

# Funkční body

## Základní úvaha



Výroba software je v podstatě výrobní proces, který vyžaduje lidskou práci.

Takže je to prosté, stačí určit:

- jednotku výroby
- cenu práce za výrobu této jednotky

# Odhady pomocí funkčních bodů



Funkční body = normalizovaná metrika softwarového projektu

- Měří aplikační oblast, nezkoumá technickou oblast
- Měří aplikační funkce a data, neměří kód

International Function Point Users Group - [www.ifpug.org](http://www.ifpug.org)

Literatura:

Capers Jones: *Applied Software Measurement* (1997)

*Estimating Software Costs* (1998)

## Funkční body - princip



- Předběžný odhad s použitím omezené informace
- Měří vstupy, výstupy, dotazy, vnitřní paměti, vnější paměti

Princip odhadu:

- velikost projektu X složitost X rizikové faktory = odhad

# Typy funkčních bodů



Funkční body vztažené k transakčním funkcím:

- Externí vstupy (EI - External Inputs)
- Externí výstupy (EO - External Outputs)
- Externí dotazy (EQ - External Enquiry)

Funkční body vztažené k datovým funkcím:

- Vnitřní logické soubory (ILF - Internal Logical Files)
- Soubory vnějšího rozhraní (EIF - External Interface Files)

## Interní logické soubory (ILF)



Každá velká logická skupina uživatelských dat nebo informací použitých pro řízení aplikace představuje jeden ILF.

Zahrneme každý logický soubor, nebo v případě DB, každé logické seskupení dat z pohledu uživatele, které je vytvořeno, používáno, nebo udržováno aplikací.

Spíše než fyzické soubory započteme každé logické seskupení dat tak, jak je viděno z pohledu uživatele a jak je definováno při analýze požadavků nebo návrhu dat.

Nezapočteme soubory, které nejsou přístupné uživateli prostřednictvím vnějšího výstupu nebo dotazu a které nejsou nezávisle udržovány.

## Interní logické soubory (ILF)



1. Logická entita nebo skupina entit z pohledu uživatele. (1 ILF)
2. Logický interní soubor generovaný nebo udržovaný aplikací. (1 ILF)
3. Uživatelem udržovaná tabulka(y) nebo soubor(y). (1 ILF)
4. Datový soubor nebo soubor s řídicí informací, který aplikace použije při sekvenčním zpracování a údržbě. (1 ILF)
5. Atributová entita udržovaná pouze prostřednictvím hlavní entity. (0 ILF)
6. Asociativní entity vytvořené průnikem nebo spojením obsahující pouze klíčový atribut (0 ILF)
7. Přechodný nebo třídící soubor (dočasný soubor) (0 ILF)
8. Soubor vytvořený proto, že byla použita určitá technologie (např. indexový soubor) (0 ILF)
9. Soubor s předlohou (vzorem), který aplikace pouze čte. (0 ILF, 1 EIF)

## Soubory externího rozhraní (EIF)



Započteme každou velkou logickou skupinu uživatelských dat nebo řídicí informace používané aplikací.

Tato informace musí být udržována jinou aplikací. Zahrňte každý logický soubor nebo logickou skupinu dat z pohledu uživatele.

Započteme každou velkou logickou skupinu uživatelských dat nebo řídicí informace, která je extrahována aplikací z jiné aplikace ve formě souboru externího rozhraní.

Extrakce nemá mít za následek změnu v některém z interních logických souborů. Pokud ano, pak započteme do EI místo do EIF.



## Soubory externího rozhraní (EIF)



1. Soubory nebo záznamy extrahované z jiné aplikace (použité pouze jako odkazy) (1 EIF)
2. Databáze čtená pomocí jiné aplikace (1 EIF)
3. Vnitřní logický soubor jiné aplikace použitý jako transakce (0 EIF, 1 EI)
4. Systém HELP, bezpečnostní soubor, chybový soubor čtený nebo odkazovaný aplikací, který pochází z jiné aplikace, která soubory udržuje (2 EIF)

## Externí vstupy (EI)



Započteme každá unikátní uživatelská data nebo zadání uživatelských povelů, která vstoupí přes externí rozhraní do aplikace a přidá, mění, ruší nebo jinak pozmění data (např. přiřazení, přemístění, ...) v interním logickém souboru.

Započteme také řídicí informaci, která vstoupí přes aplikační hranici a zajistí soulad s funkcí specifikovanou uživatelem.

Externí vstup by měl být považován za unikátní, pokud logický návrh vyžaduje logiku zpracování odlišnou od ostatních externích vstupů.

