

14 Úvod do plánování projektu



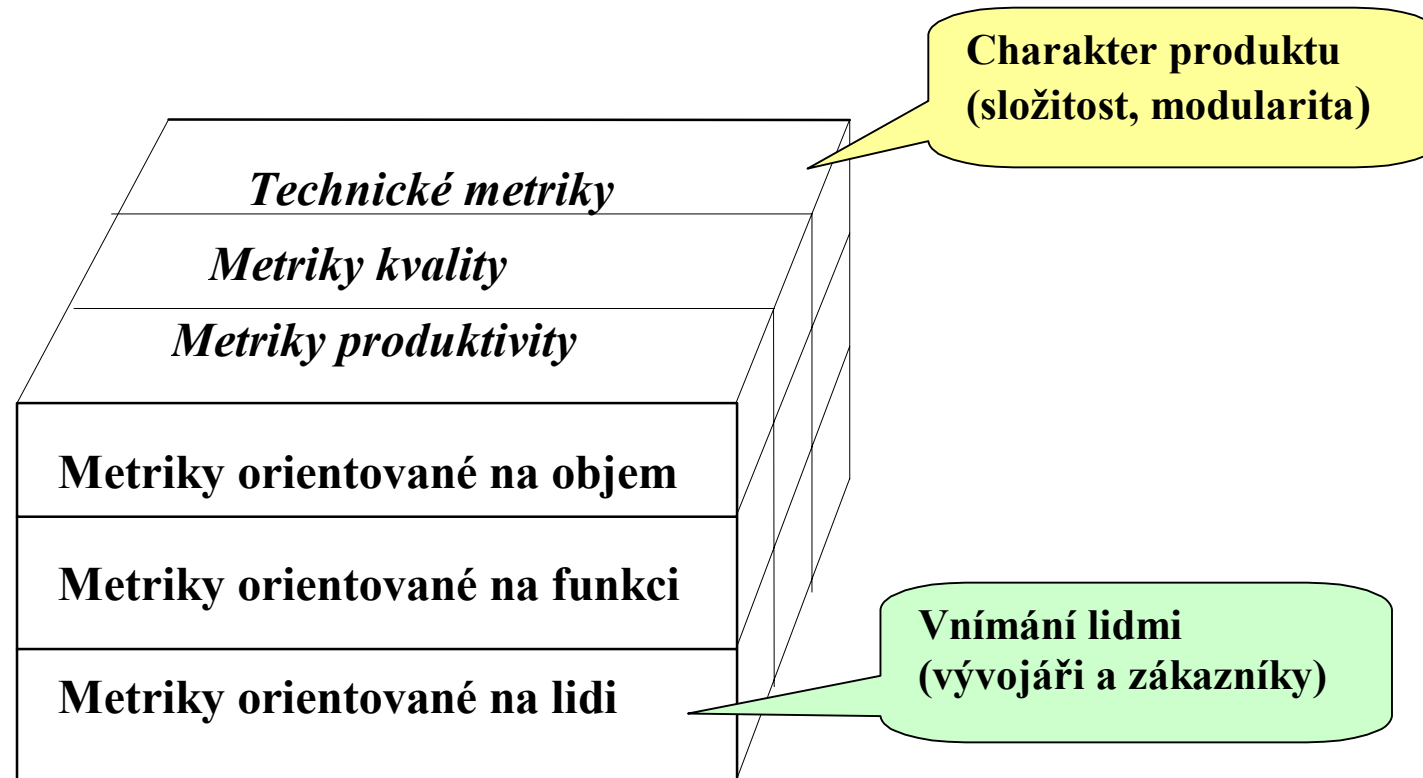
- rozbor zadání → odhad pracnosti, doby řešení, nákladů, ... → analýza rizik → strategie řešení → organizace týmu
→ **PLÁN PROJEKTU**

14.1 Softwarové metriky

SW metriky - soustava veličin používaných ke kvantitativnímu ohodnocení (hlavně produktivity a kvality) procesu vývoje SW a samotného produktu.

