnosti), aby sa dalo merať zlepšenie po implementácii.

26. PROJEKTY BUSINESS ANALYTIKY

- **Charakteristika**
Business analytické projekty sa zameriavajú na spracovanie a interpretáciu dát pre podporu rozhodovania.
- **Výzvy**
 - požiadavky sa môžu meniť dynamicky podľa zistení
 - zložité dátové štruktúry
 - integrácia s viacerými zdrojmi dát
- **Kľúčové požiadavky**
 - definovanie metrík a ukazovateľov
 - špecifikovanie transformácií a analýz
 - pochopenie rozhodovacích procesov používateľov
- **Iteratívny prístup**
Business analytické projekty sa často vyvíjajú inkrementálne, pričom sa požiadavky doladujú na základe postupných výsledkov.

27. PROJEKTY VSTAVANÝCH A REÁLNYCH ČASOVÝCH SYSTÉMOV

- **Charakteristika**
Embedded (vstavané) systémy sú kombináciou hardvéru a softvéru, často s prísnyimi časovými obmedzeniami.



- **Špecifiká požiadaviek**
 - presná definícia reakčných časov
 - definovanie interakcie s hardvérom
 - vysoká spoľahlivosť a bezpečnosť
- **Modelovanie**

Používajú sa diagramy stavov, sekvenčné diagramy, prípadne architektonické náčrty.
- **Výzvy**
 - komplexné testovanie
 - vysoké nároky na kvalitu a odolnosť
 - obmedzené zdroje (pamäť, výkon)

28. PRAKTIKY RIADENIA POŽIADAVIEK

- **Riadenie požiadaviek**

Znamená sledovať, meniť a udržiavať požiadavky počas celého životného cyklu projektu.
- **Hlavné úlohy**
 - verzovanie požiadaviek
 - sledovanie stavu a zmien
 - riešenie konfliktov
 - udržiavanie väzieb na testy a kód
- **Atribúty požiadaviek**

Každá požiadavka by mala mať identifikátor, prioritu, stav, zodpovednú osobu a odkaz na zdroje.
- **Riadenie v agilnom prostredí**

Aj v agilných tímoch sa riadenie požiadaviek používa, len vo viac neformálnej a flexibilnej podobe (napr. product backlog).

29. ZMENY

- **Prečo riadiť zmeny?**

Zmeny sú nevyhnutné, ale nekontrolované zmeny vedú k chaosu a prečerpaniu rozpočtu.
- **Politika riadenia zmien**

Treba mať jasný proces, ako sa budú požiadavky meniť, kto má schvaľovacie právomoci a ako sa zmeny dokumentujú.



- **Change Control Board (CCB)**
Skupina, ktorá posudzuje a schvaľuje zmeny.
- **Analýza dopadov**
Každá navrhnutá zmena musí byť vyhodnotená z pohľadu dopadu na rozpočet, čas a kvalitu.
- **Zmeny v agilných projektoch**
V agilnom svete sa zmeny prijímajú priebežne a sú prirodzenou súčasťou iteratívneho vývoja.

30. VÄZBY V REŤAZCI POŽIADAVIEK

- **Prečo sledovať väzby?**
Požiadavky sú navzájom previazané — funkčné, nefunkčné, obchodné ciele, testy a implementácia. Ak sa jedna zmení, ovplyvní to ďalšie.
- **Trasovateľnosť**
Znamená schopnosť spätne dohľadať, prečo a ako bola požiadavka vytvorená, a naopak — z požiadavky určiť, kde je implementovaná.
- **Výhody trasovateľnosti**
 - jednoduchšie zmenové riadenie
 - lepšie plánovanie testov
 - vyššia kvalita dokumentácie

31. NÁSTROJE NA INŽINIERSTVO POŽIADAVIEK

- **Druhy nástrojov**
 - nástroje na správu požiadaviek (napr. DOORS, JIRA)
 - modelovacie nástroje (napr. Enterprise Architect, Visio)
 - nástroje na spoluprácu (napr. Confluence)
- **Výber nástroja**
Závisí od veľkosti projektu, počtu zainteresovaných strán a zložitosti požiadaviek.
- **Integrácia**
Ideálne je, ak nástroje vedia komunikovať so systémami na testovanie a správu verzií, aby sa zabránilo duplicitám.
- **Nevýhody**
Príliš zložitý nástroj môže odradiť tím — dôležitá je rovnováha medzi funkcionalitou a jednoduchosťou používania.

32. ZLEPŠOVANIE PROCESOV POŽIADAVIEK

- **Prečo zlepšovať?**
Lepší proces požiadaviek znižuje riziko zlyhania a zlepšuje spokojnosť zákazníka.
- **Prístupy**
 - hodnotenie aktuálnych procesov
 - definovanie cieľového stavu
 - priebežné iteratívne zlepšovanie
- **Modely** **zrelosti**
Např. CMMI alebo SPICE pomáhajú organizáciám určiť úroveň svojich procesov a plánovať ich zlepšenie.
- **Odporúčania**
Začať s malými, postupnými zmenami a zapojiť celý tím do budovania lepších praktík.

33. POŽIADAVKY A RIADENIE RIZÍK

- **Riziko v požiadavkách**
Nedostatočne pochopené alebo neoverené požiadavky predstavujú jedno z najväčších rizík v projektoch.
- **Techniky riadenia rizík**
 - analýza pravdepodobnosti a dopadu
 - identifikácia rizikových oblastí
 - plánovanie záloh a mitigácií
- **Prepojenie s požiadavkami**
Riziká by sa mali monitorovať priebežne a priamo súvisia s kvalitou a stabilitou požiadaviek.
- **Kultúra prevencie**
Tím by mal byť motivovaný otvorene komunikovať o rizikách a aktívne ich riešiť už vo fáze požiadaviek.

ZOZNAM ZDROJOV

- [1] Wiegers K., Beatty J. , (2013). Software Requirements (Developer Best Practices) 3rd Edition, Microsoft Press, p. 672, ISBN: 9780735679665.
- [2] Bonnema G. M. , Broenink J. F. , Veenvliet K. T. , (2015). Systems Design and Engineering 1st Edition, CRC Press, p. 132, ISBN: 9781498751261.