## Externí vstupy (EI)



1. Datová obrazovka s přidáním, změnou a rušením (3 EI)
2. Více obrazovek pohromadě zpracovaných jako jedna transakce (1 EI)
3. Dvě datové obrazovky s odlišným uspořádáním dat, ale se shodnou logikou zpracování (1 EI)
4. Dvě datové obrazovky se shodným formátem, ale s odlišnou logikou zpracování (2 EI)
5. Datová obrazovka s více unikátními funkcemi (1 EI za každou funkci)
6. Automatický vstup dat nebo transakcí z jiné aplikace (1 EI na každý typ transakce)

## Externí vstupy (EI)



7. Vstup uživatelských povelů do aplikace (1 EI)
8. Vstupní formuláře (OCR) s jednou transakcí (1 EI)
9. Funkce úpravy dat, která následuje za dotazem (1 EI a 1 EQ)
10. Individuální výběry na obrazovce s menu (0 EI)
11. Oprava uživatelem udržované tabulky nebo souboru (1 EI)
12. Duplikát obrazovky, která již byla započtena jako vstup (0 EI)
13. Externí vstupy zavedené pouze kvůli technologii (0 EI)
14. Výběr položky ze seznamu (0 EI)

## Externí výstupy (EO)



Započteme každá unikátní uživatelská data nebo řídicí data, která opouští externí hranici měřeného systému.

Externí výstup je považován za unikátní, pokud má odlišná data, nebo pokud vnější návrh (jiná aplikace) vyžaduje odlišnou logiku zpracování oproti jiným externím výstupům.

Externí výstupy se často skládají z hlášení, výstupních souborů zasílaných jiné aplikaci nebo zpráv pro uživatele.

## Externí výstupy (EO)



1. Výstup dat na obrazovku (1 EO)
2. Souhrnná zpráva - dávkové zpracování (1 EO)
3. Automatická data nebo transakce směrem k jiným aplikacím (1 EO)
4. Chybové zprávy vrácené jako výsledek vstupní transakce (0 EO)
5. Záložní soubory (0 EO)
6. Výstup na obrazovku a na tiskárnu (2 EO)
7. Výstupní soubory vytvořené z technických důvodů (0 EO)
8. Výstup sloupcového a zároveň koláčového grafu (2 EO)
9. Dotaz s vypočtenou informací (1 EO, 0 EQ)

## Externí dotazy (EQ)



Jako vnější dotaz započti každou unikátní vstupně/výstupní kombinaci, kde vstup je příčinou a generuje výstup.

Vnější dotaz je považován za unikátní, pokud se od ostatních dotazů odlišuje typem výstupních datových elementů, nebo pokud vyžaduje odlišnou logiku zpracování v porovnání s ostatními externími dotazy.

## Externí dotazy (EQ)



1. On-line vstup a on-line výstup beze změny v datových souborech (1 EQ)
2. Dotaz následovaný změnovým vstupem (1 EQ/1 EI)
3. Vstup a výstup na obrazovce s nápovědou (na dané úrovni) (1 EQ)
4. On-line vstup s bezprostředním tiskem dat beze změny dat (1 EQ)
5. Výběr ze seznamu nebo umístění s dynamickými daty (1 EQ)
6. Výběr ze seznamu nebo umístění se statickými daty (0 EQ)
7. Požadavek na zprávu obsahující neodvozená data (1 EQ)



## Výpočet funkčních bodů



Před výpočtem musíme EI, EO, EQ, ILF, EIF rozřídít do skupin podle vah.

Váhy	nížká	průměrná	vysoká	celkem
EI	___ x 3 +	___ x 4 +	___ x 6 =	_____
EO	___ x 4 +	___ x 5 +	___ x 7 =	_____
EQ	___ x 3 +	___ x 4 +	___ x 6 =	_____
ILF	___ x 7 +	___ x 10 +	___ x 15 =	_____
EIF	___ x 5 +	___ x 7 +	___ x 10 =	_____

Neupravené funkční body celkem \_\_\_\_\_

## Matice složitosti vstupů (EI, EQ)



FTRs	1-4 DETs	5-15 DETs	16+DETs
0-1	nízká	nízká	průměrná
2-3	nízká	průměrná	vysoká
4+	průměrná	vysoká	vysoká

- FTR = File Types (User Data Groups) Referenced
- DET = Data Element Type (Attribute)
- RET = Record Element Type (User View)

## Matice složitosti výstupů (EO, EQ)



FTRs	1-4 DETs	5-15 DETs	16+DETs
0-1	nízká	nízká	průměrná
2-3	nízká	průměrná	vysoká
4+	průměrná	vysoká	vysoká

- FTR = File Types (User Data Groups) Referenced
- DET = Data Element Type (Attribute)
- RET = Record Element Type (User View)

## Matrice složitosti souborů (ILF, EIF)



RETs	1-19 DETs	20-50 DETs	51+ DETs
1	nízká	nízká	průměrná
2-4	nízká	průměrná	vysoká
5+	průměrná	vysoká	vysoká

