

Uživatelské požadavky, definice funkcionality systému, vztah mezi funkcionálními a nefunkcionálními (např. kapacita, škálovatelnost, robustnost) požadavky na systémy, technická specifikace

1. Význam požadavků

Inženýrství požadavků (SW Requirements engineering) je termín používaný k popisu aktivit souvisejících se zjišťováním, dokumentací a údržbou množiny požadavků na softwarový systém.

- Zastupuje odhalování způsobu, jak a k čemu uživatelé daný systém potřebují.
- Jedná se o klíčový faktor celého vývoje softwarového projektu.
- Nedostatečně specifikované požadavky a nedostatečné zapojení uživatelů, jsou dvěma hlavními **příčinami konečného neúspěchu** celého projektu.

2. Definice požadavků

Požadavek lze definovat jako „**specifikaci toho, co by mělo být implementováno**“. Rozlišují se dva typy požadavků:

- **Funkční požadavky**, které určují, jaké chování bude systém nabízet.
- **Nefunkční požadavky**, které určují vlastnosti nebo omezující podmínky, kladené na daný systém.

Požadavky jsou základem systému, neboť jsou vyjádřením toho, **CO** by měl systém dělat, nikoli však způsobu, **JAK** toho docílit. Požadavky je vhodné uspořádat do **taxonomie** (**FURPS** - functionality, usability, reliability, performance, supportability). Nejjednodušší rozdělení požadavků je na funkční a nefunkční. Lze je však dále jemněji uspořádávat do kategorií, které závisí na typu vyvíjeného softwaru. K těmto účelům je velmi užitečné použít nástroj pro správu požadavků. Na základě požadavků dochází k vypracování **Use Case** pro systémy, kde to má smysl (interakce s uživateli) a **analýzy**.

3. Chyby při sběru požadavků

- špatně vyhodnocené požadavky
- nepřesné požadavky
- neúplné požadavky
- navzájem si odporující požadavky
- nevyřčené, ale přesto očekávané požadavky
- přehnané požadavky
- obtížně verifikovatelné (nekvantitativní) požadavky – např. systém bude potřebovat málo paměti
- nedostatečné zapojení zákazníka a uživatelů do procesu získávání požadavků
- chybějící priorita
- vytváření něčeho, co nikdo nechtěl
- specifikace není v systému pro správu verzí

- nedefinovaný proces změnových řízení

4. Funkční požadavky

Funkční požadavek definuje konkrétní funkci softwarového systému nebo jeho komponenty. Tato funkce je popsána jako **množina vstupů, chování a výstupů**. Funkčními požadavky mohou být výpočty, manipulace s daty a jejich zpracování či jiné specifické operace, které by systém měl být schopen splnit. Plán implementace funkčních požadavků je upřesněn ve fázi návrhu systému.

Formulace požadavku má typicky následující podobu:

<ID> <název systému> musí/může/nesmí <popis funkce>

5. Nefunkční požadavky

Nefunkční požadavky podporují množinu funkčních. Často jsou označovány jako “kvality systému” a definují, **za jakých okolností bude systém pracovat**. Kladou specifická omezení a vlastnosti na systém jako celek, jeho návrh a vývoj. Nefunkční požadavky **ovlivňují architekturu systému**.

Některé kategorie nefunkčních požadavků:

- Dostupnost - vyjadřuje poměr části časového intervalu, kdy je systém použitelný, ku celkové délce tohoto intervalu.
- Robustnost - vyjadřuje schopnost systému vypořádat se s eventuálními výjimečnými událostmi, které mohou nastat při jeho běhu.
- Škálovatelnost - vyjadřuje schopnost adaptace respektive možnosti rozšíření systému ve vztahu k rostoucím objemům práce.
- Výkon - obecně vyjadřuje objem času a zdrojů, které systém potřebuje ke splnění nějakého konkrétního úkolu.
- Bezpečnost - vyjadřuje stupeň ochrany vůči nebezpečí, škodám, ztrátě a kriminálním aktivitám.
- Shoda se standardy
- integrace s jinými systémy
- formát vstupních a výstupních dat
- běhové prostředí (OS, verze Javy, verze DB)
- použité technologie
- komunikace s jinými systémy

6. Specifikace systémových požadavků

SRS - (Software Requirements Specification) - je dokument obsahující kompletní popis chování vyvíjeného systému. Obsahuje všechny funkční i nefunkční požadavky. Funkční požadavky jsou zde obvykle vyjádřeny i ve formě případů užití., ve kterých je podrobně popsán způsob interakce uživatelů se systémem.

Na základě SRS lze dělat rozhodnutí, zda hlášený problém je 1) chyba 2) požadavek na změnu 3) šedá zóna mezi tím. SRS má obvykle podobu strukturovaného dokumentu. Méně často pak podobu katalogu požadavků.

SRS musí sloužit pro:

- rozhodnutí, co patří do definovaného rozsahu projektu a co ne
- ustanovení kontraktu pro vývoj
- akceptační testování

Typická struktura dokumentu SRS:

- Úvod
- Účel dokumentu
- Rozsah projektu
- Definice
- Přehled systému
- Reference
- Souhrnný popis
- Perspektiva produktu
- Funkce produktu
- Charakteristika uživatelů
- Omezení, předpoklady a závislosti
- Specifické požadavky
- Externí rozhraní
- Funkce
- Požadavky na výkon
- Požadavky na databáze
- Omezení návrhu

7. Use Case

Případy užití sestávají z **Use case text** a **Use case diagrams**. UCT je v podstatě scénář uživatelské akce.

Pojmy k UCD:

- vazby include a extend
- actors a jejich hierarchie
- hranice systému

Struktura UCT:

- priorita
- preconditions a postconditions
- tok hlavního scénáře a rozvětvení (klíčová slova if, for, else)
- alternativní scénáře

8. Zdroje informací

[1] ARLOW, Jim, NEUSTADT, Ila. UML2 : a unifikovaný proces vývoje aplikací. Brno : Computer Press, a.s., 2007. 567 s.

[2] Přednášky předmětu A4M33SEP - Softwarové inženýrství pro praxi

[3] http://en.wikipedia.org/wiki/Software_Requirements_Specification

[4] http://en.wikipedia.org/wiki/Functional_requirements

[5] http://en.wikipedia.org/wiki/Non-functional_requirements