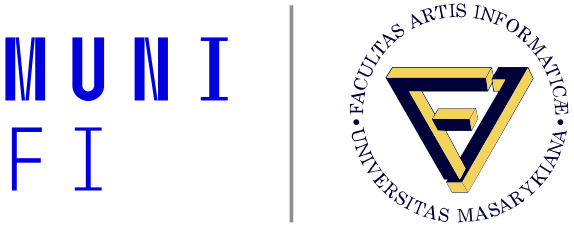


Function Point Analysis

PA017 SW Engineering II → Aspects of SW Development Management

Jaroslav Ráček **Josef Spurný**

Faculty of Informatics, Masaryk University

October 25, 2022

Motivation

- For COCOMO, we need KLOC estimation which might be challenging to estimate correctly (remember 4:1 variance in initial estimation)
- Can we avoid that...?
- Can we focus on the user perspective instead...?
- Yes, we can – Function point analysis

Core ideas

- In the analytical phase of project, we collect detailed functional requirements on a system
- Hence, we can focus on functionality to estimate SW cost
- Since we do not count KLOC, we are language-independent (high-level vs. low-level language)
- In its essence, producing software is manufacturing process requiring human work
- Therefore, we simply need to determine:
  - Manufacturing unit
  - Work cost for producing this unit

Estimations using function points

Function point = normalized software project measure

- Measurement focused on application aspect, not on technical aspect
- Size measure, not complexity measure
- Measurement of functions and data, not on code

International Function Point Users Group - www.ifpug.org

Capers Jones:

- Applied Software Measurement (1997)
- Estimating Software Costs (1998)