- FTR = File Types (User Data Groups) Referenced
- DET = Data Element Type (Attribute)
- RET = Record Element Type (User View)



14 charakteristik hodnocených podle stupně vlivu na aplikaci  
Každý faktor je hodnocený ve stupnici 0 – 5 takto:

- 0 = bez vlivu
- 1 = náhodný
- 2 = mírný
- 3 = průměrný
- 4 = významný
- 5 = podstatný

## Obecné charakteristiky systému - faktory



1. Vyžaduje systém spolehlivé zálohování a zotavení?
2. Jsou vyžadovány datové komunikace?
3. Existuje distribuované zpracování?
4. Je výkonnost kritická?
5. Poběží systém v stávajícím intenzivně využívaném operačním prostředí?
6. Systém požaduje on-line vstup dat?
7. Vyžaduje on-line vstup dat použití vstupní transakce přes více obrazovek nebo operací?

## Obecné charakteristiky systému - faktory



8. Jsou hlavní soubory opravovány on-line?
9. Jsou vstupy, výstupy, soubory a dotazy složité?
10. Je vnitřní zpracování složité?
11. Je kód navrhován s cílem znovupoužití?
12. Jsou konverze a instalace zahrnuty v návrhu?
13. Je systém navrhován pro násobné instalace u různých organizací?
14. Je aplikace navrhovaná tak, aby zajistila změny a snadné používání na straně uživatele?

## Počet funkčních bodů



Počet funkčních bodů

=

[0.65 + (0.01 x *součet hodnocení charakteristik systému*)]

x

[*počet nepřizpůsobených funkčních bodů*]



# Nové a upravované projekty



Type of Project	Project Function Points	Application Function Points
		Installed Function Pts. (IFP)
Development Project	Project FP = New (Added) FP + Conversion FP	Application FP = New (Added) FP
Enhancement Project	Project FP = Added FP + Changed FP + Deleted FP + Conversion FP	Application FP = Original FP - Deleted FP + Added FP + $\Delta$ Changed FP

## Postup výpočtu FP



1. Identifikujte a spočtete ILF, EIF, EI, EO, EQ. Pro každou ILF a EIF identifikujte počet RET a počet DET. Pro každou EI, EO a EQ, identifikujte počet FTR a DET
2. S použitím matice složitosti spočtete počet jednoduchých, průměrných a složitých položek EI, EO, EQ, ILF, EIF.
3. Spočtete *Počet neupravených funkčních bodů*.
4. Určete hodnoty 14 charakteristik systému.
5. Sečtete charakteristiky a určete Faktor technické složitosti systému.
6. Určete *Počet upravených FP systému*.

# Odhady velikosti (Capers Jones)



1 Funkční bod = X příkazů

- 320 - základní assembler
- 213 - makro assembler
- 128 - C
- 107 - COBOL
- 107 - FORTRAN
- 80 - PL/I
- 71 - Ada 83
- 64 - C++
- 54 - Ada 95
- 32 - Visual BASIC
- 22 - Smalltalk
- 16 - PowerBuilder
- 13 - SQL

## Další odhady



- FP 1.15 předpovídá přibližný počet stran papírové dokumentace SW projektu
- FP 1.2 předpovídá přibližný počet vytvořených testovacích případů
- FP 1.25 předpovídá přibližný chybový potenciál u nových SW projektů
- FP 0.4 předpovídá přibližný plán vývoje v kalendářních měsících
- FP / 150 předpovídá přibližný počet pracovníků potřebných pro řešení aplikace
- FP / 750 předpovídá přibližný počet pracovníků údržby, kteří budou udržovat aplikaci v aktuálně požadovaném stavu

## Přibližné velikosti aplikací



- Vstup objednávky 1,250 FP
- Zpracování daňového přiznání 2,000 FP
- Účtování telefonních služeb 11,000 FP
- Rezervace letenek 25,000 FP
- OS Windows 95 85,000 FP
- Telefonní přepojovací systém 12,000 FP

