

# Kapitola 4 – Špecifikácia požiadaviek (Inžinierstvo požiadaviek)

# Inžinierstvo s požiadavkami



- ✧ Proces vytvárania **služieb**, ktoré zákazník od systému vyžaduje, a **obmedzenia**, za ktorých systém funguje a je vyvíjaný.
- ✧ Systémové požiadavky sú popisy systémových služieb a obmedzení, ktoré sú generované počas procesu navrhovania požiadaviek.

# Čo je to požiadavka ?



- ✧ Môže siahať od abstraktného vyjadrenia služby alebo systémového obmedzenia na vysokej úrovni až po podrobnú matematickú formálnu špecifikáciu.
- ✧ Požiadavky môžu plniť dvojakú funkciu. Obe tieto vyhlásenia možno nazvať požiadavkami:
  - Môže byť základom pre ponuku na zákazku – preto musí byť otvorená pre výklad
  - Môže byť základom pre samotnú zmluvu – preto musí byť podrobne definovaná

# Abstrakcia požiadaviek (Davis)



„Ak chce firma uzavrieť zmluvu na veľký projekt vývoja softvéru, musí svoje potreby definovať dostatočne abstraktným spôsobom, aby riešenie nebolo vopred definované. Požiadavky musia byť napísané tak, aby sa o zákazku mohli uchádzať viacerí dodávateľia, ktorí možno ponúkajú rôzne spôsoby uspokojenia potrieb klientskej organizácie. Po zadaní zákazky musí dodávateľ napísať pre klienta definíciu systému podrobnejšie, aby klient pochopil a mohol overiť, čo softvér urobí. Oba tieto dokumenty možno nazvať *dokumentom požiadaviek na systém*.“

# Druhy požiadaviek



## ✧ Používateľské požiadavky

- Príkazy v prirodzenom jazyku plus diagramy služieb, ktoré systém poskytuje, a jeho prevádzkové obmedzenia. Napísané pre zákazníkov.

## ✧ Systémové požiadavky

- Štruktúrovaný dokument s podrobným popisom funkcií, služieb a prevádzkových obmedzení systému. Definuje, čo sa má realizovať, takže môže byť súčasťou zmluvy medzi klientom a dodávateľom.

# Používateľské a systémové požiadavky



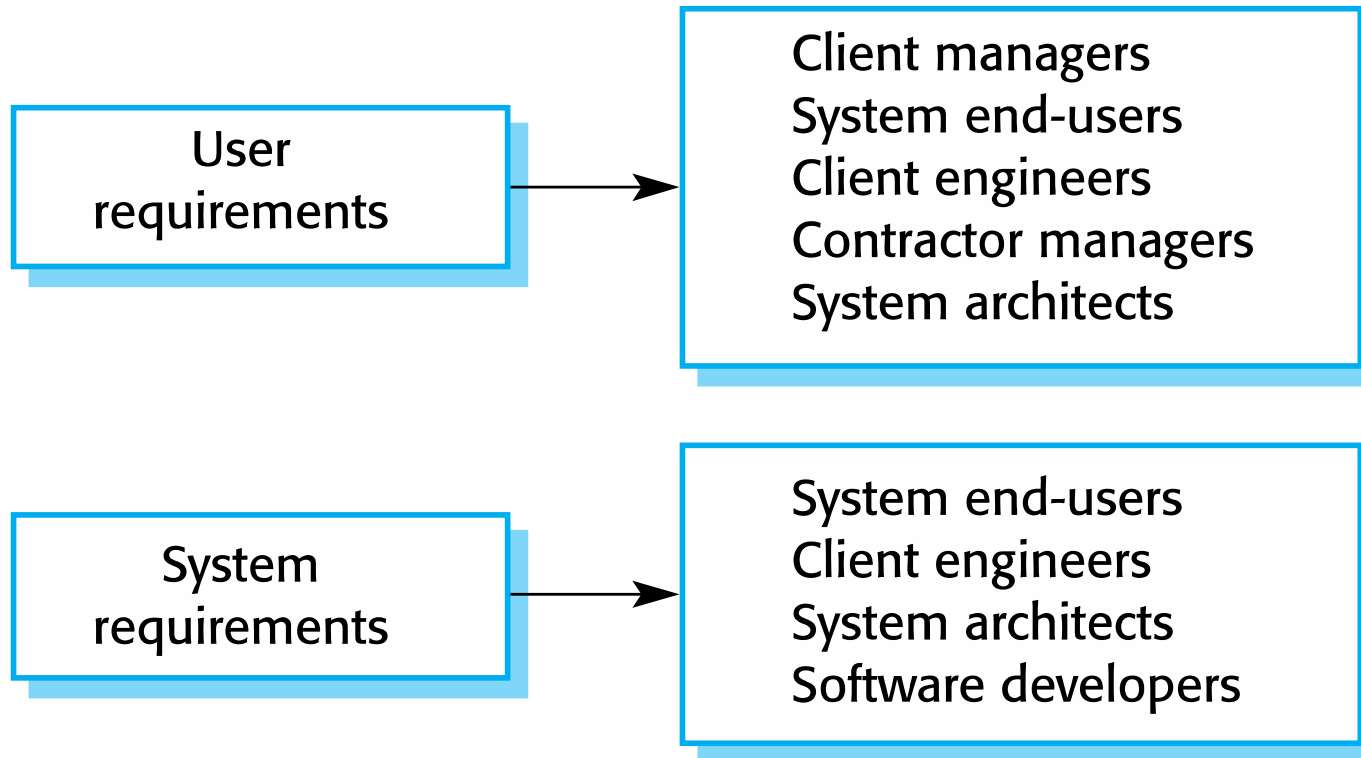
## User requirements definition

- 1.** The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

## System requirements specification

- 1.1** On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- 1.2** The system shall generate the report for printing after 17.30 on the last working day of the month.
- 1.3** A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
- 1.4** If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
- 1.5** Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

# Kto využíva tieto typy požiadaviek



# Zainteresované strany systému

---



- ✧ Každá osoba alebo organizácia, ktorá je nejakým spôsobom ovplyvnená systémom a má teda oprávnený záujem
- ✧ Typy zainteresovaných strán
  - Koncoví používatelia
  - Systémoví manažéri
  - Majitelia systému
  - Externé zainteresované strany



# Zainteresované strany v systéme Mentcare



- ✧ Pacienti, ktorých informácie sú zaznamenané v systéme.
- ✧ Lekári, ktorí sú zodpovední za hodnotenie a liečbu pacientov.
- ✧ Sestry, ktoré koordinujú konzultácie s lekármi a podávajú niektoré liečby.
- ✧ Recepčné, ktoré riadia návštevy pacientov.
- ✧ IT pracovníci, ktorí sú zodpovední za inštaláciu a údržbu systému.

## Zainteresované strany v systéme Mentcare (2)



- ✧ Manažér lekárskej etiky, ktorý musí zabezpečiť, aby systém spĺňal aktuálne etické smernice pre starostlivosť o pacienta.
- ✧ Manažéri zdravotnej starostlivosti, ktorí získavajú manažérske informácie zo systému.
- ✧ Pracovníci zdravotnej dokumentácie, ktorí sú zodpovední za zabezpečenie toho, aby sa informácie o systéme mohli udržiavať a uchovávať, a že postupy vedenia záznamov boli správne implementované.

