5 Požadavky a jejich specifikace

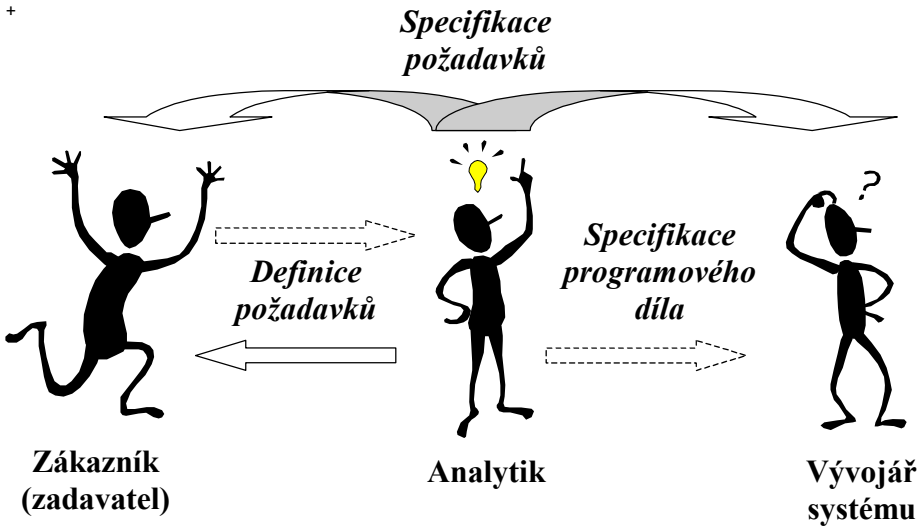
5.1 Inženýrství požadavků (requirements engineering)

- proces stanovení služeb, které by měl vyvíjený systém poskytovat a omezení, za nichž musí pracovat
- CO má systém dělat, ne JAK je to zařídit
- požadavky - funkční - popisují požadovanou službu systému
- nefunkční - omezení kladená na systém nebo proces vývoje

Definice požadavků - zadání v přirozeném jazyce, příp. diagramy udávající požadované služby systému a omezení. Je vytvořen na základě informace od zákazníka. (manažerská úroveň)

Specifikace požadavků - strukturovaný dokument, který detailně popisuje služby a omezení. Může sloužit k uzavření kontraktu. (technická úroveň)

Specifikace programového díla - abstraktní popis vyvíjeného programového systému, základ pro návrh a implementaci, může obsahovat další detaily. (implementační úroveň)



Př)

Definice: Program musí zobrazovat obrázky uložené v různých grafických formátech.

Specifikace:

Program musí:

- umožnit zobrazovat v prostředí MS Windows 95 obrázky v grafických formátech PCX, TIFF, JPEG a GIF.

- poskytnout uživateli formou dialogu možnost výběru zobrazovaného souboru; různý grafický formát souborů by měl být rozlišen různými ikonami.
- V rámci dialogu by mělo být možné zobrazit náhled obrázku.
- ...

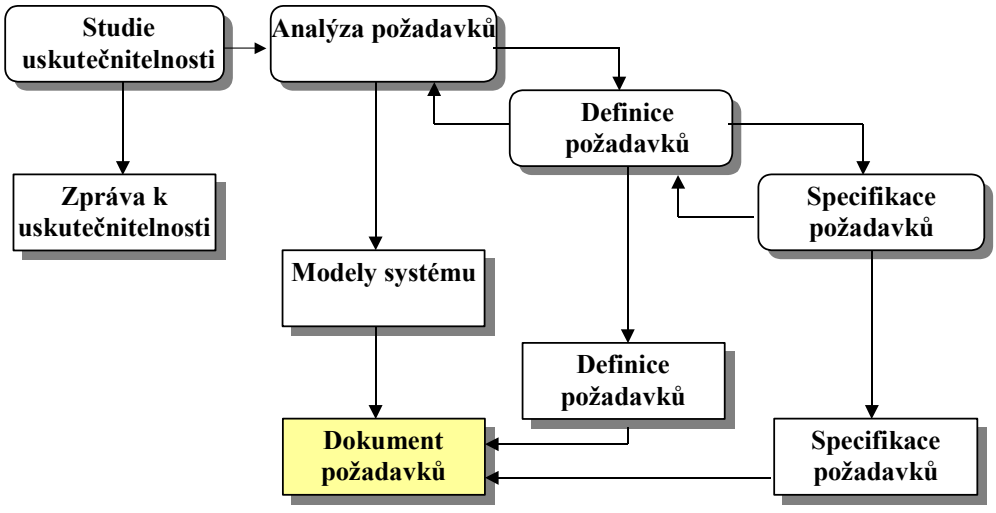
- často v přirozeném jazyce - nevýhody →strukturovanější zápis

