**Funkční požadavky**

*Funkční požadavky představují základní předmět systému a jsou měřeny konkrétními prostředky jako jsou například hodnoty dat, logika a algoritmy rozhodování. Funkční požadavky specifikují, co má produkt dělat.*

***NPŘ:***

* Registrace uživatelů
* Možnost přidávání, editování, mazání pouze svých příspěvků
* Přidávat komentáře

**Nefunkční požadavky**

*Nefunkční požadavky představují vlastnosti v oblasti chování, které musí mít stanovené funkce jako například výkonnost, uživatelnost atd. Nefunkčním požadavkům lze přiřadit konkrétní metodu mení. Tento vzor uvádí příklady kvantifikace nefunkčních požadavků. Nefunkční požadavky specifikují, jaké vlastnosti má produkt mít.*

* + Rychlost (počet transakcí za sekundu)
  + Velikost (počet MB na soubor)
  + Jednoduchost na použití (doba výcviku zaměstnanců)
  + Spolehlivost (pravděpodobnost nespolehlivosti)
  + Robustnost (čas, po kterém se restartuje)
  + Portabilita (Počet cílových systémů)

**Požadavky produktu –** požadavky určující kvalitu s jakou bude produkt fungovat, jako rychlost, spolehlivost etc.

**Organizační požadavky** – organoizační procedury a omezení – standardy procesu, implementační požadavky eg.

**Externí požadavky** – faktory, které jsou systému externí, například interoperabilita, požadavky legislativy eg.

**Dostupnost –** Pravděpodobnost, že systém bude v dané časové době funkční a schopen podat pořadované služby.

* Jak dlouho by měl systém fungovat bez selhání
* Jak nejdéle by měl být mimo provoz

**Spolehlivost –** Pravděpodobnost, že systém bude po určitou časovou dobu fungovat bez pádu, za určitou časovou dobu za určitých podmínek

* Jak je detekována systémová chyba
* Jak často může chyba nastat
* Co se bude dít, až chyba nastane
* Jednalo- li se o systémovou chybu, chybu uživatele, selhhání systému

**Bezpečí –** vlastnost, že systém může fungovat bez ohrožení, které by se odráželo na jeho schopnosti fungovat nebo aby nezpůsobil žádnou škodu lidskému životu, či prostředí systému.

* Nehoda, hazard, poškození, risk

**Bezpečnost** – vlastnost odrážející schopnost systému bránit se proti externím útokům

**Výkon –** je o časování, odpověď na určité události (přerušení, zprávy, požadavky od uživatelů) Citlivé na hardware, počet připojených uživatelů, operační systém, implementaci.

* Pro finanční systém například (kolik transakcí se udělá za vteřinu)

**Možnost modifikace –** cena za změnu

* Co se může změnit? Platforma, na které systém běží, prostředí, ve kterém běží. Tato možnost by měla být obsáhnuta při instalaci

**Možnost testování** - Možnost, v jakých podmínkách je schopen systém demonstrovat své vlastnosti v testování. (Až 40% období vývoje systému zahrnuje testování)

* Ukazuje se, jak efektivní jsou testy na určité úrovni
* Aby byl systém správně testovatelný, musí být možné kontrolovat vstup a výstup každé komponenty

**Použitelnost –** se zabývá tím, jak jednoduchý je software k použití pro uživatele

* Nauka systémových vlastností
* Efektivně používat systém
* Minimalnizovat dopad chyb
* Adaptovat systém uživatelským potřebám
* Zvýšení důvěry a spokojenosti

**Ogranizační požadavky**

* Vývojové požadavky (programovací jazyk, vývojové prostředí)
* Operační požadavky (Platforma, na které systém poběží, životnost)
* Požadavky prostředí (integrace s legacy systems)

**Time to Market –** Pokud je vývoj či vydání systému ohroženo konkurencí, je doba vývoje důležitá

* Často se používají starší již použité elementy nebo COTS

**Propustnost**

* **-** Počet transakcí za sekundu (tps)
* Počet zpracovaných zpráv za sekundu (mps)

**Doba odezvy**

zpoždění při zpracování byznys transakce

**Blokování**

Zpracování požadavku může být zpožděno nebo blokováno kvůli potřebě vyčkat na uvolnění zdrojů nebo na nutnost ukončení procesu, na který další proces navazuje.

**Optimalizace pro různé prohlížeče**

Systém by měl být funkční na nejrozšířenějších prohlížečích. K dosažení tohoto cíle je nutné vytvořit správný a validní kód a provést testování aplikace.

**Udržovatelnost a rozšiřitelnost**:

Musíme specifikovat, zda bude možné nadále rozšiřovat aplikaci o další funkčnost a jak by byl celý proces časově a finančně náročný.

Udržitelnost definuje schopnost opravení nedostatků systému bez toho, aby se ovlivnila některá další část

**Škálovatelnost**

Škálovatelnost je schopnost podpořit kvalitu služeb tím, že zvýšíme kapacity systému bez toho, aby došlo ke změnám v software.

Kapacita systémů může být definována například jako počet simultánně běžících procesů nebo počet online uživatelů. Pokud systém umí navyšovat kapacity podle toho, jaké jsou zrovna na software vyvíjeny nároky, a neustále drží požadovaný výkon na akceptovatelném limitu, tak se jedná o dobře škálovatelný systém.