## Produktivita (FP a člověkoměsíc)



Nezkušený tým, nestrukturované metody, běžné nástroje, jazyky na nízké úrovni	2.50
Nezkušený tým, nestrukturované metody, nástroje CASE, jazyky na nízké úrovni	3.50
Nezkušený tým, strukturované metody, běžné nástroje, jazyky na nízké úrovni	4.00
Zkušený tým, nestrukturované metody, běžné metody, jazyky na nízké úrovni	4.50
Nezkušený tým, nestrukturované metody, běžné nástroje, jazyky na vysoké úrovni	5.00
Nezkušený tým, strukturované metody, nástroje CASE, jazyky na nízké úrovni	6.00
Nezkušený tým, nestrukturované metody, nástroje CASE, jazyky na vysoké úrovni	7.00
Zkušený tým, nestrukturované metody, nástroje CASE, jazyky na nízké úrovni	8.00
Nezkušený tým, strukturované metody, běžné nástroje, jazyky na vysoké úrovni	8.50
Zkušený tým, strukturované metody, běžné nástroje, jazyky na nízké úrovni	9.00
Zkušený tým, nestrukturované metody, běžné nástroje, jazyky na vysoké úrovni	10.00
Zkušený tým, strukturované metody, nástroje CASE, jazyky na nízké úrovni	12.00
Nezkušený tým, strukturované metody, nástroje CASE, jazyky na vysoké úrovni	14.00
Zkušený tým, nestrukturované metody, nástroje CASE, jazyky na vysoké úrovni	18.00
Zkušený tým, strukturované metody, běžné nástroje, jazyky na vysoké úrovni	25.00
Zkušený tým, strukturované metody, nástroje CASE, jazyky na vysoké úrovni	40.00

Zdroj: Jones: *Estimating Software Costs*

## Příklad konkrétních měření

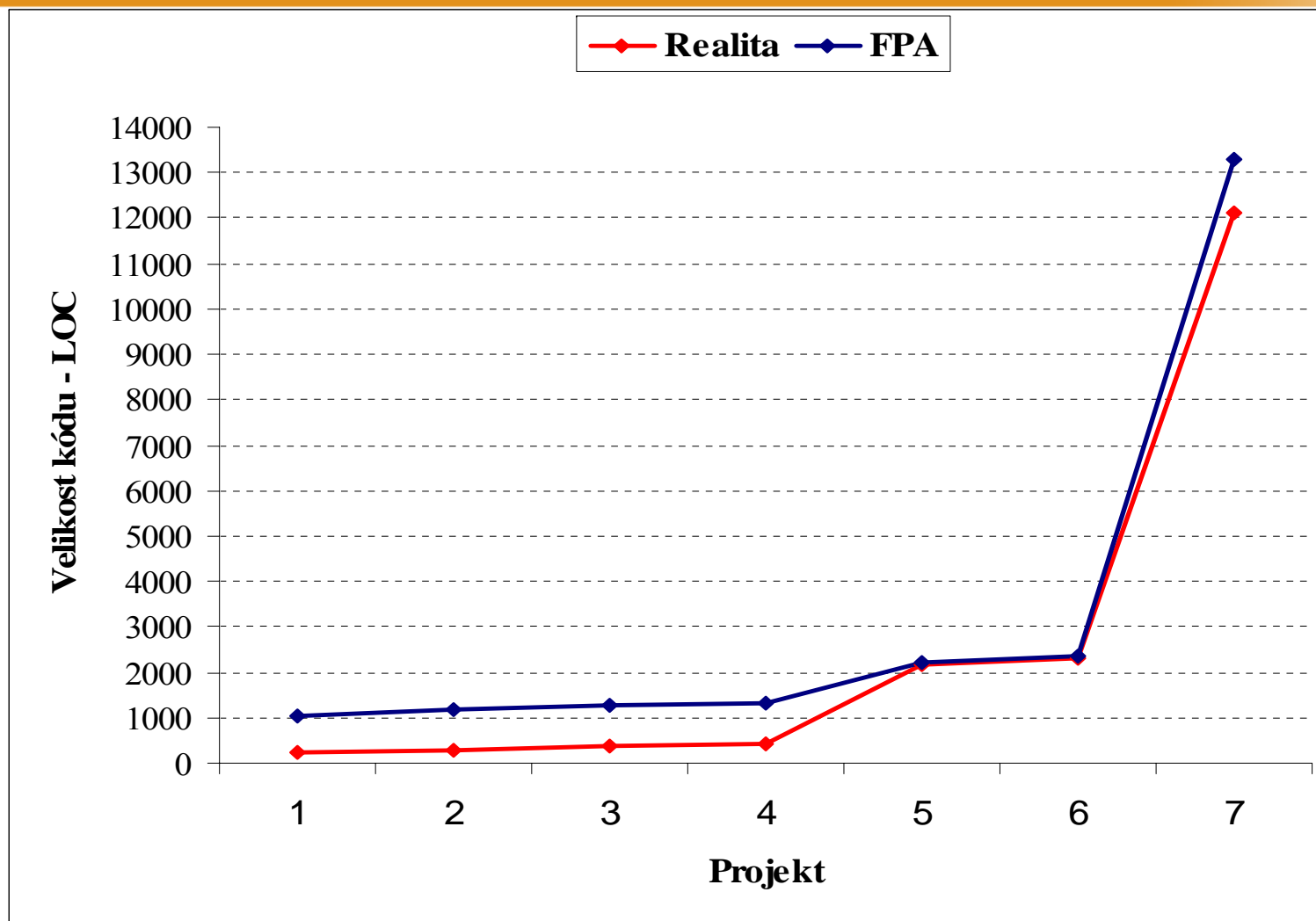


V roce 2005 bylo na FI změřeno 7 studentských projektů (diplomových, bakalářských a seminárních prací) v jazycích C, C++, Java a Delphi.