Př) Grafický editor

> 3.4.1 Kopírování prvků
>
> 3.4.1.1 Editor poskytne prostředek, který umožní uživateli vytvořit kopii specifikovaného prvku. Kopie je vytvářena od právě vybraného prvku (viz 5.1).
>
> 3.4..1.2 Operace je provedena následující posloupností akcí:
>
> 1. Uživatel určí místo, kde chce kopii vytvořit, pomocí kurzoru.
>
> *Zdůvodnění:* Uživatel nejlépe ví, kde chce kopii umístit.
> *Specifikace*: spec3_4_1

Cíl: úplné pochopení požadavků a omezení, základ pro kontrakt, základ návrhu a implementace.

- Proces inženýrství požadavků



- Dokument požadavků na programové dílo
  - oficiální dokument o tom, co se od vývojářů požaduje (CO)
  - dokument by měl být dobře strukturovaný s minimem vzájemných odkazů (změny)
  - jde o kombinaci definice a specifikace požadavků
  - možná struktura:

- ♦ *Úvod* - potřeba, stručně funkce, okolí, vztah ke strategickým cílům organizace,
- ♦ *Slovník pojmů* - definice použitých technických termínů,
- ♦ *Modely systému* - modely ukazující vztahy prvků systému a vztahy systému a okolí,
- ♦ *Definice funkčních požadavků* - popis poskytovaných služeb,
- ♦ *Definice nefunkčních požadavků* - omezení na produkt a proces vývoje (výkonnost, odezva, standardy, …),
- ♦ *Rozvoj systému* - předpoklady, na nichž je systém založen, očekávané změny díky rozvoji HW, měnících se potřeb uživatele,
- ♦ *Specifikace požadavků* - detailní popis funkčních požadavků, případné zpřesnění nefunkčních,
- ♦ *Validační kritéria* - třídy testů pro ověření implementace

případně ještě:

- ♦ *Hardware* - popis speciálního HW, konfigurace kupovaného HW,
- ♦ *Databáze* - datový model,
- ♦ *Bibliografie*
- ♦ *Index* - pro lepší organizaci.

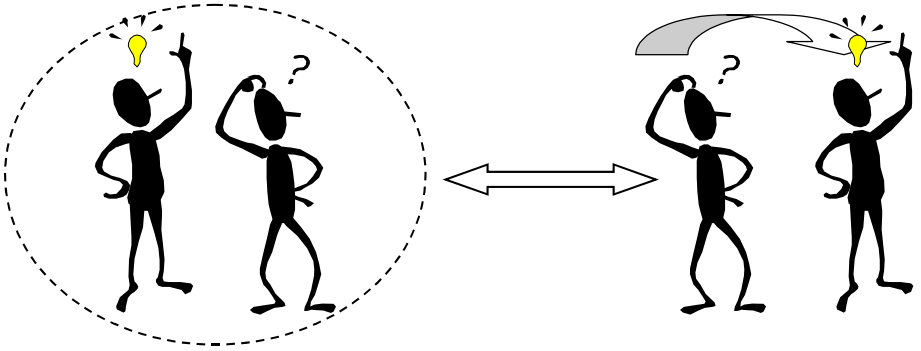
- existují standardy (IEEE, US DoD, ...)