- Proč měříme?
 - hodnocení kvality produktu,
 - hodnocení produktivity tvůrců SW,
 - hodnocení přínosů nových nástrojů a metod,
 - vytvoření základu pro odhady,
 - zdůvodnění požadavků na nové nástroje, vybavení, ...

měření $\left\{ \begin{array}{l} \text{přímé (náklady, pracnost, doba, velikost kódu-LOC,} \\ \text{rychlost, počet chyb, ...)} \\ \text{nepřímé (funkčnost, kvalita, složitost, efektivnost,} \\ \text{spolehlivost, udržitelnost, ...)} \end{array} \right.$



- **Metriky orientované na objem**
- přímé míry

Název projektu	Pracnost [člměs.]	Náklady [Kč]	KLOC	Stran dokumentace	Chyb	Lidí

→ lze vytvořit řadu základních metrik produktivity a kvality:

Průměry: Produktivita = $KLOC/pracnost$

Kvalita = $chyb/KLOC$

Další např: $Náklady_na_LOC = Náklady/LOC$

$Dokumentace = Stran_dokumentace/KLOC$

- ⊕ - LOC je výsledkem SW projektů, snadno měřitelný, řada modelů pro odhady ho používá jako vstup
- ⊖ - závislost na jazyku, nevhodné pro neprocedurální jazyky, při odhadech neznáme předem velikost

- **Metriky orientované na funkci**
 - nepřímé, zaměřují se na funkčnost nebo užitečnost programu
 - metoda funkčních bodů - FP (Albrecht 1979) - vychází z empirického vztahu mezi počitatelnými veličinami a ohodnocením složitosti

Parametr	Počet		Váha				
			Jedn.	Prům.	Slož.		
Uživatelské vstupy		x	3	4	6	=	
Uživatelské výstupy		x	4	5	7	=	
Uživatelské dotazy		x	3	4	6	=	
Soubory		x	7	10	15	=	
Externí rozhraní		x	5	7	10	=	
Celkem	→						

→ výpočet funkčních bodů:

$$FP = \text{Celkem} * [0.65 + 0.01 * \text{SUM}(F_i)]$$

korekce dle složitosti (14 otázek ohodnocených 0 až 5)

- metoda FP navržena pro aplikace obchodních inf.systémů, příp. systémové a VT aplikace

1986 - rozšíření zohledněním algoritmické složitosti,

→ vhodné i pro RT a vestavěný a řídicí SW

- ⊕ - nezávislost na jazyku, vhodnější pro odhady
- ⊖ - nemá fyzikální význam, vyžaduje zkušenost

- **Metriky kvality**

- měřena během procesu (složitost programu, modularita, ...)
- měřena po předání uživateli (počet chyb, udržitelnost, ...)

- faktory (viz McCall):

- činnost produktu (jak funguje),
- revize produktu (jak lze provádět změny),
- převod produktu (jak lze přenášet)

- nejrozšířenější metriky:

Správnost - míra souladu s požadovanou funkcí

$N_{def}/KLOC$

Udržitelnost - snadnost provádění oprav, modifikací, vylepšení

MTTC(Mean-Time-To-Change) - včetně analýzy, návrhu, ...

Integrita - schopnost odolat útoku na bezpečnost systému.

T (Thread) - pravděpodobnost výskytu útoku

S (Security) - pravděpodobnost, že útok bude odvrácen

Integrita = $\Sigma [1 - T*(1 - S)]$

Použitelnost - míra přátelskosti programu.

4 charakteristiky:

- zručnost potřebná pro zvládnutí,
- čas potřebný k dosažení středně efektivního používání

- zvýšení produktivity,
- subjektivní hodnocení

- vztah metrik orientovaných na objem a na funkci
 - hrubý odhad:

Jazyk	LOC/FP (průměr)
Asembler	320
C	128
COBOL	105
FORTRAN	105
Pascal	90
Ada	70
OO jazyky	30
4GL	20
generátory kódu	15
Tab.procesory	6
graf.jazyky	4

Lze použít produktivitu z LOC (a FP) pro porovnávání? NE

Faktory produktivity:

