

Přehled metodiky vývoje GIS aplikací

(vytvoreno pro seminář na FIMU: Vybrané kapitoly z GIS, podzimní semestr)

Lekce 2: Model požadavků

Kategorizace požadavků	2
Hlediska (dimenze) kvality software podle metody FURPS.....	2
Častá rozšíření FURPS	2
Hlavní činnosti při analýze požadavků	3
Identifikace zúčastněných stran (stakeholders)	3
Interview se zúčastněnými stranami.....	3
Vytvoření seznamu (katalogu) požadavků	3
Vytvoření modelu požadavků.....	4
Stanovení měřitelných cílů	5
Prototypování.....	5
Přiřazení požadavků prvkům modelu navrhovaného systému	5

Kategorizace požadavků

Požadavky lze kategorizovat několika způsoby. Jednu z možností poskytuje metoda FURPS, která byla vytvořena společností [Hewlett-Packard](#) na základě potřeby definovat, jak poznat a ověřit kvalitu dodávaného [software](#). První zmínky o této metodě pocházejí z roku 1986 a veřejně myšlenky publikovali Robert Grady a Deborah Caswell v knize „Software Metrics: Establishing a Company–Wide Program“ v roce 1987.

FURPS se dívá na kvalitu software nebo informačního systému z pěti základních hledisek: funkčnost, užitečnost, spolehlivost, výkon a rozšiřitelnost. Metoda na nejvyšší úrovni definuje, co by mělo být hodnoceno, ale nespécifikuje, jakým způsobem mají být oblasti hodnoceny. V těchto jednotlivých oblastech musí být dále vytvářeny konkrétní metriky a jejich hodnoty. Pokud chcete dodávat zákazníkům opravdu dobrý software, tak je vhodné začít už při plánování testování zohledňovat tyto oblasti.

Hlediska (dimenze) kvality software podle metody FURPS

- **F (functionality) – funkčnost.** Zaměřuje se na hlavní funkčnosti a schopnosti programu (chování), zda software naplňuje požadavky byznysu a podporuje byznys procesy.
- **U (usability) – vhodnost k použití.** Hodnotí se zejména z pohledu koncového uživatele; jak snadno lze aplikaci použít, jakým celkovým dojmem působí aplikace, dokumentace a školící materiály.
- **R (reliability) – spolehlivost.** Jedná se o hodnocení četnosti a závažnost chyb, přesnosti zpracování vstupů a výstupů. Pro vyjádření spolehlivosti se často používá metrika [MTBF](#) (mean time between failures), což je střední doba mezi chybami nebo selháními. V této oblasti se také sledují možnosti obnovení provozu a zotavení se z výpadku. Patří sem také zátěžové testy a testy zotavení aplikace po selhání některých komponent řešení.
- **P (performace) – výkon.** Hodnocení celkové rychlosti odezev systému a zpracování klíčových byznys aktivit. Zároveň se sledují i technické parametry testovaného systému, např. vytížení zdrojů OS, zatížení síťového provozu, rozložení zátěže na jednotlivé komponenty systému.
- **S (supportability) – schopnost být udržována.** Dalším hlediskem hodnocení je oblast údržby a podpory aplikace, její testovatelnosti. V této oblasti se taktéž hodnotí i přizpůsobitelnost a rozšiřitelnost o nové vlastnosti. Důležitá je také schopnost zapojení aplikace do existujících procesů podpory a údržby SW.

V poslední době se metoda FURPS rozšířila o znaménko +, na FURPS+. Postupem času totiž přestaly jednotlivé kategorie dostačovat, a proto bylo zavedeno toto „plus“, které označuje, že je metoda rozšířena o další dílčí elementy nebo podoblasti.

Častá rozšíření FURPS

- Implementace – různá omezení, jazykové mutace, speciální nástroje a HW
- Rozhraní – efektivita rozhraní na externí IT systémy

- Operační systémy – speciální požadavky na OS a jeho případnou úpravu, včetně návrhu další správy
- Obchodní a právní aspekty – licencování a případná právní omezení pro různé demografické regiony

Pro potřeby projektu v semináři Vybrané kapitoly z GIS postačí členění na funkční a nefunkční požadavky.