# Agilné metódy a požiadavky



- ✧ Mnohé agilné metódy tvrdia, že vytváranie podrobných systémových požiadaviek je stratou času, keďže požiadavky sa menia tak rýchlo.
- ✧ Dokument s požiadavkami je preto vždy neaktuálny.
- ✧ Agilné metódy zvyčajne využívajú inžinierstvo prírastkových požiadaviek a môžu vyjadrovať požiadavky ako „*príbehy používateľov*“
- ✧ To je praktické pre obchodné systémy, ale problematické pre systémy, ktoré vyžadujú analýzu pred dodaním (napr. kritické systémy) alebo systémy vyvinuté niekoľkými tímami.

# Funkcionálne a nefunkcionálne požiadavky



## ✧ Funkcionálne požiadavky

- Výpisy služieb, ktoré má systém poskytovať, ako má systém reagovať na konkrétne vstupy a ako sa má správať v konkrétnych situáciách .
- Môžu uviesť, čo by systém nemal robiť.

## ✧ Nefunkcionálne požiadavky

- Obmedzenia služieb alebo funkcií ponúkaných systémom, ako sú časové obmedzenia, obmedzenia procesu vývoja, štandardy atď .
- Často sa vzťahujú na systém ako celok a nie na jednotlivé funkcie alebo služby.

## ✧ Doménové požiadavky

- Služby a obmedzenia systému z oblasti prevádzky

# Funkcionálne požiadavky



- ✧ Popisujú funkčnosť alebo systémové služby.
- ✧ Závisia od typu softvéru, očakávaných používateľov a typu systému, kde sa softvér používa.
- ✧ Funkcionálne požiadavky používateľov môžu predstavovať abstraktné vyhlásenia o tom, **čo by mal systém robiť**.
- ✧ Funkcionálne systémové požiadavky by mali **podrobne popisovať systémové služby**.

# Mentcare systém: funkcionálne požiadavky



- ✧ Používateľ musí mať možnosť vyhľadávať v zoznamoch schôdzok pre všetky kliniky.
- ✧ Systém vygeneruje každý deň pre každú kliniku zoznam pacientov, u ktorých sa očakáva, že sa v daný deň dostavia na schôdzku.
- ✧ Každý zamestnanec používajúci systém musí byť jednoznačne identifikovaný svojím 8-miestnym číslom zamestnanca.

# Nepresnosť



- ✧ Problémy vznikajú, keď nie sú presne stanovené funkčné požiadavky.
- ✧ Nejednoznačné požiadavky môžu vývojári a používatelia interpretovať rôznymi spôsobmi.
- ✧ Zvážte výraz „hľadať“ v požiadavke č.1
  - Zámer používateľa – vyhľadať meno pacienta na všetkých stretnutiach vo všetkých ambulanciách;
  - Výklad vývojára – hľadanie mena pacienta v jednotlivej ambulancii. Používateľ si vyberie kliniku a potom vyhľadáva.

# Požiadavky na úplnosť a konzistentnosť



✧ V zásade by požiadavky mali byť úplné a konzistentné.

## ✧ Úplné

- Mali by obsahovať popisy všetkých požadovaných zariadení.

## ✧ Konzistentné

- V popisoch zariadení systému by nemali byť žiadne konflikty alebo rozpory.

✧ V praxi je z dôvodu systémovej a environmentálnej zložitosti nemožné vytvoriť úplný a konzistentný dokument požiadaviek.



# Nefunkcionálne požiadavky



- ✧ Tieto požiadavky definujú vlastnosti systému a obmedzenia, napr. spoľahlivosť, čas odozvy a požiadavky na úložisko. Obmedzeniami sú schopnosť I/O zariadenia, reprezentácie systému atď.
- ✧ Aký by mal systém byť.
- ✧ Môžu byť špecifikované aj procesné požiadavky, ktoré vyžadujú konkrétne IDE, programovací jazyk alebo metódu vývoja.
- ✧ **Nefunkcionálne požiadavky môžu byť dôležitejšie** ako funkčné požiadavky. Ak tieto nie sú splnené, systém môže byť zbytočný.

# Klasifikácia nefunkcionálnych požiadaviek



## ✧ Požiadavky na produkt

- Požiadavky, ktoré špecifikujú, že dodaný produkt sa musí správať určitým spôsobom, napr. rýchlosť vyhotovenia, spoľahlivosť atď.

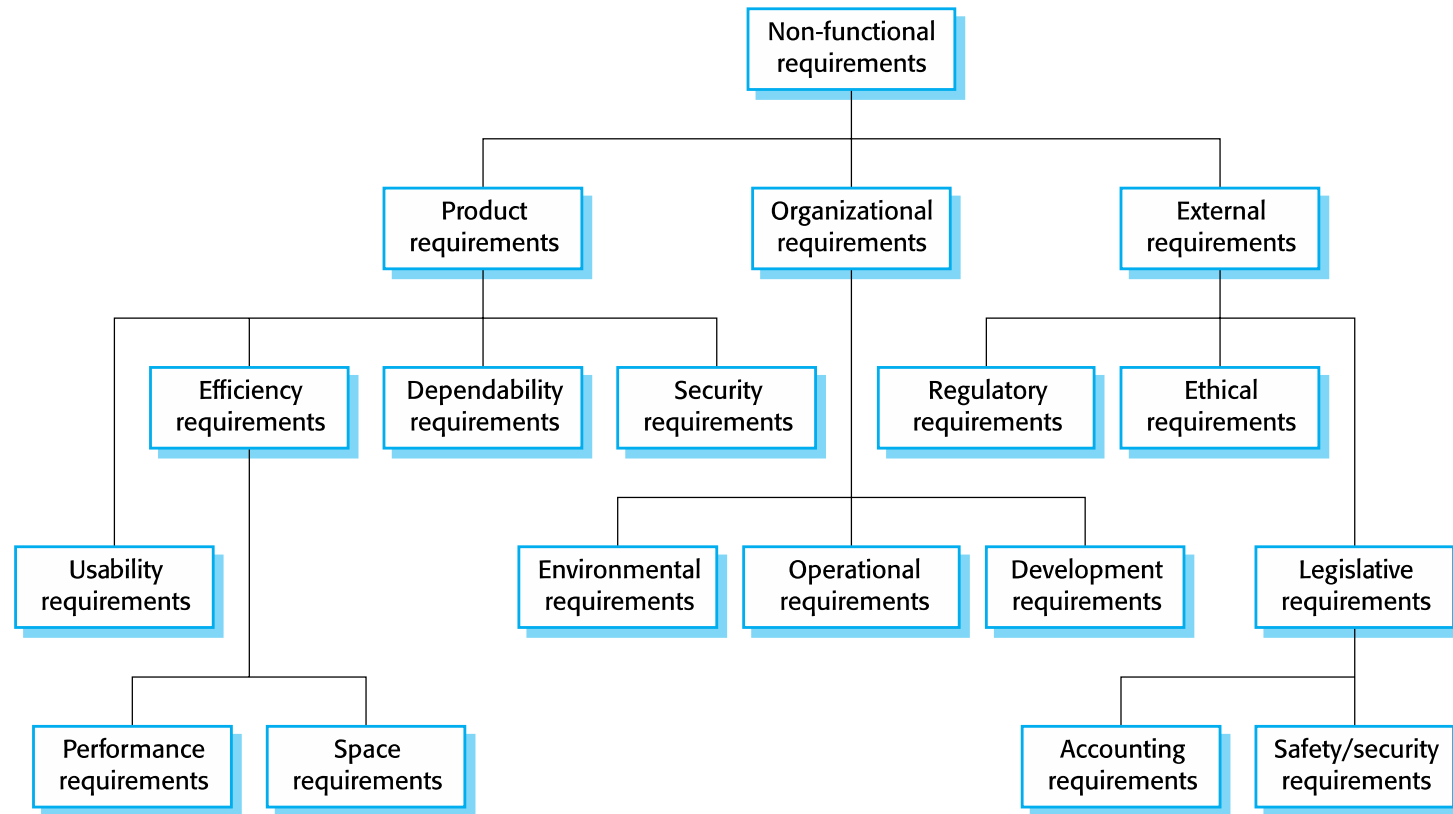
## ✧ Organizačné požiadavky

- Požiadavky, ktoré sú dôsledkom organizačných politík a postupov, napr. používané procesné štandardy, implementačné požiadavky atď.