Přepočtem naměřených funkčních bodů byl proveden odhad velikosti zdrojového kódu, který byl porovnán se skutečností.

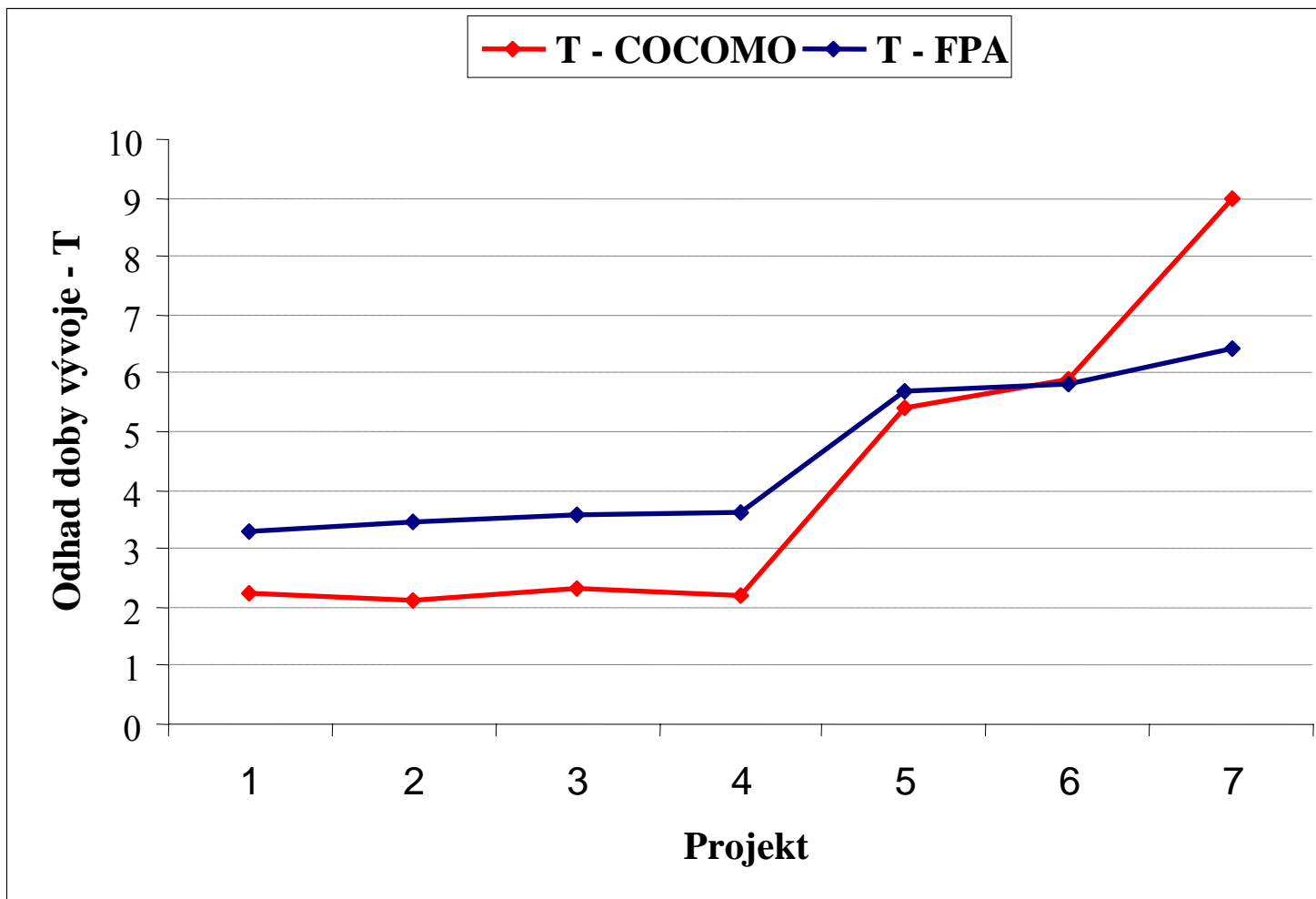
Doba potřebná na řešení projektu byla odhadnuta pomocí COCOMO i FPA. Výsledky byly porovnány.