Hlavní činnosti při analýze požadavků

Identifikace zúčastněných stran (stakeholders)

Identifikace zúčastněných stran (stakeholderů). Zúčastněné strany se neomezují pouze na organizaci, která pořizuje IS. Dalšími zúčastněnými stranami mohou být:

- ty organizace, které se integrují na pořizovaný IS
- jakékoli tzv. back office systémy nebo organizace (které podporují pořizovaný IS)
- vrcholový management

Interview se zúčastněnými stranami

Rozhovory (interview) se zúčastněnými vysvětlují požadavky, které jsou definovány v zadání projektu, a odhalují další požadavky, které v zadání původně nebyly. Požadavky zúčastněných stran mohou být protichůdné, požadavky je tedy nutné konsolidovat. V rámci interview je vhodné identifikovat i ty požadavky nebo vlastnosti systému, které pořizovaný IS nebude mít.

Při zahájení projektu je nutné stanovit, kdo bude mít rozhodující slovo při finalizaci požadavků.

Vytvoření seznamu (katalogu) požadavků

Jedním z tradičního způsobu dokumentování požadavků je seznam požadavků. Tento způsob použijeme i v projektu v semináři Vybrané kapitoly z GIS.

Požadavky zařazené do návrhu systému - tabulka obsahuje požadavky uživatele (klienta) zahrnuté do návrhu systému.

P. č.	Popis požadavku	Typ (F/N)	Zdroj	Kritérium splnění
1				
2				
...				

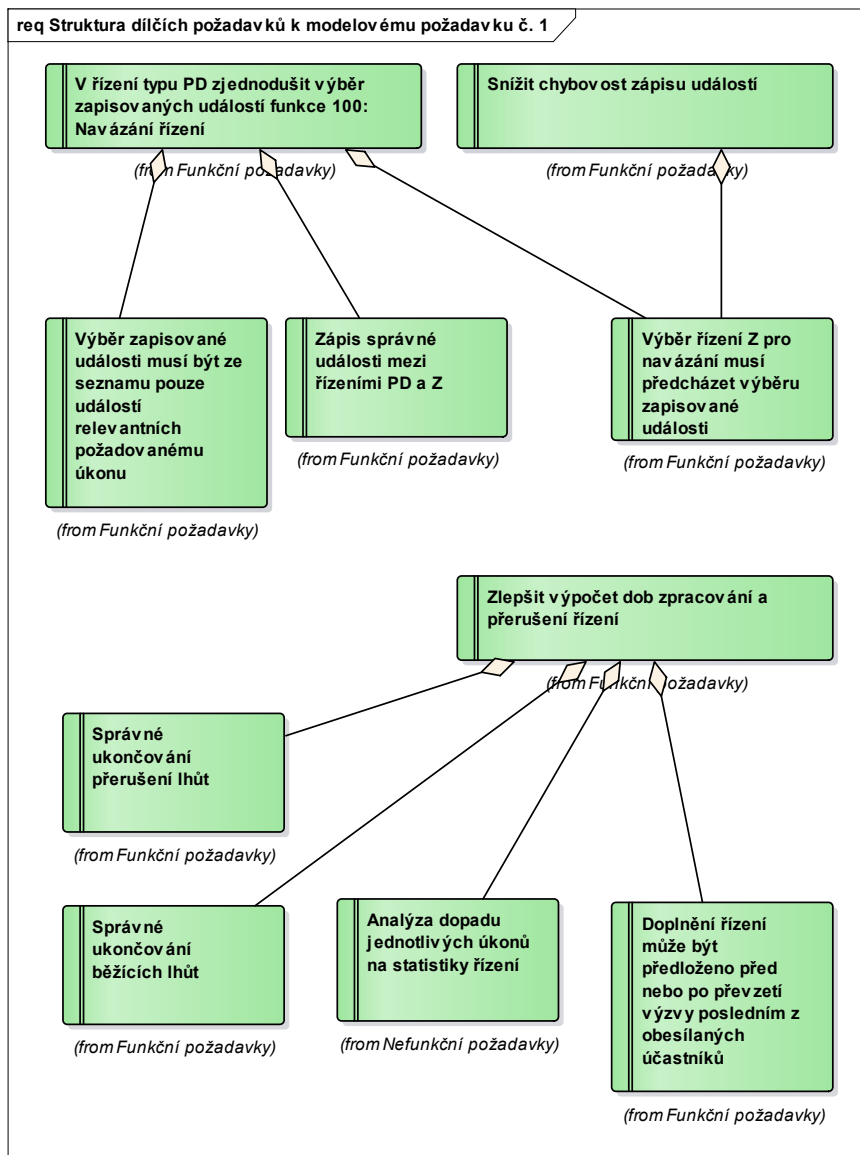
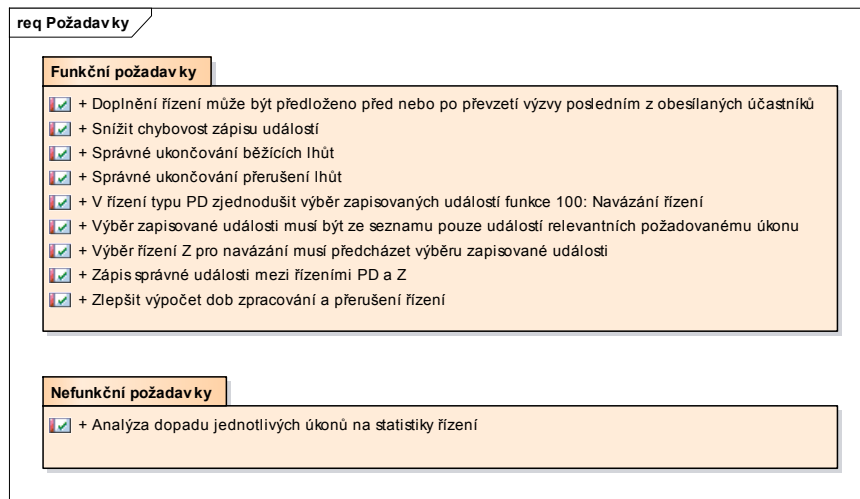
Požadavky nezařazené do návrhu - tabulka obsahuje požadavky uživatelů (klienta), které nejsou zařazené do návrhu systému (pokud takové existují).