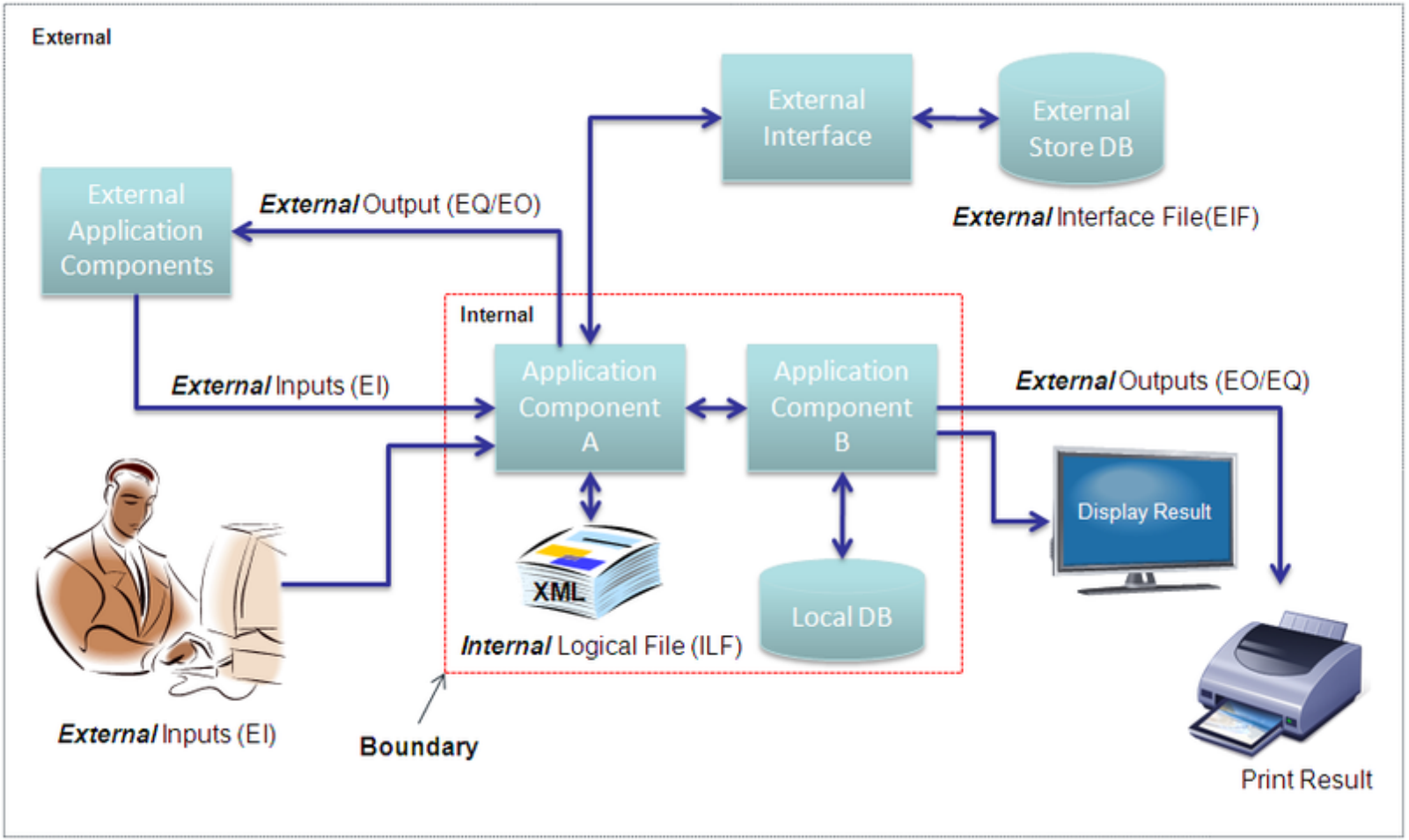
Function point - principle

- Preliminary estimation with limited information
- At the end of analytical phase, when we have screen mockups and detailed description of functionality, we can calculate FP count with relatively good precision ($\pm 10\%$)
- Measures used: inputs, outputs, enquiries, internal memories, external memories

General estimation concept

- Est = project size * complexity * risk factors

Function points overview



Link to source

Types of function points

FP related to transaction functions

- EI – external inputs
- EO – external outputs
- EQ – external enquiry

FP related to data functions

- ILF – internal logical files
- EIF – external interface files

ILF – internal logical files

- 1 ILF = each large logical set of user data or information used to manage application
- We include every logical file or each logical data set from end-user perspective that are created, used or maintained by the application
- Rather than physical files, we count logical data set as perceived by the end-user, or as it is defined during analysis or design phase of the project
- We exclude files that are not accessible to end-user through UI and that are not maintaned independently

ILF – internal logical files

- Logical entity or entity group from end-user perspective = 1 ILF
- Logical internal file generated or maintained by an application = 1 ILF
- Matrix or file maintained by end-user = 1 ILF
- Data file or configuration file used by an application to process data = 1 ILF
- Attribute entity maintained only by parent entity = 0 ILF
- Associative entities containing only key attribute = 0 ILF
- Internal temporary or classifier entity = 0 ILF
- A file created as a result of using certain technology (e.g., index file) = 0 ILF
- External sample file, read only = 0 ILF

EIF – external interface files

- We include every large logical set of user data or control information used by an application
- These data/information are maintained by an external application. We include every logical file or logical data set from end-user perspective
- We include every large logical set of user data or control information that are collected from other application via an interface
- The extraction shall not trigger a change to any internal logical files. If this happens, then we count this as External Input (EI) instead

Example: stationary radar for speeding

EIF – external interface files

- Files or records extracted from other application (used as references, links) = 1 EIF
- Database read by other application = 1 EIF
- Internal logical file of other application used as a transaction = 0 EIF, 1 EI

EI – external inputs

- We include every unique user data or enquiries that enter an application via interface and perform CRUD operation over internal logical data file
- We include control information which enters via interface and ensures consistency with user-defined function
- External input shall be considered as unique if design requires processing logic that is different from other external inputs
- Typically, each user screen (data entry set) is one FP, it doesn't matter whether it is done via one screen (one big page) or multiple consecutive pages

EI – external inputs

- GUI screen allowing to add, edit or cancel = 3 EI
- Set of related screens that are processed as one transaction = 1 EI
- Two different screens with different layout of data entry, but with same processing logic = 1 EI
- Two different screens with same format, but with different processing logic = 2 EI
- Single screen with multiple unique functions = 1 EI per each function
- Automated data input or transaction from other application = 1 EI per each transaction

EI – external inputs

- Entry of user inputs/commands into application = 1 EI
- Data entry (OCR) with one transaction = 1 EI
- Data modification function following a enquiry = 1 EI and 1 EQ
- Individual choices from menu on a screen = 0 EI
- Modification of a matrix or file maintained by an end-user = 1 EI
- Duplicate screen that has been already included = 0 EI
- External inputs used only because certain technology is used = 0 EI
- Make a selection from list of values (dropdown menu) = 0 EI

