**Zadání úlohy:**

V týmu 4 členů splňte následující úkoly:

Definujte SW inženýrství. ①

Uveďte definice a příklady vnější aspekty kvality programu ② (včetně ekonomických ukazatelů ③)

Uveďte vnitřní aspekty kvality programu ②

Aspekty strukturovaného programování: 1) Popište třívrstvou architekturu návrhu 2) Popište význam a výhody modularity systému 3) Co jsou metadata a jak je možné je využít 4) Příklad vlivu formální úpravy ①

Načrtněte graf závislosti „kvality programu“ na době řešení projektu (SW fyzika, Putnamanova rovnice) ④

**Pomůcky**

PC s kancelářským balíkem MS Office a přístupem na internet

**Řešení**

**Žák ①**

SW inženýrství:

Aplikace systematického, normovatelného, měřitelného přístupu (engineering) k vývoji, provozu a údržbě SW.

Systematická činnost, která se podílí na návrhu, implementaci a testování software pro optimalizaci a podporu výroby.

Do SW inženýrství tak patří veškerá „vyčíslitelná“ problematika. Obory návrhu a údržby SW, na které lze uplatnit principy matematiky a fyziky např. mimo jiné i posuzování kvality programů, popis struktury a chyb programů, dokumentace apod.

**Aspekty strukturovaného programování**

Strukturované programování (též strukturovaný programovací jazyk) označuje v [informatice](http://cs.wikipedia.org/wiki/Informatika) [programovací](http://cs.wikipedia.org/wiki/Programov%C3%A1n%C3%AD) techniku, kdy se implementovaný [algoritmus](http://cs.wikipedia.org/wiki/Algoritmus) rozděluje na dílčí úlohy (tj. procedury či [funkce](http://cs.wikipedia.org/wiki/Funkce_%28programov%C3%A1n%C3%AD%29)), které se spojují v jeden celek.

1) návrh shora dolů – přechod z nejvyšší úrovně abstrakce (u IS např. z globální architektury, u tříúrovňové architektury a u databázových systémů konceptuální úroveň apod.) na úroveň konkrétního definování funkcí (logická vrstva) až po zdrojový kód programu (fyzická vrstva)

2) modularita (stavebnicovou strukturu) systému – princip řešení je důležitá součást návrhu (např. předávání parametrů a obsahu proměnných mezi funkcemi či moduly)

výhody modulárních systémů – oddělení uživatelských rozhraní od dat (např. databázové systémy – front-end a back-end)

3) vhodná volba datových a řídících struktur a zacházení s nimi – návrh metadat (METIS), datových struktur (velikost celků i priority jednotlivých operací), řídících struktur (sekvence, větvení, cykly)

4) formální úprava programu (validace) – splnění nefunkčních požadavků programu a to včetně akceptačního testování struktury a chyb programů, dokumentace apod.

**Dokumentace**

 uživatelská – manuály a učebnice

 řešitelská (programátorská) – většinou se netiskne

**Chyby systému**

 syntaktické – formální chyby v příkazech a jejich sestavení

 sémantické – např. chybně sestavené výpočty

 pragmatické (logické) – např. řešení jiné problematiky, než byl požadavek

**Žák ②**

Aspekty kvality programu:

 1) vnější

* věcná správnost
	+ - verifikace – potvrzení správnosti, pravosti; ověření
		- validace – prověření; ověření
* ekonomická efektivnost (ROI)
* robustnost – pevný, mohutný, odolný proti chybám
* dokonalost řešení uživatelského rozhraní (UI)
* estetika (tvoří až 50 % při posuzování SW)
* časová a paměťová optimalizace
* flexibilita (obecný, přizpůsobivý)
* dokonalost (existence) dokumentace

2) vnitřní

* srozumitelnost zdrojového kódu („čistota“ a správnost metadat)
* modifikovatelnost (možnost úprav, změn a rozšíření)
* estetika
* dokonalost (existence) dokumentace

**Žák ③**

ROI je zkratka z anglického *Return On Investments*, tedy **návratnost investic**. Jako ROI (někdy též ROI**index**) označujeme poměr vydělaných peněz k penězům investovaným. ROI tedy udává výnos v procentech z utracené částky.