- Validace požadavků

- požadavky musí definovat systém, jaký chce zákazník
- hlavní kontrolované aspekty:
  - Platnost
  - Úplnost
  - Konzistence
  - Reálnost
- uživatel musí za definicí a specifikací vidět požadovaný systém ⇒ srozumitelnost, využití prototypování
- pravidelné posuzování uživateli a vývojáři (oponentury - formální, neformální)

- Vývoj požadavků
  - v průběhu analýzy se požadavky zpřesňují a mění, mění se i v průběhu vývoje (rozsáhlé projekty) ⇒ „zmrazit“/počítat s nimi
  - dokument požadavků by měl být konzistentní se systémem

5.2 Analýza požadavků

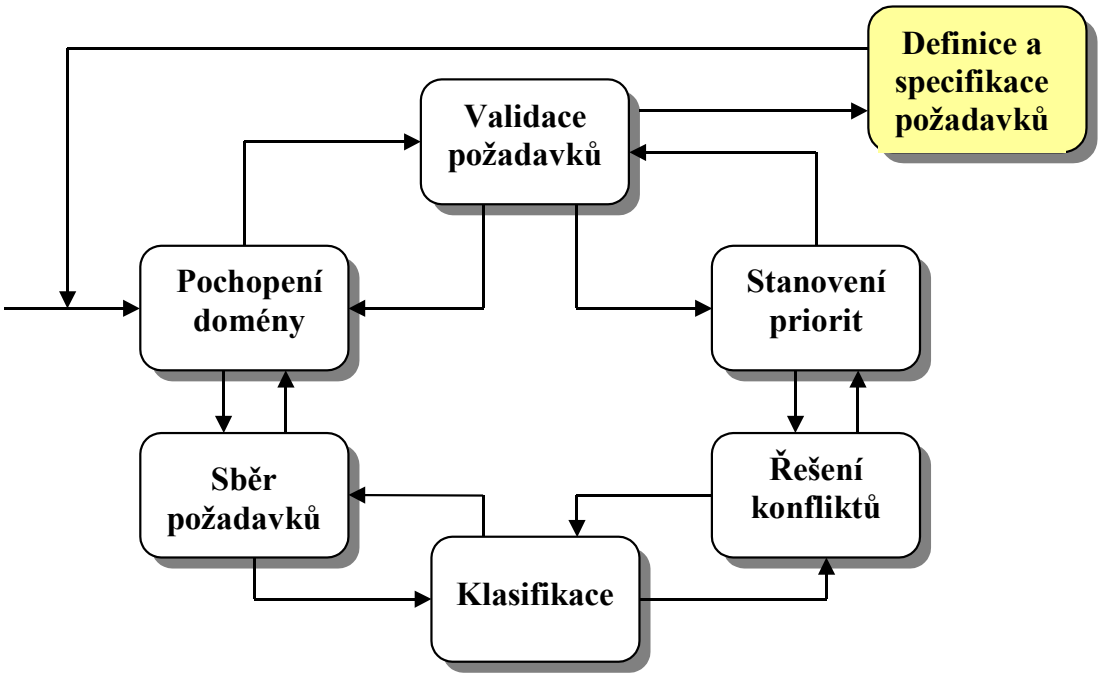


Uživatel (stakeholder)

Analytik

Problémy:

- nejasná představa, neschopnost zformulovat,
- neochota spolupracovat,
- různé „jazyky“,
- různí uživatelé, různé požadavky, analytik musí najít všechny potřebné zdroje,
- měnící se prostředí → mění se požadavky.



- během analýzy může vzniknout několik různých modelů

- **Analýza orientovaná na pohledy (viewpoint-oriented)**
  - **zpravidla více typů koncových uživatelů**
  - **Př) Informační systém fakulty**
    - **oddělení děkanátu, ústavy, studenti, knihovna, rozvrhář, správce sítě, správce výpočetní techniky, správce programového vybavení, …**
- **Analýza založená na metodě (method-based)**
  - **nejrozšířenější přístup**
  - **výsledkem aplikace metody (metodologie, metodiky) je sada modelů systému**
  - **metody zaměřené pouze na analýzu, jiné blíže návrhu**
  - **metoda typicky zahrnuje:**
    - ♦ ***Definici postupu*** **- návaznost kroků,**
    - ♦ ***Modelovací techniky*** **- typy modelů a notace,**
    - ♦ ***Závazná pravidla pro modely*** **- jména, konzistence, …,**
    - ♦ ***Doporučení pro návrh*** **- pro dosažení dobrého návrhu,**
    - ♦ ***Šablony zpráv*** **- způsob prezentace (diagramy + text).**