EO – external outputs

- We include every unique data or control data leaving application via an interface
- External output is considered as unique if it has unique data, or if design of an external system requires different processing method
- External outputs are often reports, output files sent to other application or messages for user

EO – external outputs

- Data export on a screen = 1 EO
- Summarizing report – batch processing = 1 EO
- Automated data export or transaction towards other applications = 1 EO
- Error messages returned as a result of input transaction = 0 EO
- Backup files = 0 EO
- Output shown on screen and sent to printer = 2 EO
- Output files created for technical purposes = 0 EO
- Output shown as bar chart as well as pie chart = 2 EO
- Output showing result of calculation = 1 EO

EQ – external enquiry

- We include every unique input/output tuple where input is cause and output is effect
- EQ is considered unique when it cause different data element types to be produced, or when it requires different data processing logic to be used (when compared to other EQ)

EQ – external enquiry

- Online input followed by online output without change to data files = 1 EQ
- Enquiry followed by a change input = 1 EQ and 1 EI
- Input and output on a help screen = 1 EQ
- Online input leading to printing without data change = 1 EQ
- Selection from list of values with dynamic data = 1 EQ
- Selection from list of values with static data = 0 EQ

Function point calculation

Before the calculation, we have to classify EI, EO, EQ, ILF, EIF by measures

Measure	low	average	high	total
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	_____

Total unmodified function points: _____

Input complexity measure (EI, EQ)

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

FTRs	1-4 DETs	5-15 DETs	16+DETs
0-1	low	low	average
2-3	low	average	high
4+	average	high	high

Output complexity measure (EO, EQ)

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

FTRs	1-4 DETs	5-15 DETs	16+DETs
0-1	low	low	average
2-3	low	average	high
4+	average	high	high

File complexity measure (ILF, EIF)

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

RETs	1-19 DETs	20-50 DETs	51+DETs
0-1	low	low	average
2-4	low	average	high
5+	average	high	high

General system characteristics – evaluation scale

- So far, we have focused only on the data perspective
- However, we need to consider other parameters which would impact work expenditure to deliver given system
- These parameters include performance, criticality, security and other aspects

- 14 characteristics evaluated on a scale according to an impact on application, where
  - 0 = no impact
  - 1 = random impact
  - 2 = low impact
  - 3 = average impact
  - 4 = significant impact
  - 5 = substantial impact

General system characteristics – factors

1. Does system require reliable backup and recovery?
2. Are data communications required?
3. Is there a distributed processing?
4. Is performance critical?
5. Will the system be operational in current intensely used operation environment?
6. Does system require online data entry?
7. Does online data entry require input transactions over multiple screens or operations?

General system characteristics – factors

8. Are main files maintained online?
9. Are input, outputs, files and enquiries complex?
10. Is internal processing complex?
11. Is the code designed such that it is reusable?
12. Are conversion and installations included in the design?
13. Is system designed to be installed multiple times for different organizations?
14. Is system designed to support changes and to be user-friendly?

Function points count

Function points count

= $(0.65 + \frac{GSP}{100}) * UFP$, where

GSP = sum of general system characteristics evaluation
UFP = unmodified function points

Note: FP can be used to estimate any type of SW: desktop or mobile app, SW in cars, home appliances, etc.

New and modification projects

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 + Δ Changed FP

How to calculate FP

1. Identify and calculate ILF, EIF, EI, EO, EQ.
2. For each ILF and EIF, identify RET and DET count. For each EI, EO, EQ, identify FTR and DET.
3. Use complexity matrix to determine count of simple, average and complex items of EI, EO, EQ, ILF and EIF.
4. Calculate unmodified FP count.
5. Evaluate 14 system characeristics.
6. Sum up the characteristics to determine *Technical complexity factor*.
7. Calculate Modified FP count = SW size from functional perspective

Size estimation (Caspers Jones)

1 FP = X commands (LOC)

- Basic assembler: 320
- Macro assembler: 213
- C: 128
- FORTRAN: 107
- C++: 64
- SQL: 13