## ✧ Externé požiadavky

- Požiadavky, ktoré vyplývajú z faktorov, ktoré sú externé pre systém a proces jeho vývoja, napr. požiadavky na interoperabilitu, legislatívne požiadavky atď.

# Nefunkcionálne požiadavky



# Implementácia nefunkcionálnych požiadaviek



- ✧ Nefunkcionálne požiadavky môžu ovplyvniť skôr celkovú architektúru systému ako jednotlivé komponenty.
  - Napríklad, aby ste sa uistili, že sú splnené požiadavky na výkon, možno budete musieť zorganizovať systém tak, aby sa minimalizovala komunikácia medzi komponentmi.
- ✧ Jedna nefunkcionálna požiadavka, ako je požiadavka na bezpečnosť, môže generovať množstvo súvisiacich funkcionálnych požiadaviek, ktoré definujú systémové služby, ktoré sú požadované.
  - Môže tiež vytvárať požiadavky, ktoré obmedzujú existujúce požiadavky.

# Príklady nefunkcionálnych požiadaviek v systéme Mentcare



## Požiadavka na produkt

Systém Mentcare bude dostupný pre všetky kliniky počas bežnej pracovnej doby (Po – Pi, 8:30 – 17:30). Prestoje v rámci bežného pracovného času nesmú presiahnuť päť sekúnd za jeden deň.

## Organizačná požiadavka

Používatelia systému Mentcare sa autentifikujú preukazom totožnosti zdravotníckeho úradu.

## Externá požiadavka

Systém bude implementovať ustanovenia o ochrane súkromia pacienta, ako je uvedené v HStan-03-2006-priv.

# Ciele a požiadavky



✧ Nefunkcionálne požiadavky môže byť veľmi ťažké presne stanoviť a nepresné požiadavky môže byť ťažké overiť.

## ✧ Cieľ

- Všeobecný zámer používateľa, akým je jednoduchosť používania.

## ✧ Merateľná a overiteľná nefunkcionálna požiadavka

- Výrok využívajúci nejakú mieru, ktorú možno objektívne testovať (merať).

✧ Ciele sú užitočné pre vývojárov, pretože vyjadrujú zámery používateľov systému.

# Požiadavky na použiteľnosť



- ✧ Systém by mal byť ľahko použiteľný pre zdravotnícky personál a mal by byť organizovaný tak, aby sa minimalizovali chyby používateľa. (**Cieľ**)
- ✧ Zdravotnícky personál musí byť schopný používať všetky funkcie systému po štyroch hodinách školenia. Po tomto školení priemerný počet chýb, ktoré urobia skúsení používatelia, nepresiahne dve chyby za hodinu používania systému. (**Merateľná a overiteľná nefunkcionálna požiadavka**)

# Metriky na špecifikovanie nefunkcionálnych požiadaviek



Cieľ	Miery
Rýchlosť	Spracované transakcie za sekundu Čas odozvy používateľa/udalosti Čas obnovenia obrazovky
Veľkosť	MB Počet čipov ROM
Jednoduchosť použitia	Čas na tréning Počet pomocných rámcov
Spoľahlivosť	Priemerný čas do zlyhania Pravdepodobnosť nedostupnosti Miera výskytu porúch Dostupnosť
Robustnosť	Čas na reštart po zlyhaní Percento udalostí, ktoré spôsobili zlyhanie Pravdepodobnosť poškodenia údajov pri zlyhaní
Prenosnosť	Percento cieľových závislých vyhlásení Počet cieľových systémov