P. č.	Požadavek	Zdroj	Zdůvodnění
1			
2			
...			

Vytvoření modelu požadavků

Využití UML prvku „requirement“.

Příklad:



Stanovení měřitelných cílů

Osvědčenými postupy při specifikaci požadavků je klást si otázku, aby bylo možné odhalit skutečný cíl požadavku. Analytici pak můžou navrhnout testy tak, aby bylo možné zjistit, nakolik při plnění každého požadavku bylo dosaženo cíle.

Prototypování

Prototypy jsou reálné modely (mock-ups) aplikací (ukázky, jak bude aplikace vypadat). Prototyp umožňuje uživatelům vizualizovat aplikaci, která dosud nebyla vytvořena. Prototypy pomohou uživatelům získat představu o tom, jak bude systém vypadat, aby uživatelé mohli ověřit design uživatelského interface před finálním vytvořením systému. Zavedením prototypů dochází k významnému zlepšení komunikace mezi uživateli a vývojáři.

Vytváření prototypů aplikace vede k menším pozdějším změnám v aplikaci, a tedy nižším celkovým nákladům. Nicméně, existuje několik problémů spojených s prototypováním:

- Jakmile manažeři vidí prototyp, můžou jen těžko pochopit, že konečný výsledek nebude vyroben ve stejném čase.
- Designéři často používají prototyp v reálném systému, protože se obávají, aby 'neztráceli čas' a nezačínali znovu.
- Prototypy jsou nápomocné hlavně při návrhu uživatelských rozhraní a návrhu obrazovek. Nemohou nic říct o tom, jaký byl původní požadavek.
- Designéři a koncoví uživatelé se soustředí příliš na uživatelské rozhraní a příliš málo na tu část systému, která slouží k podpoře procesů.

Přiřazení požadavků prvkům modelu navrhovaného systému

Pokud je vytvořen model požadavků (například v UML), je vhodné jednotlivé požadavky nebo jejich skupiny v průběhu analýzy přiřazovat prvkům modelu. Typy prvků jsou voleny podle toho, jakého charakteru jsou požadavky. Nejčastěji:

- prvkům typu use case, které vyjadřují funkční specifikaci modelu, jsou přiřazovány požadavky na funkčnost systému,
- prvkům typu class, které reprezentují entity (konceptuálního) datového modelu, jsou přiřazovány požadavky na data systému.

Příklad:

