

4 Technické aspekty softwarového projektu, projektová dokumentace: uživatelská specifikace, technická specifikace a návrh, testování, validace a integrace. Systémy pro správu konfigurace a podporu vývoje. (A4B33SI)

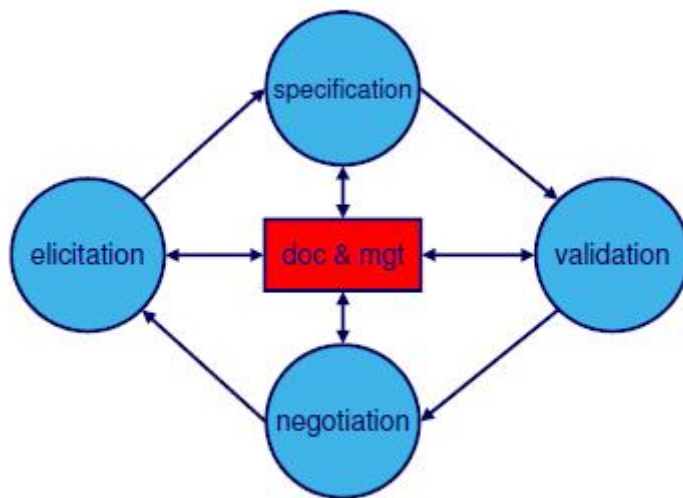
4.1 Technické aspekty softwarového projektu

- řízení projektu - viz otázka 03
- dodržování ISO - spolehlivost, efektivita, užitečnost, udržitelnost, flexibilita...

4.2 Projektová dokumentace

4.2.1 Uživatelská specifikace

- zjištění aktuálního stavu, jak s ním jsou uživatelé spokojeni, co jím vadí, co by potřebovali
- výsledkem je *Specifikace požadavků*
- nutnost:
 - vše správně pochopit (*elicitation*)
 - správně popsat (*specification*)
 - sjednat podstatu problému (*validation*)
 - sjednat hranice problému (*negotiation*)
 - => opakuj, dokud není vše jasné!



4.2.2 Technická specifikace a návrh

- vychází z uživatelské specifikace a přesně stanovuje, co se má udělat
- ! vyvarova se obecnému jazyku => kniha \neq svazek knihy; přítomna \times nepřítomna (ztracena, zapůjčena, zničena,...)
- vypracování všech use-cases - hierarchické uspořádání => rozložení do menších celků
 - ideálně přiřazení konkrétních stakeholderů
- priority: MoSCoW
 - Must have (nejvyšší priorita)
 - Should have (chtěné)
 - Could have (když zbyde čas/peníze)
 - Won't have (dnes ne)
- stanovení funkčních a nefunkčních (rychlost, spolehlivost, jednoduchost,...) požadavků

CMM (Capability Maturity Model)

- na začátku opakovatelný level (analýza, requirements,...) => definitivní level (technické řešení, V+V,...)

SPI (Software Process Increment)

- stanovení hypotézy => sesbírání dat informací => interpretace dat \bigcirc znovu, dokud nemáme vše

Odhad ceny

1. kvantitativní modely
2. kvalitativní modely
3. **COCOMO** (Constructive Cost Model) - dobře dokumentované na výpočty, podle typu a velikosti projektu
4. **FPA** (Function Point Analysis) - stanovení ceny dle struktury, vstupů, funkcí,...

- těžké správně odhadnout

4.2.3 Testování

- nutné testovat, dle metodiky je nejvhodnější testovat už v průběhu vývoje (levnější)
- poměr 2:3 (tester:vývojář)
- testování je měření kvality SW - ISO 9000
 - funkčnost (správnost, spolehlivost), inženýrské řešení (efektivita, dokumentace), adaptibilita (opětovné použítá, údržba)
- kritéria pokrytí testů:
 1. řádky - každý řádek se vykoná alespoň jednou => nedostatečné
 2. větve - každá musí být alespoň 1 pravdivá a 1 nepravdivá
 3. podmínky - zkontroluje všechny možnosti nastalé podmínky a vyhodnotí je (vyžaduje armáda a letectví)
 4. úplné pokrytí cest - v praxi neproveditelné
- definování pomocí grafů:
 - *uzly* = objekty, o které se zajímáme
 - *hrany* = vztah objektů a relací mezi uzly
 - postup:
 - * definuj graf
 - * definuj relace
 - * navrhni testy pro pokrytí uzlů a hran
 - * otestuj a porovnej s očekávanými výsledky
 - * nahvrhni a otestuj testy smyček
- po testování je nutné mít přesné a podrobné specifikace! (jinak se nedá testovat, není co)

- automatizace testování:
 - klady:
 - * častější testování
 - * ověření na nové verzi programu
 - * opakovatelnost testů
 - zápory:
 - * nereálná očekávání
 - * slabší testovací praxe
 - * údržba automatizovaných testů

4.3 Systémy pro správu konfigurace a podporu vývoje

WBS (Work Breakdown Structure) - rozložení projektu na menší struktury

PERT chart (Program Evaluation and Review Technique) - analýza úkolu a rozvržení času na dokončení jednotlivých částí rozložení projektu na menší struktury

GANTT chart - zobrazuje rozložení a stav jednotlivých částí v čase

SVN - synchronizace kódu, verzování,...