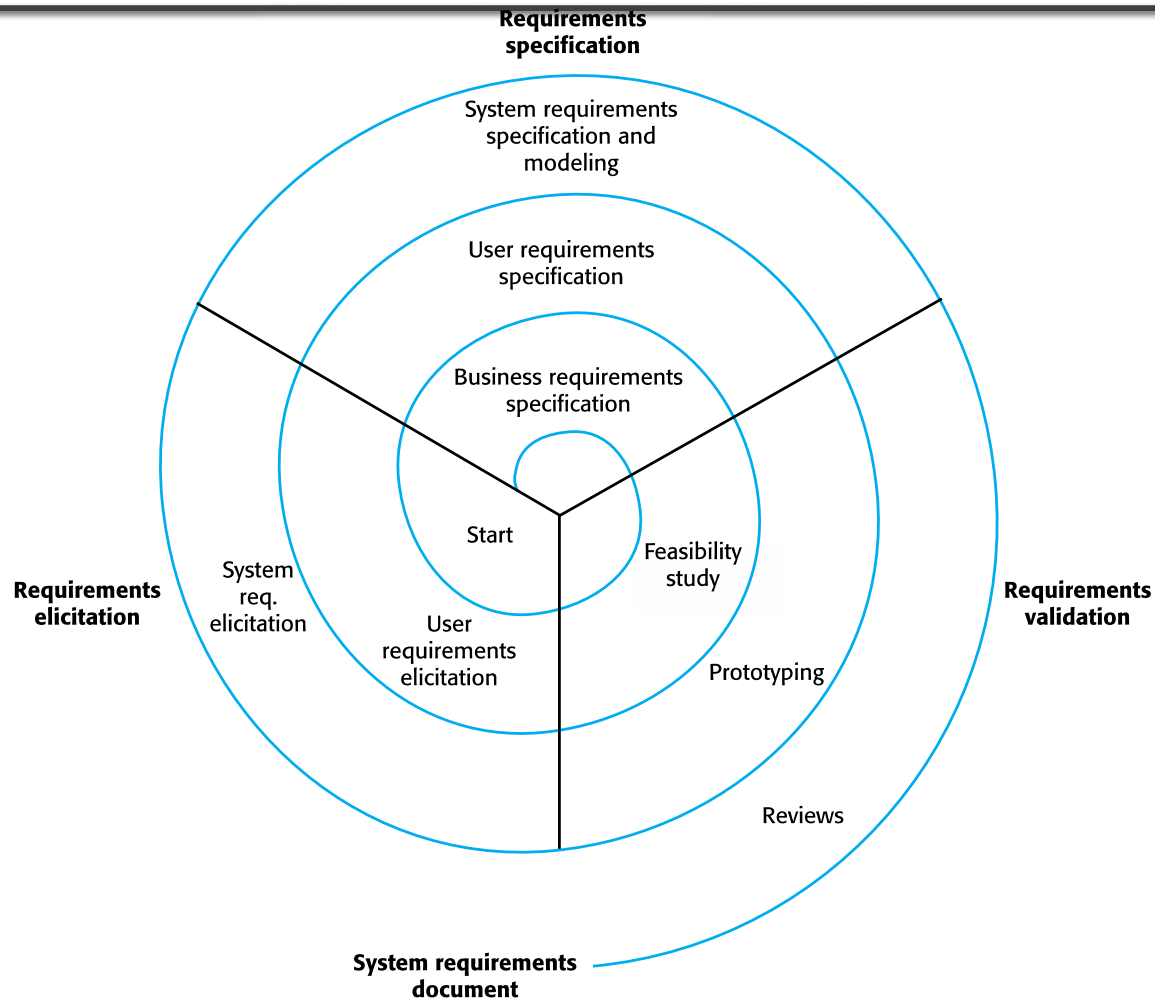


# Inžiniersky proces na špecifikáciu požiadaviek



- ✧ Procesy používané pre špecifikáciu sa značne líšia v závislosti od aplikačnej domény, zainteresovaných ľudí a organizácií.
- ✧ Existuje však množstvo všeobecných činností spoločných pre všetky procesy
  - Získavanie požiadaviek;
  - Analýza požiadaviek;
  - Validácia požiadaviek;
  - Riadenie požiadaviek.
- ✧ V praxi je špecifikácia iteratívna činnosť, v ktorej sa tieto procesy prelínajú.

# Špirálový model na proces inžinierstva požiadaviek (špecifikácie)



# Získavanie a analýza požiadaviek



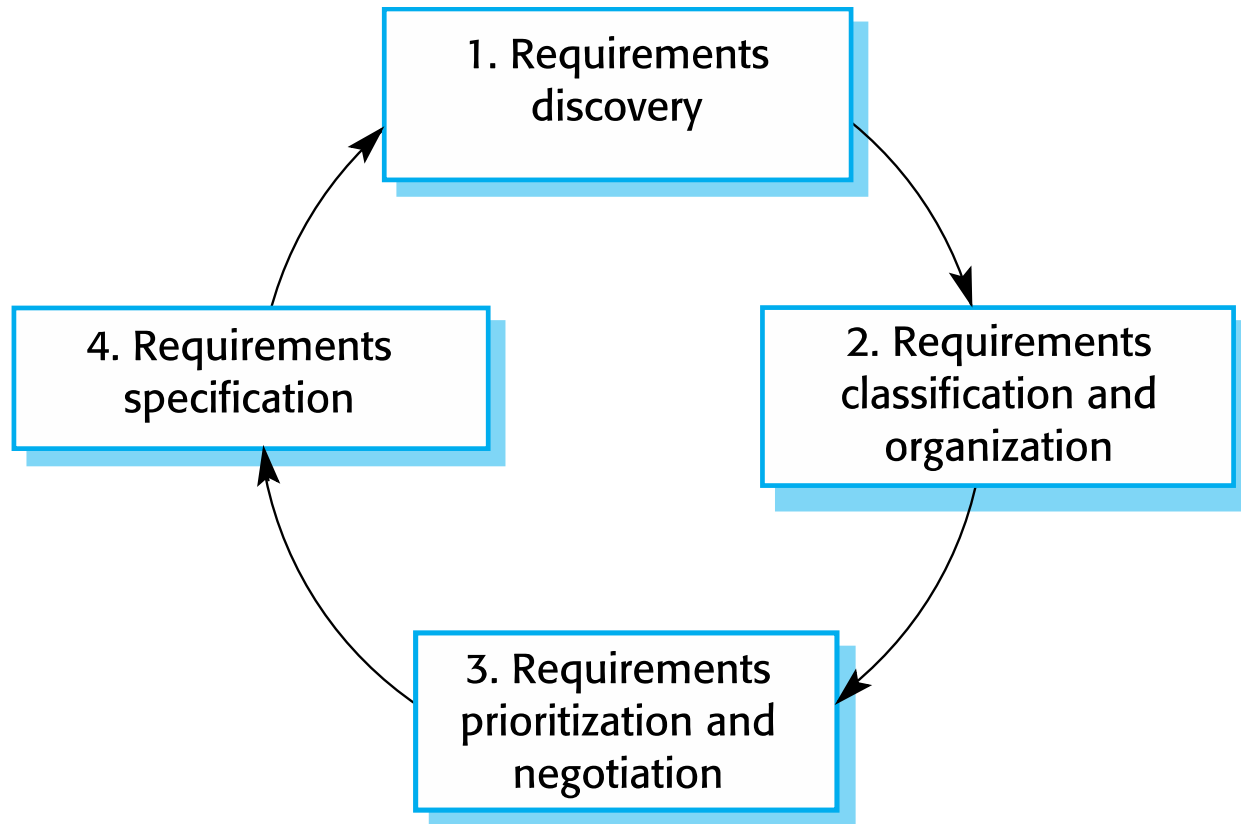
- ✧ Niekedy sa nazýva získavanie požiadaviek alebo zisťovanie požiadaviek.
- ✧ Zahŕňa technický personál pracujúci so zákazníkmi s cieľom zistiť oblasť aplikácie, služby, ktoré by mal systém poskytovať, a prevádzkové obmedzenia systému.
- ✧ Môže zahŕňať koncových používateľov, manažérov, inžinierov zapojených do údržby, doménových expertov, odborové zväzy, atď .

# Získavanie požiadaviek



- ✧ Softwaroví inžinieri spolupracujú s celým radom zainteresovaných strán, aby zistili aplikačnú doménu, služby, ktoré by mal systém poskytovať, požadovaný výkon systému, hardvérové obmedzenia, iné systémy atď.
- ✧ Etapy zahŕňajú:
  - Zisťovanie požiadaviek,
  - Klasifikácia a organizácia požiadaviek,
  - Stanovenie priorít a vyjednávanie požiadaviek,
  - Špecifikácia požiadaviek.

# Proces získavania a analýzy požiadaviek





## ✧ Zisťovanie požiadaviek

- Interakcia so zainteresovanými stranami s cieľom zistiť ich požiadavky. V tejto fáze sa zisťujú aj požiadavky na doménu.

## ✧ Klasifikácia a organizácia požiadaviek

- Zoskupuje súvisiace požiadavky a organizuje ich do koherentných zhlukov.

## ✧ Stanovenie priorít a vyjednávanie

- Uprednostňovanie požiadaviek a riešenie konfliktov požiadaviek.

## ✧ Špecifikácia požiadaviek

- Požiadavky sú zdokumentované a vložené do ďalšieho kola špirály.

# Problémy získavania požiadaviek



- ✧ Zainteresované strany nevedia, čo vlastne chcú.
- ✧ Zainteresované strany vyjadrujú požiadavky vo svojich vlastných podmienkach.
- ✧ Rôzne zainteresované strany môžu mať protichodné požiadavky.
- ✧ Organizačné a politické faktory môžu ovplyvniť systémové požiadavky.
- ✧ Požiadavky sa počas procesu analýzy menia. Môžu sa objaviť noví zainteresovaní a podnikateľské prostredie sa môže zmeniť .

# Zisťovanie požiadaviek



- ✧ Proces zhromažďovania informácií o požadovaných a existujúcich systémoch a filtrovanie užívateľských a systémových požiadaviek z týchto informácií.
- ✧ Interakcia prebieha so zainteresovanými stranami systému od manažérov až po externých regulátorov.
- ✧ Systémy majú zvyčajne celý rad zainteresovaných strán.