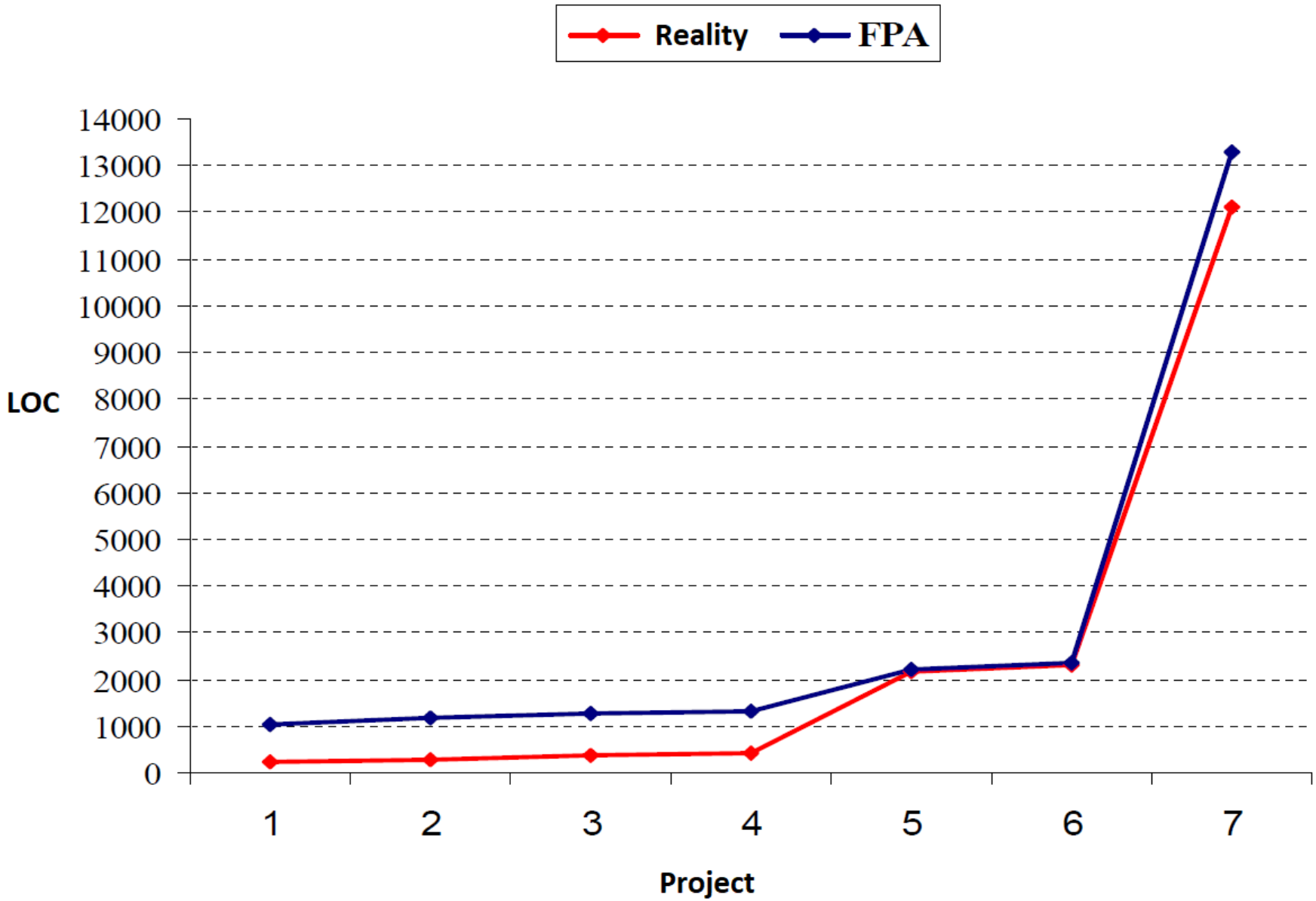
Further estimations

- $FP^{1.15} \approx$ SW project paper documentation pages count
- $FP^{1.2} \approx$ test scenarios count
- $FP^{1.25} \approx$ error potential for new SW
- $FP^{0.4} \approx$ calendar months needed for SW delivery
- $\frac{FP}{150} \approx$ employees needed for SW delivery
- $\frac{FP}{750} \approx$ employees needed to maintain SW in desired state

