Function Points

Goal

- A measure of size (delivered functionality) of an application
- Independent of technology and people

Definition

- fp = A*EI + B*EO + C*EQ + D*EIF + E*ILF
  - EI = number of Input Item
  - EO = output item
  - EQ = Inquiry
  - EIF= External Interface File
  - ILF = Internal Logical File

SOftEng http://softeng.polito.it

- Coefficients A,B,C,D,E

	Level of Complexity		
Component	**Simple**	**Average**	**Complex**
Input item	3	4	6
Output item	4	5	7
Inquiry	3	4	6
Master file	7	10	15
Interface	5	7	10

SOftEng http://softeng.polito.it

Function Points

- Ex, with all 'average' complexity

$$FP = 4 \times EI + 5 \times EO + 4 \times EQ + 10 \times ILF + 7 \times EIF$$

If EI =1, EO = 1, EQ = 1, ILF =1 , EIF = 1
Then FP = 4 + 5+ 4+ 10 + 7 = 30

SOftEng http://softeng.polito.it

Steps

1. Define counting type
2. Define boundary
3. Identify components, classify (type, complexity)
4. Adjust

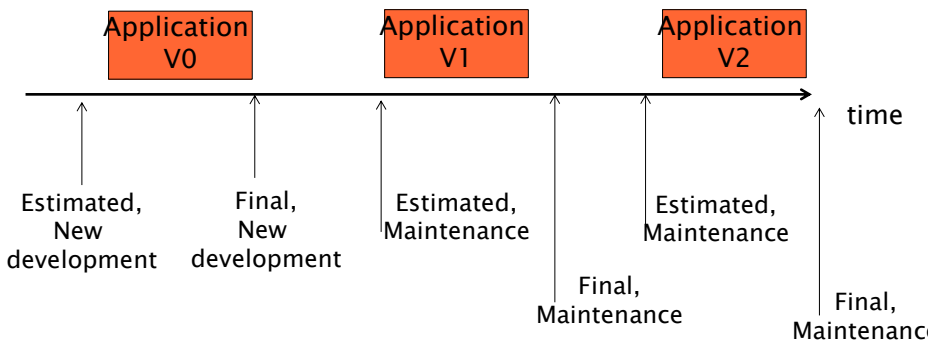
SOftEng http://softeng.polito.it

1 Counting type

- Estimated / final
  - Estimated: count before application exists
  - Final: count on delivered and installed application
- Development / maintenance
  - Development: new application
  - Maintenance: existing application

SOftEng

1 Counting type



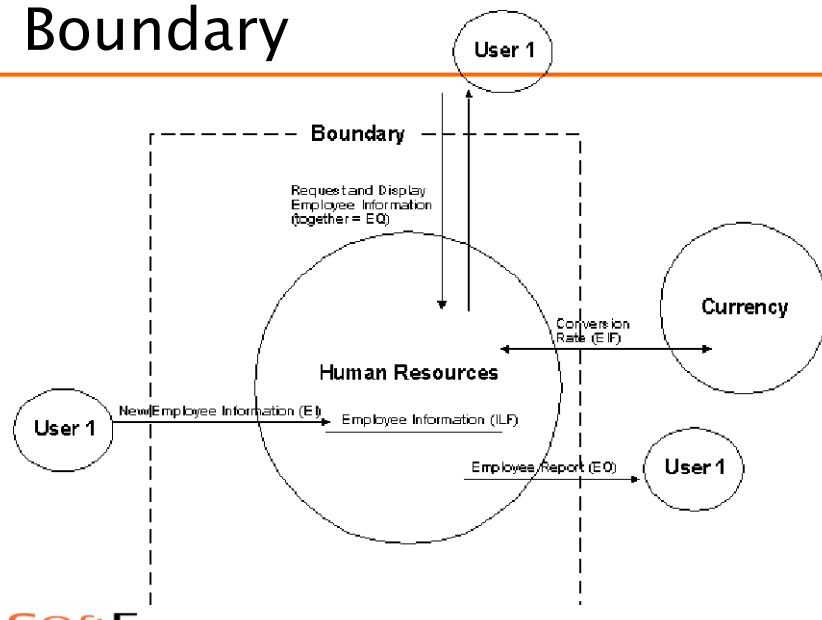
SOftEng

2 Boundary

- Define what is in the application(s), what is out
  - In should be counted, out not counted
- Depends on 'user' considered
  - User = stakeholder, defines or actually uses functionality of the application(s)
  - Cfr context diagram in UML

SOftEng

2 Boundary



SOftEng

3 Components

- Classify each component of application per type (EI, EO, EQ, ILF, EIF) and complexity (simple, average, or complex).
  - Assign appropriate number of function points
  - Sum gives UFP (unadjusted function points)

	Level of Complexity		
Component	**Simple**	**Average**	**Complex**
Input item	3	4	6
Output item	4	5	7
Inquiry	3	4	6
Master file	7	10	15
Interface	5	7	10