Nyní již tušíte, že pro základní výpočet ROI stačí dosadit do vzorce:

ROI (%) = výnosy / investice \* 100

**Pro lepší představu výpočtu**ROI**konkrétní příklad:** prodáváte knihy, z prodeje každé máte 100 Kč. Platíte si [PPCreklamu](http://www.adaptic.cz/marketing/ppc-reklama/), jeden návštěvník vás stojí korunu. Web máte velmi dobře zpracovaný, dokážete prodat knihu každému 20. návštěvníkovi – [konverzní poměr](http://www.adaptic.cz/znalosti/slovnicek/konverzni-pomer/) máte 5 %.

**Jak z toho spočítat**ROI**?** Prodej jedné knihy vás stojí 20 Kč (20 návštěvníků po koruně), výnosy jsou 100 Kč. ROItedy bude 100 / 20 = 500 %. Prodej knih se vám vyplácí. Pokud by však ROI vyšel pod 100 % (např. kdybyste prodali jen každému stodesátému návštěvníkovi), už se vám tento způsob prodeje nevyplatí. ROI**nižší než sto procent znamená ztrátu.**

Z výše uvedeného příkladu je zřejmě, že ROI je jedním z **rozhodujících marketingových ukazatelů** [obchodní efektivity webu](http://www.adaptic.cz/znalosti/efektivni-web/obchodni-efektivita-webu/).

Stejný ekonomický princip poměru investice zdrojů (časových, lidských, know-how,…) je možné aplikovat na tvorbu SW.

**Další ukazatele (využitelné v metrikách a ekonomickém hodnocení implementace SW řešení)**

**TCO** – Total Cost of Ownership – Celková cena vlastnictví – metodika (firmy Gartner Group) popisující proces získání ukazatele zahrnujícího veškeré náklady vlastnictví za určité období v příslušné měně.

TCO zahrnuje náklady na:

* nákup a vlastnictví
* provoz a používání systému nebo jeho částí ve všech fázích životního cyklu

Jako reakce na výtky, že ukazatel TCO preferuje investice do IS nebo informační systémy s nejnižšími náklady bez ohledu na přínosy, vznikl ukazatel **TVO** – Total Value of Ownership (celková hodnota vlastnictví).

Od celkových přínosů informačního systému označovaných jako **TBO** – Total Benefits of Ownership (celkové přínosy vlastnictví) se odečte hodnota ukazatele TCO.

TVO = TBO – TCO

Ukazatel TVO částečně odstraňuje některé slabiny ukazatele TCO – především bere v úvahu přínosy informačního systému. Ponechává si všechny ostatní slabiny TCO a přidává k nim ještě mnohdy nejasné a nepřesné vyjádření přínosů informačního systému (TBO), které se často velmi těžko finančně oceňují.

Řešením je využití ukazatele **TVOp** – Total Value of Opportunity (celkové náklady příležitosti). Tento ukazatel vychází z obecně přijímaného ekonomického pojmu Opportunity Cost (cena příležitosti – Při hospodářské činnosti vynakládáme vzácné a omezené zdroje, abychom dosáhli zvoleného cíle, současně tím ztrácíme příležitost je vynaložit na dosažení odlišného cíle. **Abychom získali toto, vzdali jsme se jiného**.).

**Žák ④**

**Softwarová fyzika**

Vychází z empirických znalostí.

Např. závislost kvality programu na době řešení projektu:

Putnamova rovnice:

$$N≅cT^{\frac{1}{3}}D^{\frac{4}{3}}$$

Kde:

* + - * N – délka programu (počet řádků, SLOC – source lines of code)
			* T – spotřeba práce (s – průměrný počet řešitelů, MM – men month; MD – Menday)
			* D – doba realizace

Grafické vyjádření závislosti pracnosti na čase projektu

**D**

**C**

**B**

**A**

N
(SLOC)

T
(MD)

A – neřešitelná oblast – i při velkém množství pracovníků je projekt v daném čase neřešitelný

B – krátký termín – projekt je splnitelný, ale kód obsahuje řadu chyb a je většinou neoptimalizovaný (příliš komplikovaný a dlouhý)

C – optimální časové rozložení – stabilní projekt

D – dlouhý termín – projekt se v čase „ztrácí“ – pro tvůrce není aktuální a postupně se již vytvořené části stávají zastaralými

Př. dopadu Putnamovy rovnice – při zkrácení termínu na 83 % je pracnost projektu dvojnásobná.