# Porovnání odhadnuté velikosti kódu

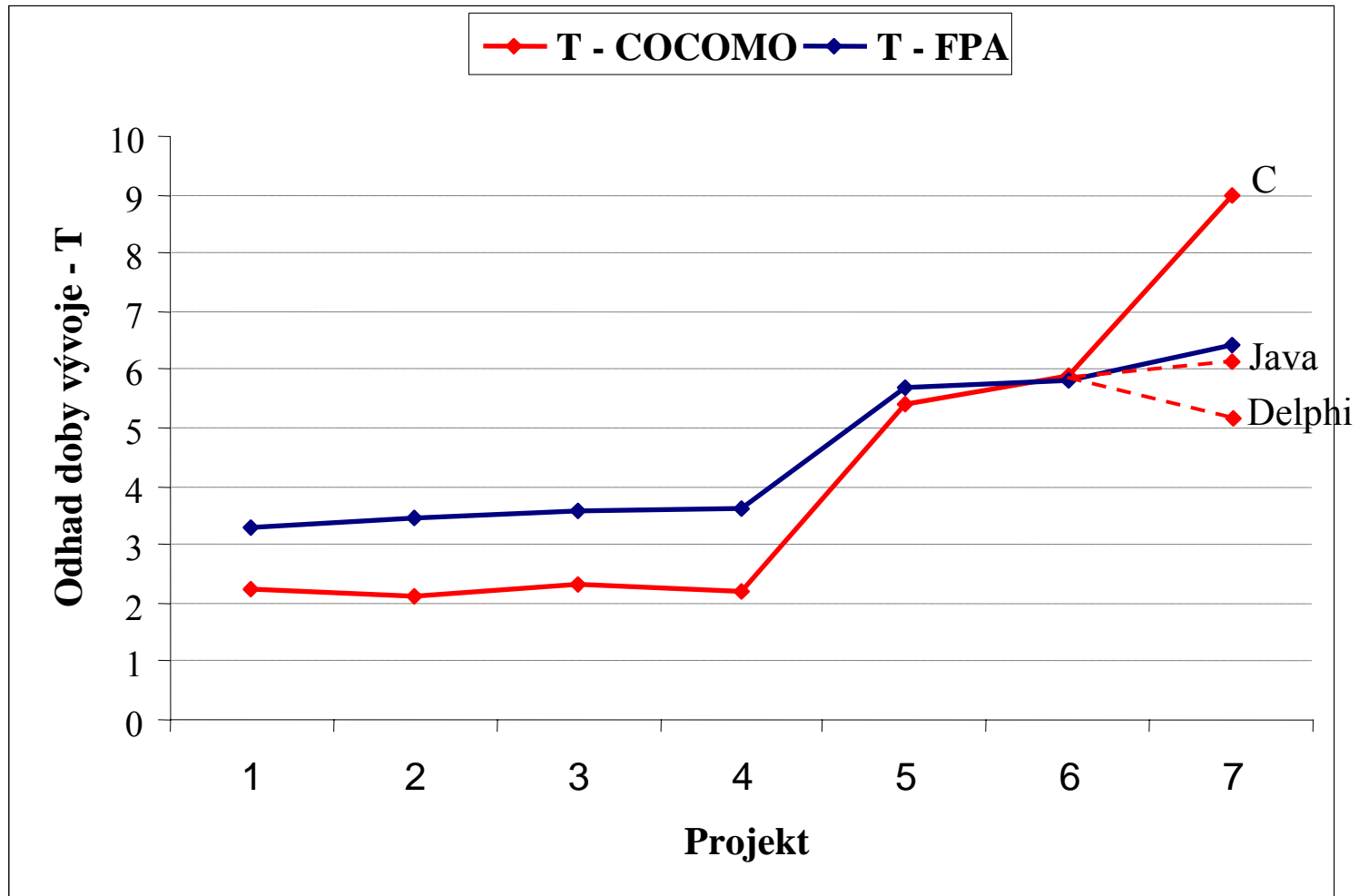




# Porovnání odhadnuté doby projektu



# COCOMO a použití jiného jazyka u projektu č.7



## Příklad 1 - Banka



Umožňuje přidávat nové zákazníky a rušit zákazníky v kartotéce. Systém podporuje transakce vkladu a výběru, při výběru kontroluje překročení povoleného úvěru.

Při překročení zobrazí varovnou zprávu.

Zákazníci se mohou dotázat na stav účtu pomocí terminálu.

Bankéř si může vyžádat seznam zákazníků, kteří přečerpali účet.

Úkol:

- Ohodnoťte jednotlivé funkce podle dále uvedeného seznamu a stanovte funkční body.