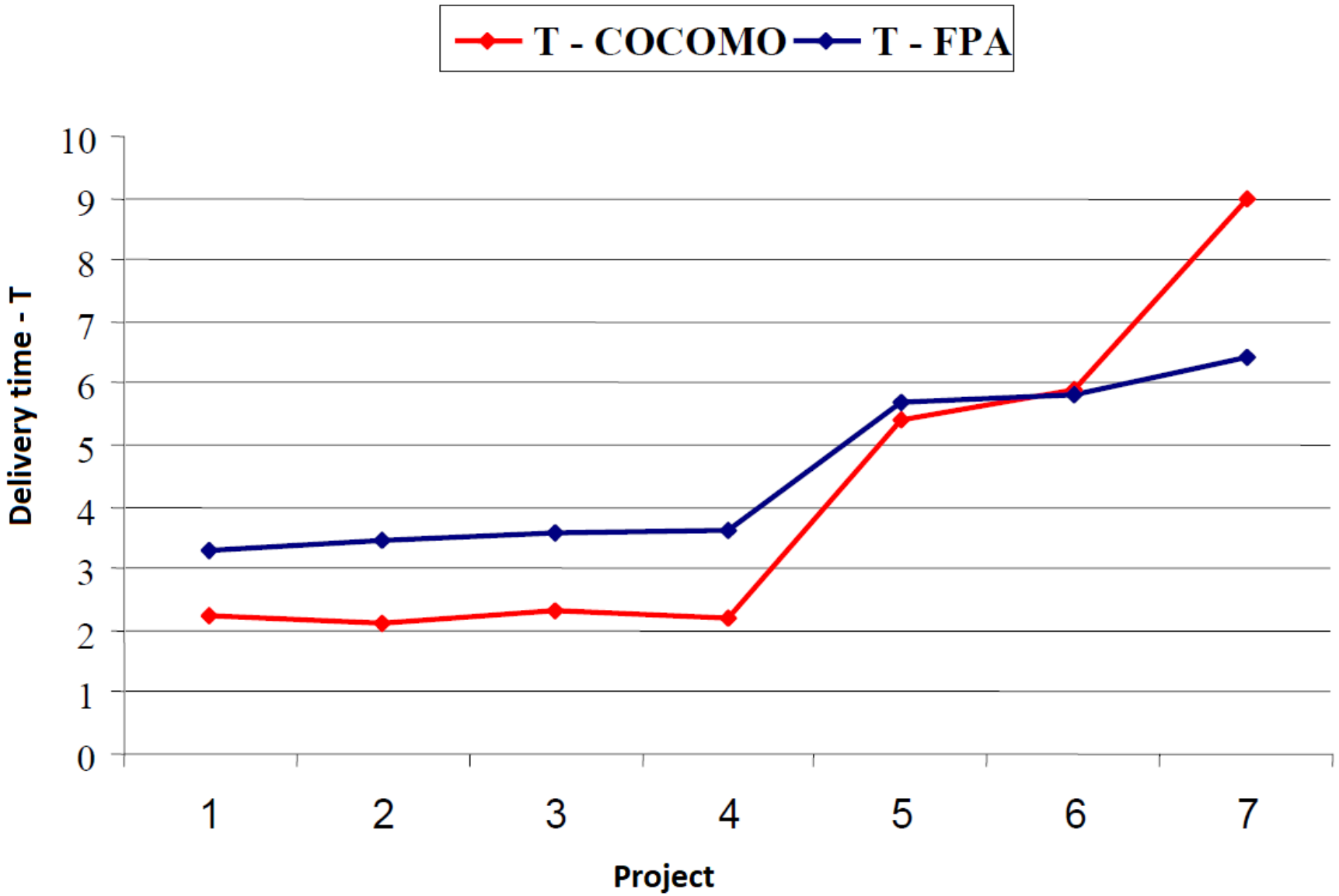
Measurement examples

- 7 student projects (Bachelor/Masters theses, seminar projects) written in C, C++, Java and Delphi @ FI MU were measured
- FP calculation was done, LOC estimation based on this calculation, then compared to reality
- Time required to deliver project was estimated using COCOMO2 and FP, results were compared

Comparison of real and estimated code size



Comparison of real and estimated project time



MUNI
FACULTY
OF INFORMATICS