- ✧ Formálne alebo neformálne rozhovory so zainteresovanými stranami sú súčasťou väčšiny procesov špecifikácie
- ✧ Typy rozhovoru
  - Uzavreté rozhovory na základe vopred stanoveného zoznamu otázok
  - Otvorené rozhovory, kde sa skúmajú rôzne problémy so zainteresovanými stranami.
- ✧ Efektívny rozhovor
  - Bud'te otvorení, vyhnite sa vopred vytvoreným predstavám o požiadavkách a bud'te ochotní počúvať zainteresované strany.
  - Vyzvite účastníka rozhovoru, aby rozbehol diskusiu pomocou otázky odrazového mostíka, návrhu požiadaviek alebo spoluprácou na prototype systému.

# Rozhovory v praxi



- ✧ Zvyčajne ide o kombináciu uzavretých a otvorených rozhovorov.
- ✧ Rozhovory sú dobré na získanie celkového pochopenia toho, čo zainteresované strany robia a ako môžu interagovať so systémom .
- ✧ Anketári musia byť otvorení bez vopred vytvorených predstáv o tom, čo by mal systém robiť
- ✧ Musíte podnietiť používateľov, aby hovorili o systéme tým, že navrhnete požiadavky, a nie jednoducho sa ich spýtate, čo chcú.

# Problémy s pohovormi



- ✧ Aplikační špecialisti môžu na opis svojej práce používať jazyk, ktorý nie je pre inžiniera požiadaviek ľahké pochopiť.
- ✧ Rozhovory nie sú dobré na pochopenie doménových požiadaviek
  - Inžinieri požiadaviek nerozumejú špecifickej doménovej terminológii;
  - Niektoré znalosti domény sú také známe, že je pre ľudí ťažké ich formulovať alebo si myslia, že to nestojí za to formulovať ich.



- ✧ Sociálny vedec trávi značný čas pozorovaním a analýzou toho, ako ľudia skutočne fungujú.
- ✧ Ľudia nemusia svoju prácu vysvetľovať ani formulovať.
- ✧ Je možné pozorovať dôležité sociálne a organizačné faktory.
- ✧ Etnografické štúdie ukázali, že práca je zvyčajne bohatšia a zložitejšia, ako naznačujú jednoduché modely systémov.

# Rozsah etnografie



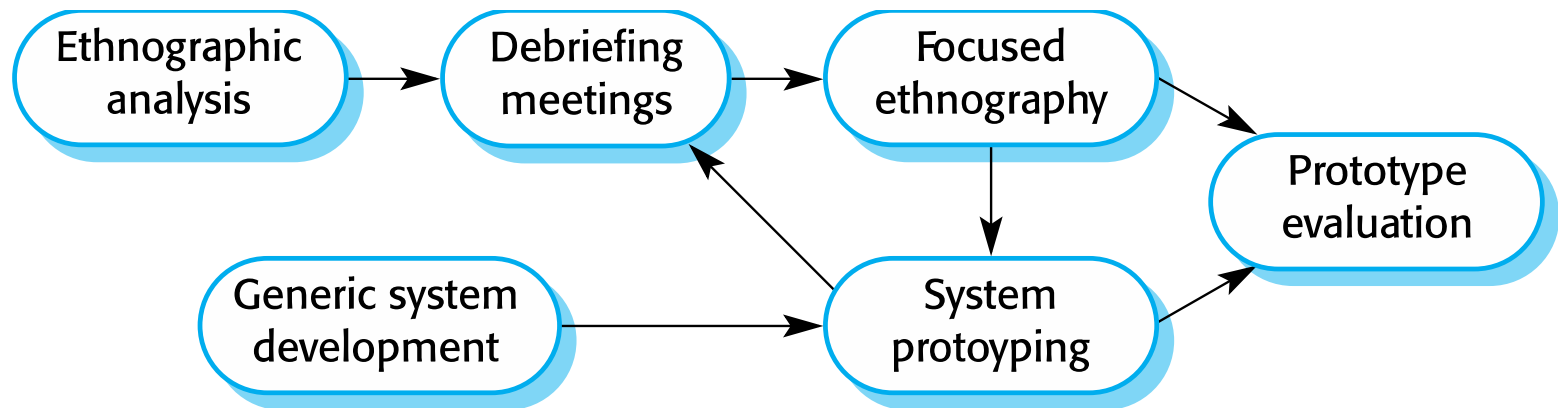
- ✧ Požiadavky, ktoré sú odvodené zo spôsobu, akým ľudia skutočne pracujú, a nie zo spôsobu, ktorý definície procesov naznačujú, že by mali pracovať.
- ✧ Požiadavky, ktoré sú odvodené od spolupráce a uvedomenia si aktivít iných ľudí .
  - Uvedomenie si toho, čo robia iní ľudia, vedie k zmenám v spôsoboch, akými robíme veci.
- ✧ Etnografia je účinná na pochopenie existujúcich procesov, ale nedokáže identifikovať nové funkcie, ktoré by mali byť pridané do systému.

# Zameraná etnografia



- ✧ Vyvinuté v projekte študujúcim proces riadenia letovej prevádzky
- ✧ Spája etnografiu s prototypovaním
- ✧ Výsledkom vývoja prototypu sú nezodpovedané otázky, na ktoré sa zameriava etnografická analýza.
- ✧ Problém etnografie je v tom, že študuje existujúce postupy, ktoré môžu mať nejaký historický základ, ktorý už nie je relevantný.

# Etnografia a prototypovanie pre analýzu požiadaviek



# Príbehy a scenáre



- ✧ Scenáre a príbehy používateľov sú skutočnými príkladmi toho, ako možno systém použiť.
- ✧ Príbehy a scenáre sú popisom toho, ako možno systém použiť na konkrétnu úlohu.
- ✧ Keďže sú založené na praktickej situácii, zainteresované strany sa k nim môžu vzťahovať a môžu komentovať svoju situáciu s ohľadom na príbeh.



# Scenáre



- ✧ Štruktúrovaná forma používateľského príbehu
- ✧ Scenáre by mali zahŕňať
  - Opis východiskovej situácie;
  - Popis normálneho toku udalostí;
  - Popis toho, čo sa môže pokaziť;
  - Informácie o iných súbežných činnostiach;
  - Popis stavu, keď sa scenár skončí.

# Špecifikácia požiadaviek



- ✧ Proces zapisovania používateľských a systémových požiadaviek do dokumentu požiadaviek.
- ✧ Požiadavky používateľov musia byť zrozumiteľné pre koncových používateľov a zákazníkov, ktorí nemajú technické vzdelanie.
- ✧ Systémové požiadavky sú podrobnejšie požiadavky a môžu obsahovať viac technických informácií.
- ✧ Požiadavky môžu byť súčasťou zmluvy na vývoj systému
  - Preto je dôležité, aby boli čo najkompletnejšie.

# Spôsoby písania špecifikácie systémových požiadaviek



Notový zápis	Popis
Prirodzený jazyk	Požiadavky sú napísané pomocou očíslovaných viet v prirodzenom jazyku. Každá veta by mala vyjadrovať jednu požiadavku.
Štruktúrovaný prirodzený jazyk	Požiadavky sú napísané v prirodzenom jazyku na štandardnom formulári alebo šablóne. Každé pole poskytuje informácie o aspekte požiadavky.
Špecifikačné a návrhové jazyky	Tento prístup používa jazyk ako programovací jazyk, ale s abstraktnejšími funkciami na špecifikáciu požiadaviek definovaním operačného modelu systému. Tento prístup sa v súčasnosti používa zriedka, hoci môže byť užitočný pre špecifikácie rozhrania.
Grafické zápisy	Na definovanie funkčných požiadaviek na systém sa používajú grafické modely doplnené textovými anotáciami; Bežne sa používajú diagramy prípadov použitia UML a sekvenčné diagramy.
Matematické špecifikácie	Tieto zápisy sú založené na matematických konceptoch, ako sú konečné stroje alebo množiny. Hoci tieto jednoznačné špecifikácie môžu znížiť nejednoznačnosť v dokumente požiadaviek, väčšina zákazníkov nerozumie formálnej špecifikácii. Nemôžu skontrolovať, či predstavuje to, čo chcú, a zdráhajú sa to prijať ako systémovú zmluvu

# Požiadavky a dizajn



- ✧ V zásade by **požiadavky** mali uvádzať, **čo** má systém robiť a **aký** má byť, a **návrh** by mal popisovať, **ako** to systém má robiť.
- ✧ V praxi sú požiadavky a dizajn neoddeliteľné
  - Architektúra systému môže byť navrhnutá tak, aby štruktúrovala požiadavky;
  - Systém môže spolupracovať s inými systémami, ktoré vytvárajú konštrukčné požiadavky;
  - Použitie špecifickej architektúry na uspokojenie nefunkčných požiadaviek môže byť doménovou požiadavkou .
  - Môže to byť dôsledok regulačnej požiadavky.

# Odporúčania pre písanie požiadaviek

---



- ✧ Vymyslite štandardný formát a použite ho pre všetky požiadavky.
- ✧ Používajte jazyk konzistentným spôsobom. Používa sa pre povinné požiadavky, malo by sa používať pre požadované požiadavky.
- ✧ Pomocou zvýraznenia textu identifikujte kľúčové časti požiadavky.
- ✧ Vyhnite sa používaniu počítačového žargónu .
- ✧ Zahrňte vysvetlenie (dôvod), prečo je požiadavka potrebná.

# Problémy s prirodzeným jazykom



## ✧ Nejasnosť

- Presnosť je ťažká bez toho, aby bol dokument ťažko čitateľný.

## ✧ Zmätok požiadaviek

- Funkčné a nefunkčné požiadavky bývajú zamieňané.

## ✧ Zlúčenie požiadaviek

- Spolu môže byť vyjadrených niekoľko rôznych požiadaviek.

# Príklad požiadaviek na softvérový systém inzulínovej pumpy



3.2 Systém meria hladinu cukru v krvi a podáva inzulín, ak je to potrebné, každých 10 minút. *(Zmeny hladiny cukru v krvi sú relatívne pomalé, takže častejšie meranie nie je potrebné; menej časté meranie by mohlo viesť k zbytočne vysokej hladine cukru.)*

3.6 Systém musí každú minútu spustiť samočinný test s podmienkami, ktoré sa majú testovať, a súvisiacimi činnosťami definovanými v tabuľke 1. *(Samočinný test môže odhaliť hardvérové a softvérové problémy a upozorniť používateľa na skutočnosť, že normálna prevádzka môže byť nemožné.)*

# Štruktúrované špecifikácie

---



- ✧ Prístup k písaniu požiadaviek, kde je obmedzená sloboda autora požiadaviek a požiadavky sú písané štandardným spôsobom.
- ✧ Toto funguje dobre pre niektoré typy požiadaviek, napr. požiadavky na vstavaný riadiaci systém, ale niekedy je to príliš rigidné na písanie požiadaviek na obchodný systém.



# Špecifikácie založené na formulároch

---



- ✧ Definícia funkcie alebo entity.
- ✧ Popis vstupov a odkiaľ pochádzajú.
- ✧ Popis výstupov a kam smerujú.
- ✧ Informácie o informáciách potrebných na výpočet a iných použitých entitách.
- ✧ Opis akcie, ktorá sa má vykonať.
- ✧ Podmienky pred a po (ak je to vhodné).
- ✧ Vedľajšie účinky (ak existujú).

# Štruktúrovaná špecifikácia požiadavky na inzulínovú pumpu



## Insulin Pump/Control Software/SRS/3.3.2

**Function** Compute insulin dose: safe sugar level.

### **Description**

Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

**Inputs** Current sugar reading (r2); the previous two readings (r0 and r1).

**Source** Current sugar reading from sensor. Other readings from memory.

**Outputs** CompDose—the dose in insulin to be delivered.

**Destination** Main control loop.

# Štruktúrovaná špecifikácia požiadavky na inzulínovú pumpu



## Action

CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

## Requirements

Two previous readings so that the rate of change of sugar level can be computed.

## Pre-condition

The insulin reservoir contains at least the maximum allowed single dose of insulin.

**Post-condition**      r0 is replaced by r1 then r1 is replaced by r2.

**Side effects**      None.

# Tabuľková špecifikácia



- ✧ Používa sa na doplnenie prirodzeného jazyka.
- ✧ Obzvlášť užitočné, keď musíte definovať niekoľko možných alternatívnych postupov .
- ✧ Napríklad systémy inzulínovej pumpy zakladajú svoje výpočty na rýchlosti zmeny hladiny cukru v krvi a tabuľková špecifikácia vysvetľuje, ako vypočítat potrebu inzulínu pre rôzne scenáre.

# Tabuľková špecifikácia výpočtu pre inzulínovú pumpu



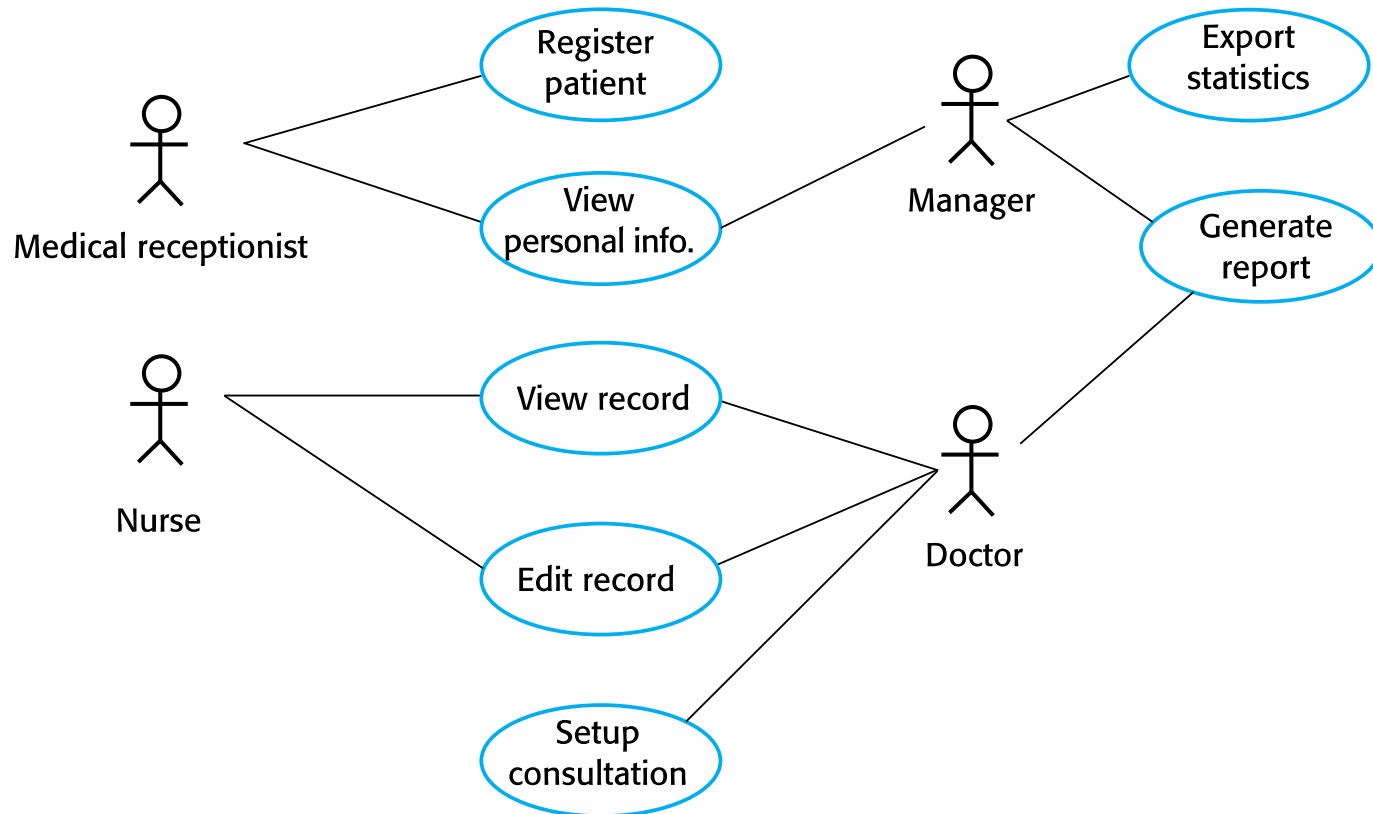
Podmienka	Akcia
Hladina cukru klesá ( $r2 < r1$ )	CompDose = 0
Stabilná hladina cukru ( $r2 = r1$ )	CompDose = 0
Hladina cukru sa zvyšuje a rýchlosť nárastu sa znižuje ( $(r2 - r1) < (r1 - r0)$ )	CompDose = 0
Zvyšovanie hladiny cukru a rýchlosť zvyšovania stabilná alebo stúpajúca ( $(r2 - r1) \geq (r1 - r0)$ )	CompDose = kolo $((r2 - r1)/4)$ Ak je zaokrúhlený výsledok = 0, potom CompDose = minimálna dávka

# Prípady použitia



- ✧ Prípady použitia sú akýmsi scenárom, ktorý je súčasťou UML .
- ✧ Prípady použitia identifikujú aktérov interakcie a popisujú samotnú interakciu.
- ✧ Súbor prípadov použitia by mal popisovať všetky možné interakcie so systémom .
- ✧ Vysokoúrovňový grafický model doplnený o podrobnejší tabuľkový popis (pozri kapitolu 5).
- ✧ Sekvenčné diagramy UML sa môžu použiť na pridanie detailov k prípadom použitia zobrazením postupnosti spracovania udalostí v systéme.

# Prípady použitia pre systém Mentcare



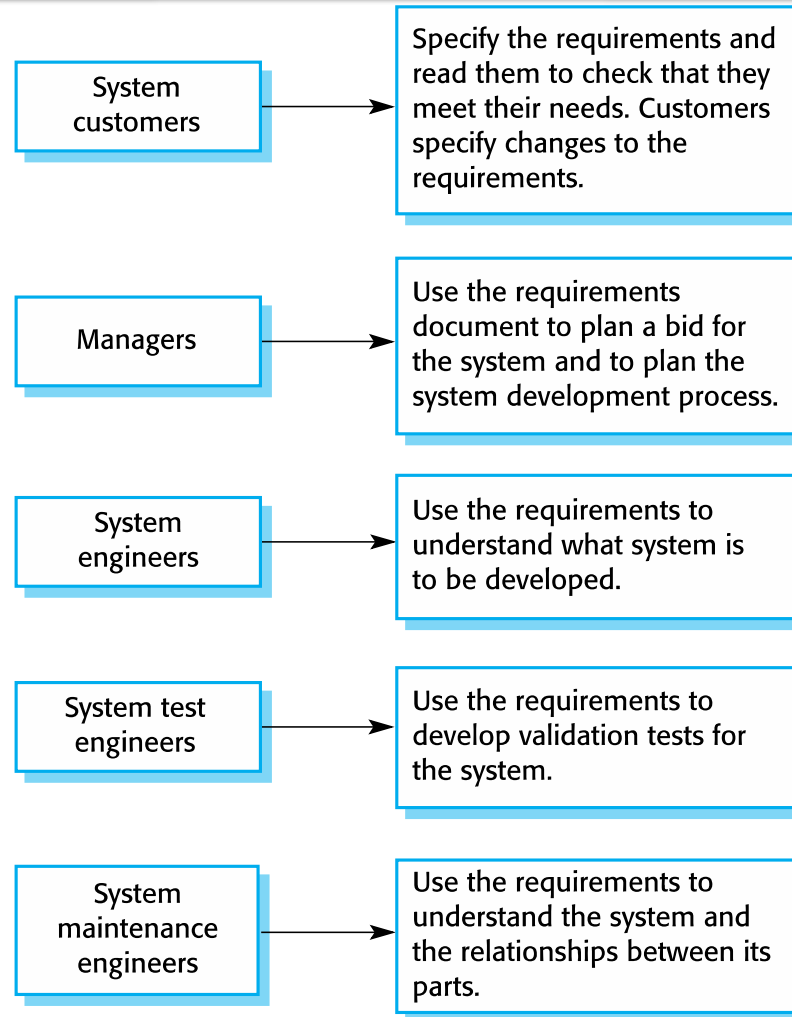
# Dokument s požiadavkami na softvér



- ✧ Dokument s požiadavkami na softvér je oficiálnym vyhlásením toho, čo sa vyžaduje od vývojárov systému.
- ✧ Mal by obsahovať definíciu požiadaviek užívateľa a špecifikáciu systémových požiadaviek.
- ✧ NIE JE to dizajnový dokument. Pokiaľ je to možné, **mal by stanoviť, ČO** by mal systém robiť, a nie AKO by to mal robiť.



# Používatelia dokumentu s požiadavkami



# Požiadavky dokumentujú variabilitu



- ✧ Informácie v dokumente požiadaviek závisia od typu systému a použitého prístupu k vývoju.
- ✧ Systémy vyvíjané postupne budú mať zvyčajne menej podrobností v dokumente požiadaviek.
- ✧ Boli navrhnuté štandardy dokumentov požiadaviek, napr. štandard IEEE. Tieto sú väčšinou použiteľné pre požiadavky veľkých projektov systémového inžinierstva.

# Overenie požiadaviek

---



- ✧ Zaoberá sa preukázaním, že požiadavky definujú systém, ktorý zákazník skutočne chce.
- ✧ Náklady na chyby v požiadavkách sú vysoké, takže validácia je veľmi dôležitá
  - Oprava chyby požiadaviek po doručení môže stáť až 100-násobok nákladov na opravu chyby implementácie.