## Příklad 1 - Banka



- **Externí vstupy:** přidej nového zákazníka, zruš zákazníka, vklady, výběry, požadavek na seznam zákazníků, kteří přečerpali účet
- **Externí výstupy:** varování o významném přečerpání, seznam zákazníků s přečerpaným účtem
- **Externí dotazy:** dotazy na stav účtu
- **Interní soubory:** soubor Zákazníci

# Příklad 1 - Banka



funkce	# souborů	# typů záznamů	#dat. elem.	složitost	skóre
přidej zákazníka					
zruš zákazníka					
vklad					
výběr					
požadavek zprávy o přečerpaných účtech					
varovná zpráva					
zpráva o přečerpání					
dotaz na stav účtu					
soubor <b>Zákazníci</b>					

## Příklad 2 - Kurzy a školení

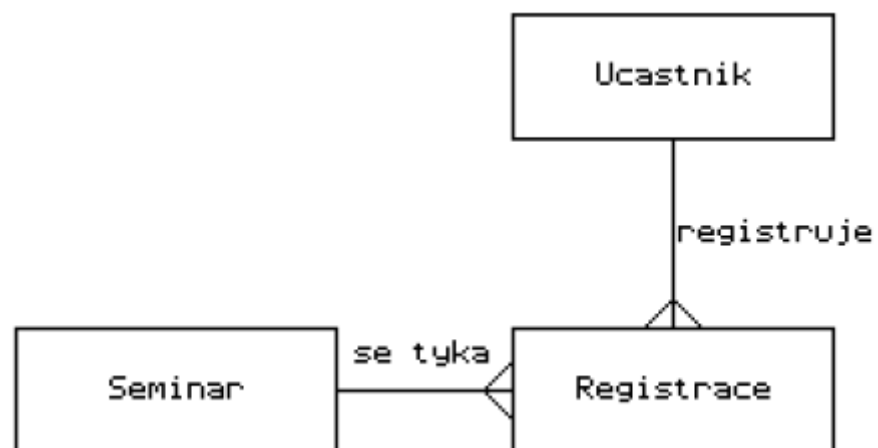


Firma pořádá kurzy a školení. Systém eviduje základní údaje o kurzech, lektorech, posluchačích a místech konání.

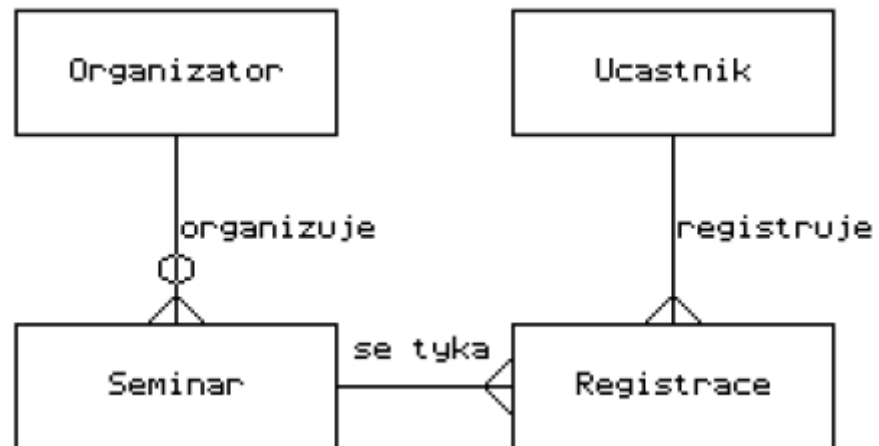
Úkol:

- Pro následující verze datových modelů odhadněte požadované funkce a stanovte jejich klasifikaci podle FP.
- Stanovte funkční body. U poslední verze stanovte faktor technické složitosti a vypočtete upravené FP.