SOftEng http://softeng.polito.it

4 Adjust

- Compute technical complexity factor (TCF)
  - Assign value from 0 ("not present") to 5 ("strong influence throughout") to each of 14 factors such as transaction rates, portability
  - Add 14 numbers ⇒ total degree of influence (DI)

    $TCF = 0.65 + 0.01 \times DI$
  - Technical complexity factor (TCF) lies between 0.65 and 1.35

1. Data communication
2. Distributed data processing
3. Performance criteria
4. Heavily utilized hardware
5. High transaction rates
6. Online data entry
7. End-user efficiency
8. Online updating
9. Complex computations
10. Reusability
11. Ease of installation
12. Ease of operation
13. Portability
14. Maintainability

SOftEng http://softeng.polito.it

4 Adjust

- Number of function points (FP) given by
  $FP = UFP \times TCF$

SOftEng http://softeng.polito.it

Components – data

- ILF internal logical file
- EIF external interface file
  - File is a coherent group of data
    - CFR class in conceptual data model
  - Internal: maintained inside the application
  - External: maintained externally

SOftEng http://softeng.polito.it

Components – data

- RET : group of attributes
- DET : single, unrepeated attribute
  - Attribute as defined in UML class diagram

SOftEng
http://softeng.polito.it

Components – data

RET\DET	1-19	20 - 50	> 50
1	low	low	medium
2 - 5	low	medium	high
> 5	medium	high	high

ILF	
Complexity	UFP
low	7
medium	10
high	15

EIF	
Complexity	UFP
low	5
medium	7
high	10

SOftEng
http://softeng.polito.it

Components – transactional

- EI External Input
  - Processes input from outside the boundary, mantains ILF or changes behavior of application
- EO External Output
  - Sends info outside the boundary, is more than simple ILF reading (applies some processing)
- EQ External Inquiry
  - Sends info outside the boundary, no processing

Components – transactional

	EI	EO	EQ
Change behaviour of application	Main goal	possible	forbidden
Maintain ILF(s)	Main goal	possible	forbidden
Send info to user	possible	Main goal	Main goal

SOftEng
http://softeng.polito.it

Components – transactional

EI

FTR\DET	1-4	5-15	> 15
0-1	Low	Low	Medium
2	Low	Medium	High
> 2	Medium	High	High

EO/EQ

FTR\DET	1-5	6-19	> 19
0-1	Low	Low	Medium
2-3	Low	Medium	High
> 3	Medium	High	High

FTR is ILF or EIF

EI / EQ

Complexity	UFP
Low	3
Medium	4
High	6

EQ

Complexity	UFP
Low	4
Medium	5
High	7

Counting variants – estimation

Excluding adjustment

- D_FP = UFP
  - D_FP = development FP
- M_FP = ADD + CHANGE + DEL
  - M_FP = maintenance FP
  - ADD: added functions FP
  - CHANGE: changed functions FP
  - DEL: deleted functions FP

SOftEng http://softeng.polito.it

Counting variants – final

SOftEng

Function Points

- suitable for MIS
  - use of adjustment factors delicate
  - FP expert should do estimate
    - long, expensive
- conversion tables FP – LOC
  - Cobol 110
  - C 128–162
  - C++ 53–66
  - Java 53–62
- conversion tables FP – effort
  - www.ifpug.org

SOftEng

FP

- Advantage
  - Independent of technology
  - Independent of programmer
  - Well established and standardized
- Downside
  - Counting long and expensive
  - Transaction system oriented (no real time, no embedded systems)

SOftEng http://softeng.polito.it

FP vs. LOCS

	FP	LOCs
Depend on prog language	N	Y
Depend on programmer	N	Y
easy to compute	N, must be done by trained person	Y, tool based (after end of project)
Applicable to all systems	N, transaction oriented	Y

SOftEng http://softeng.polito.it

FP as unit of exchange

- Company A bids for FP
  - Buy 10000 FP, how much? (bid)
  - providers answer, x Euro per FP
- A selects provider
  - lowest cost and other factors
- End of year, redo counting
  - 10123 FP actually delivered
  - A pays

SOftEng http://softeng.polito.it

Reminder

- Measures of size
  - FP, LOC
- Both can be computed
  - Before a project start (estimated size)
  - After a project ends (actual size)
- Both can be used to
  - Characterize productivity
    - FP/effort, LOC/effort
  - Characterize application portfolio
    - FP or LOC owned and operated by a company

SOftEng http://softeng.polito.it

Function points – variants

- Functional Size Measurement Method
- FSMM variants
  - IFPUG Function Point (ISO/IEC 20926:2003),
  - COSMIC Full Function Point (ISO/IEC 19761:2003),
  - MKII Function Point (ISO/IEC 20968:2002),
  - NESMA Function Point (ISO/IEC 24570:2005
- IFPUG is the more used, and is the one presented in these slides

SOftEng http://softeng.polito.it

Function points

- IFPUG
  - FP Counting Guide
  - Exams/ certified counters
- GUFPI
- (CNIPA)

SOftEng http://softeng.polito.it