- **Okolí systému**
  - **potřeba stanovení hranic systému (zákazník + analytik)**
  - **typicky reprezentace kontextovým diagramem**

- **Sociální a organizační faktory**
  - **analytik musí vnímat při analýze lidské i obchodní faktory (použitelnost systému, cíl nasazení), nejsou dobře definované → neexistuje systematický přístup k jejich analýze**
  - **Př) IS knihovny**
    snaha snížit počet knihovníků →neochota spolupracovat

5.3 Techniky komunikace

zákazník ⟷ analytik ⟷ vývojář

- zahájení procesu - *interview*

• FAST - Facilitated Application Specification Technique

- týmově-orientovaný přístup pro počáteční etapy analýzy

- příklad: Joint Application Development (JAD) - IBM

- společné rysy:

- **♦ schůzky týmu,**
- **♦ neutrální místo,**
- **♦ pravidla přípravy a účasti,**
- **♦ dostatečně formální agenda (ale ne příliš),**
- **♦ moderátor,**
- **♦ použití „mechanismu definice“ (náčrty, postery, ...),**
- **♦ cíl: identifikace problému, návrh prvků řešení, vyjednávání**

Př) scénář:

1. **interview - „požadavek na produkt“**
2. **domluva schůzky týmu**
3. **podklady účastníkům, příprava na schůzku (seznam objektů)**
4. **schůzka - diskuse potřeby produktu, každý své poznámky,**

společné seznamy, diskuse, odsouhlasení, rozdělení na menší týmy - minispecifikace položek seznamů, prezentace, seznam validačních kritérií, zápis definice požadavků.

Požadavek na produkt:

Průzkum ukazuje, že trh systémů pro zabezpečení domácnosti roste rychlostí 40% ročně. Rádi bychom vstoupili na tento trh se systémem na bázi mikroprocesoru, který by rozpoznal a chránil proti různým nežádoucím situacím jako je nelegální vstup, požár, záplava atd. Produkt, který bude předběžně nazván SafeHome, bude používat vhodné senzory k detekci každé situace, může být programován vlastníkem bytu a bude automaticky telefonovat monitorovací agentuře, když je detekována příslušná situace.

Minispecifikace řídicího panelu:

Připevněn na stěně, velikost asi 20 x 12 cm, obsahuje standardní klávesnici s 12 klávesami a speciálními klávesami, obsahuje plochý displej podle obrázku, veškerá interakce s uživatelem přes klávesy, lze jeho prostřednictvím celý systém zablokovat/odblokovat, SW poskytuje základní návod.

5.4 Nefunkční požadavky

- definují vlastnosti systému jako celku a omezení
- týkají se produktu i procesu vývoje (kvalita, udržovatelnost)
- typy nefunkčních požadavků:
  - požadavky na produkt
    - ♦ na použitelnost
    - ♦ na efektivnost
      - ◊ výkonnostní
      - ◊ prostorové
    - ♦ na spolehlivost
    - ♦ na přenositelnost
  - požadavky na proces
    - ♦ na dodání
    - ♦ na implementaci
    - ♦ na standardy
  - externí požadavky
    - ♦ na součinnost (interoperability)
    - ♦ etické

    - ♦ legislativní
      - ◊ ochrana soukromí
      - ◊ bezpečnostní

- nefunkční požadavky musí být ověřitelné

Př) přátelskost →doba na zaškolení, požadovaná kvalifikace, …

- příklady možných metrik:

Rychlost:	transakce/s, doba odezvy,
Velikost:	kód, požadavky na diskový prostor, paměť RAM,
Použitelnost:	doba zaškolení, rozsah nápovědy,
Spolehlivost:	střední doba bezporuchového provozu, pravděpodobnost nedostupnosti, četnost poruch a chyb,
Robustnost:	doba obnovy po poruše, pravděpodobnost zničení dat při poruše,
Přenositelnost:	procento závislého kódu, počet cílových systémů.