## Příklad 2 - Kurzy a školení

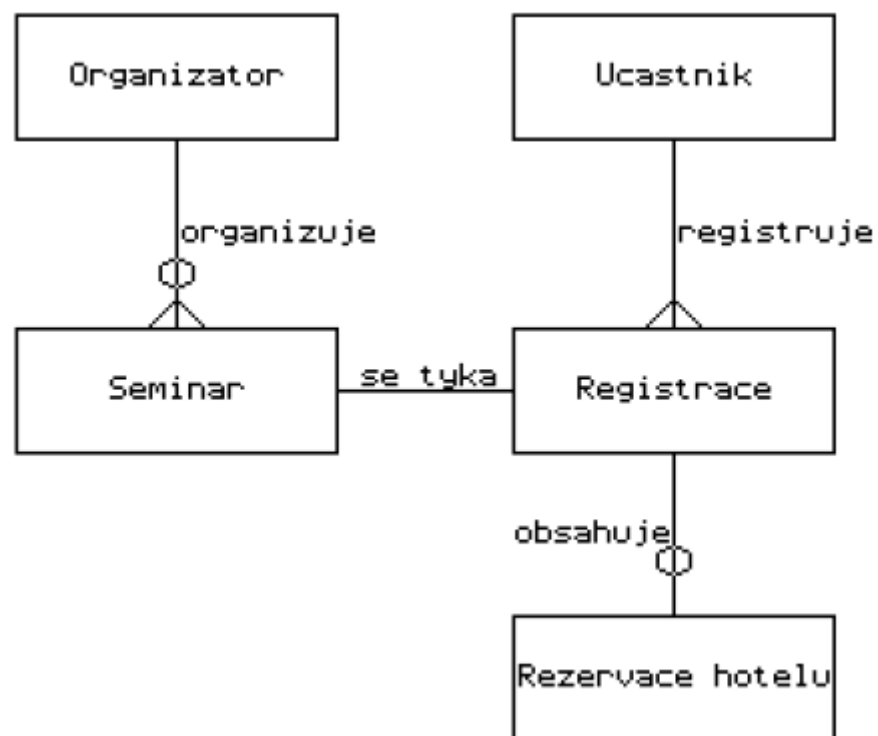


## Příklad 2 - Kurzy a školení

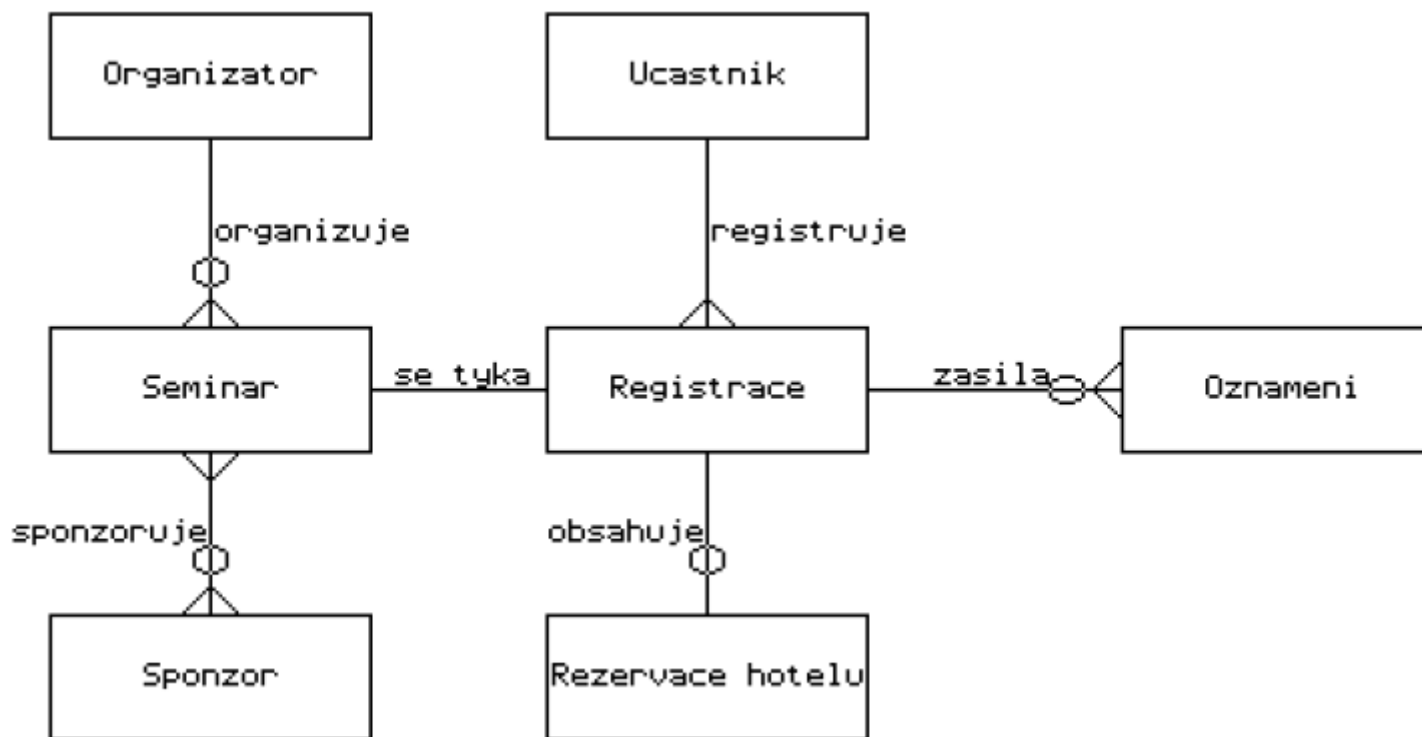




## Příklad 2 - Kurzy a školení



## Příklad 2 - Kurzy a školení



# Úkoly



- Zvolte část projektu, pro kterou vytvoříte datový model a s ním související podmnožinu požadovaných funkcí.
- Stanovte pro zvolenou část systému funkční body a faktor technické složitosti.
- S použitím programu COCOMOII stanovte přepočtem z FP délku programu v SLOC.