# Kontrola požiadaviek



- ✧ Platnosť. Poskytuje systém funkcie, ktoré najlepšie podporujú potreby zákazníka?
- ✧ Dôslednosť . Existujú nejaké konflikty požiadaviek?
- ✧ Úplnosť. Sú zahrnuté všetky funkcie požadované zákazníkom?
- ✧ Realizmus. Môžu byť požiadavky implementované vzhľadom na dostupný rozpočet a technológiu?
- ✧ Overiteľnosť. Je možné skontrolovať požiadavky?

# Techniky validácie požiadaviek

---



## ✧ Recenzie požiadaviek

- Systematická manuálna analýza požiadaviek.

## ✧ Prototypovanie

- Použitie spustiteľného modelu systému na kontrolu požiadaviek. Popísané v kapitole 2.

## ✧ Generovanie testovacích prípadov

- Vývoj testov pre požiadavky na kontrolu testovateľnosti.

# Recenzie požiadaviek



- ✧ Počas formulovania definície požiadaviek by sa mali vykonávať pravidelné kontroly.
- ✧ Do kontrol by sa mali zapojiť zamestnanci klienta aj dodávateľa.
- ✧ Preskúmania môžu byť formálne (s vyplnenými dokumentmi) alebo neformálne. Dobrá komunikácia medzi vývojármi, zákazníkmi a používateľmi môže vyriešiť problémy v počiatočnom štádiu.

# Kontrolné kontroly



## ✧ Overiteľnosť

- Je požiadavka reálne testovateľná?

## ✧ Zrozumiteľnosť

- Je požiadavka správne pochopená?

## ✧ Vysledovateľnosť

- Je jasne uvedený pôvod požiadavky?

## ✧ Prispôsobivosť

- Dá sa požiadavka zmeniť bez veľkého vplyvu na ostatné požiadavky?

# Meniace sa požiadavky



- ✧ Obchodné a technické prostredie systému sa vždy po čase mení.
  - Môže sa zaviesť nový hardvér, môže byť potrebné prepojenie systému s inými systémami, môžu sa zmeniť obchodné priority (s následnými zmenami v potrebnej podpore systému) a môžu sa zaviesť nové právne predpisy a predpisy, ktoré musí systém nevyhnutne dodržiavať.
- ✧ Ľudia, ktorí platia za systém, a používatelia tohto systému sú len zriedka tí istí ľudia.
  - Zákazníci systému kladú požiadavky z dôvodu organizačných a rozpočtových obmedzení. Tieto môžu byť v rozpore s požiadavkami koncových používateľov a po dodaní môže byť potrebné pridať nové funkcie na podporu používateľov, ak má systém splniť svoje ciele.



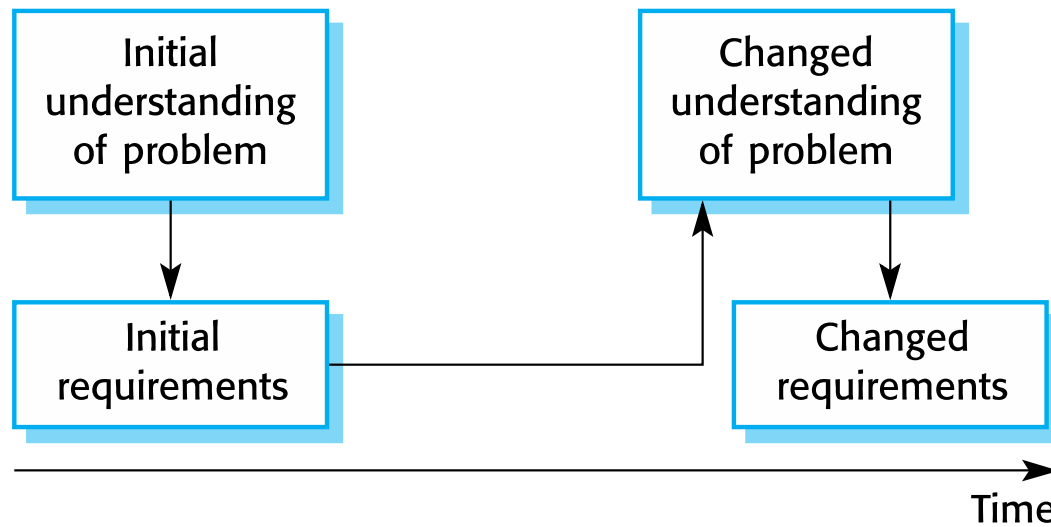
# Zmena požiadaviek

---



- ✧ Veľké systémy majú zvyčajne rôznorodú komunitu používateľov, pričom mnohí používatelia majú rôzne požiadavky a priority, ktoré môžu byť protichodné.
  - Konečné systémové požiadavky sú nevyhnutne kompromisom medzi nimi a na základe skúseností sa často zistí, že vyváženie podpory poskytovanej rôznym používateľom sa musí zmeniť.

# Vývoj požiadaviek



# Riadenie požiadaviek



- ✧ Riadenie požiadaviek je proces riadenia meniacich sa požiadaviek počas procesu inžinierstva požiadaviek a vývoja systému .
- ✧ Nové požiadavky vznikajú pri vývoji systému a po jeho uvedení do prevádzky.
- ✧ Musíte sledovať jednotlivé požiadavky a udržiavať prepojenia medzi závislými požiadavkami, aby ste mohli posúdiť vplyv zmien požiadaviek. Musíte vytvoriť formálny proces na vytváranie návrhov zmien a ich prepojenie so systémovými požiadavkami.

# Plánovanie manažmentu požiadaviek



- ✧ Stanovuje úroveň podrobností o správe požiadaviek, ktorá sa vyžaduje.
- ✧ Rozhodnutia manažmentu požiadaviek:
  - *Identifikácia požiadaviek* Každá požiadavka musí byť jednoznačne identifikovaná, aby sa dala porovnať s inými požiadavkami.
  - *Proces riadenia zmien* Ide o súbor činností, ktoré hodnotia vplyv a náklady zmien. Podrobnejšie sa tomuto procesu venujem v nasledujúcej časti.
  - *Politiky sledovateľnosti* Tieto zásady definujú vzťahy medzi každou požiadavkou a medzi požiadavkami a návrhom systému, ktorý by sa mal zaznamenať.
  - *Podpora nástrojov* Nástroje, ktoré možno použiť, siahajú od špeciálnych systémov na správu požiadaviek až po tabuľky a jednoduché databázové systémy.

# Riadenie zmeny požiadaviek



✧ Rozhodovanie, či by sa mala akceptovať zmena požiadaviek

- *Analýza problému a špecifikácia zmeny*

- Počas tejto fázy sa problém alebo návrh zmeny analyzuje, aby sa skontrolovalo, či je platný. Táto analýza sa odošle späť žiadateľovi o zmenu, ktorý môže odpovedať špecifickejším návrhom na zmenu požiadaviek alebo sa môže rozhodnúť žiadosť stiahnuť.

- *Analýza zmien a kalkulácia*

- Účinok navrhovanej zmeny sa hodnotí pomocou informácií o sledovateľnosti a všeobecných znalostí systémových požiadaviek. Po dokončení tejto analýzy sa rozhodne, či pokračovať v zmene požiadaviek alebo nie.

- *Implementácia zmeny*

- Upraví sa dokument s požiadavkami av prípade potreby aj návrh a implementácia systému. V ideálnom prípade by mal byť dokument usporiadaný tak, aby sa dali jednoducho implementovať zmeny.

# Riadenie zmeny požiadaviek