1. Lidé
2. Složitost
3. Proces
4. Produkt
5. Zdroje

14.2 Odhad SW projektu

- pracnost, náklady, doba řešení, počet lidí ...
- Jak provést odhad?
 1. Odložit na pozdější dobu
 2. Použít dekompozičních technik
 3. Použít empirický model
 4. Použít dostupný nástroj pro automatizovaný odhad
 - riziko odhadu → kombinace metod, násobení „koeficientem“
- Dekompoziční techniky
 - Princip: Dekompozice na hlavní funkce a jim odpovídající úlohy a postupný odhad.
 - Varianty:
 - a) odhad LOC nebo FP
 1. Dekompozice na dílčí funkce
 2. Odhad LOC nebo FP pro jednotlivé funkce
 - optimistický, pravděpodobný, pesimistický → vážený průměr
 3. Použití metrik produktivity z předchozích projektů

Př) CAD aplikace

- na pracovní stanici, interakce s grafickými zařízeními
- vymezení základních funkcí:
 - uživatelské rozhraní (UI),
 - 2D analýza (2D),
 - 3D analýza (3D),
 - řízení databáze (DB),
 - grafické zobrazení (GD),
 - řízení periferií (PC),
 - moduly pro analýzu (AM).

Funkce	LOC				\$/ř.	ř./měs.	Náklady	Měsíce
	Opt.	Prav.	Pes.	Oček.				
UI	1800	2400	2650	2340	14	315	32760	7.4
2D	4100	5200	7400	5380	20	220	107600	24.4
3D	4600	6900	8600	6800	20	220	136000	30.9
DB	2950	3400	3600	3350	18	240	60300	13.9
GD	4050	4900	6200	4950	22	200	108900	24.9
PC	2000	2100	2450	2140	28	140	59920	15.2
AM	6600	8500	9800	8400	18	300	151200	28.0
Celkem				33360			656680	144.5

b) Odhad pracnosti

1. Dekompozice na dílčí funkce

2. Odhad pracnosti jednotlivých etap vývoje pro jednotlivé funkce

3. Stanovení nákladů na pracovní sílu pro jednotlivé etapy

4. Stanovení celkové pracnosti a nákladů

Funkce	Etapa				Celkem
	Analýza	Návrh	Kódování	Testování	
UI	1.0	2.0	0.5	3.5	7.0
2D	2.0	10.0	4.5	9.5	26.0
3D	2.5	12.0	6.0	11.0	31.5
DB	2.0	6.0	3.0	4.0	15.0
GD	1.5	11.0	4.0	10.5	27.0
PC	1.5	6.0	3.5	5.0	16.0
AM	4.0	14.0	5.0	7.0	30.0
Celkem	14.5	61.0	26.5	50.5	152.5
Sazba [\$]	5200	4800	4250	4500	
Náklady [\$]	75400	292800	112625	227250	708075

- Empirické modely
 - empiricky odvozené formule pro odhad pracnosti, doby řešení, ...

Př) Model COCOMO (Constructive Cost Model) - Boehm 1981

$$E = f_1(LOC), D(\text{doba řešení}) = f_2(E),$$

- empirické koeficienty + cenové faktory (produktu, HW, řešitelů, projektu)

14.3 Analýza rizik

- Identifikace rizika
 - rizika projektu (rozpočet, doba, ...),
 - požadavků, technologická, dovednostní,
 - obchodní a „politická“ rizika

Metoda: např. sada otázek (kontrolní seznam rizikových faktorů)

- Projekce rizika a opatření k zabránění
 - pravděpodobnost, důsledky, opatření

Oblast zájmu - rizika se střední a vysokou pravděpodobností

Př) [vysoká fluktuace, 0.7, zvýšení nákladů o 12% a trvání o 15%]

Opatření: zjištění příčin, zajištění kontinuity, organizace týmu, dokumentace, ...

14.4 Organizace vývojových týmů

- otázka komunikace, rozhraní,...

a) nestrukturované týmy – rozdělení podle objemu

Př) osamělí vlci, horda, demokratická skupina

b) strukturované týmy – rozdělení podle profese

Př) chirurgický, hlavního programátora, ..., vícetýmová organizace