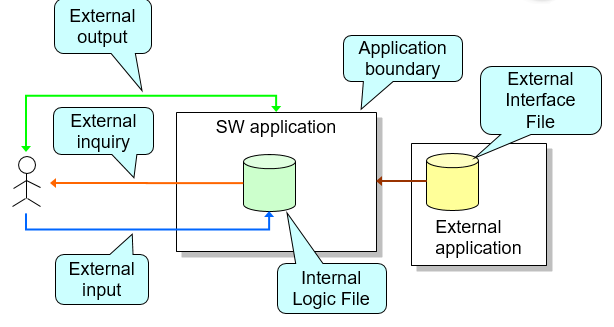
USEFUL THINGS

--------------------------------------------------------





* Function types
  + **Internal Logical File (ILF):** homogeneous set of data used and managed by the application
  + **External Interface File (EIF):** homogeneous set of data used by the application but generated and maintained by other applications

**External Input**

* + Elementary operation to elaborate data coming form the external environment

**External Output**

* + Elementary operation that generates data for the external environment
    - It usually includes the elaboration of data from logic files

**External Inquiry**

* + Elementary operation that involves input and output
    - Without significant elaboration of data from logic files

Weighting function points (slides table)

|  |  |  |  |
| --- | --- | --- | --- |
| **Function types**   * N. Inputs * N. Outputs * N. Inquiry * N. ILF * N. EIF | **Weight** | | |
| **Simple**  3  4  3  7  5 | **Medium**  4  5  4  10  7 | **Complex**  6  7  6  15  10 |

Function points identification w.r.t. mTS

Data functionalities vs Transaction functionalities

**ILF**

The application stores information about

* Users (simple) //NE CONSERVA A VOLTE ANCHE LA POSIZIONE, DICI CHE VA BENE COMUNQUE SIMPLE?
* Taxi Drivers (simple) //NON CI CONVIENE MEDIO? CONSERVA ANCHE QUI LA POSIZIONE, MA LO FA SEMPRE
* SysAdmin (simple)
* Maps (simple, most of information is external in Google Maps) //GLI INDIRIZZI CHE VANNO BENE. SONO DATI GENERALMENTE SEMPLICI, MA CE NE SONO TANTI, NON SO SE QUESTO POSSA INFLUIRE
* Reservations (simple)
* Queue (medium/complex) //MEDIUM, COMPLEX MI SEMBRA ESAGERATO. IL DATO DI PER SE é SEMPLICE, SONO LE OPERAZIONI AD ESSRE COMPLICATE

**EIF**

The external interactions of the system are the following: //ATTENZIONE, QUI PARLA DI DATI CHE VENGONO REFERENZIATI SU UN COMPONENTE ESTERNO. FORSE QUESTE API VANNO MESSE IN EXTERNAL INQUIRY

* Google Maps API (medium)
* Payment Services API (medium)

**External Inputs**

The external inputs of the system are the following: //FORSE CONVIENE FARE IL MERGE IN USER LOGIN/LOGOUT

* Customer login (simple)
* Customer logout (simple)
* Taxi Driver login (simple)
* Taxi Driver logout (simple)
* SysAdmin login (simple)
* SysAdmin logout (simple)
* Make a reservation (medium) // sys, res and cust involved
* Delete a reservation (medium) // as before
* Manage system manager (medium) //che intendi? Dici sysadmin?
* … ?
* Taxi driver position handling (simple)
* TAXI DRIVER ACCEPTS/NOT ACCEPTS A RESERVATION
* Payment acknowledge (from Payments API) (simple) però è da vedere in realtà…

Transactions between applications should be referred to as interfaces. You can only have an external output or external inquiry of data external to your application. If you get data from another application and add it to a file in your application, this is a combination get and add (external inquiry and external input).

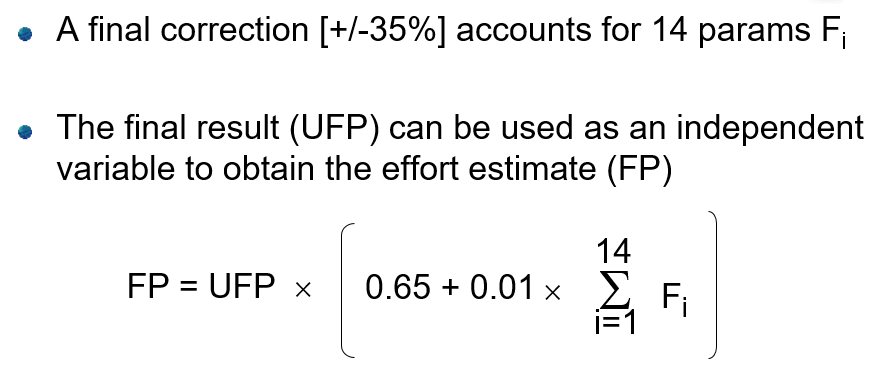
* Google Maps data (in seguito ad inquiry ottieni external inputs) (possono essere posizioni o informazioni sulle vie)

**External inquiries**

* Customer requests information about his/her profile (medium)
* Customer requests information about his/her reservations (medium)
* Taxi Driver requests information about his/her next reservations (medium) (?)
* Da aggiungere anche google maps e payments

**External outputs**

* Show the user the acknowledge of the reservation with details (hard) // several entities involved
* Show the maps on Taxi Driver’s GPS (medium)
* //SHOW DETAILS ON RESERVATION (customer application)
* //SHOW REQUEST TO TAXI DRIVER



Fi standards

These 14 GSC are

1. Data Communication

2. Distributed Data Processing

3. Performance

4. Heavily Used Configuration

5. Transaction Role

6. Online Data Entry

7. End-User Efficiency

8. Online Update

9. Complex Processing

10. Reusability

11. Installation Ease

12. Operational Ease

13. Multiple Sites

14. Facilitate Change

**Function Point Estimation Worksheet**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Complexity | | |  |
| Description | Low | Medium | High | Total |
| Inputs | \_\_\_ x 3 | \_\_\_ x 4 | \_\_\_ x 6 | \_\_\_ |
| Outputs | \_\_\_ x 4 | \_\_\_ x 5 | \_\_\_ x 7 | \_\_\_ |
| Queries | \_\_\_ x 3 | \_\_\_ x 4 | \_\_\_ x 6 | \_\_\_ |
| Files | \_\_\_ x 7 | \_\_\_ x 10 | \_\_\_ x 15 | \_\_\_ |
| Program Interfaces | \_\_\_ x 5 | \_\_\_ x 7 | \_\_\_ x 10 | \_\_\_ |

Total Unadjusted Function Points (TUFP): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(0=no effect on processing complexity; 5=great effect on processing complexity)

|  |  |
| --- | --- |
|  | 0-5 |
| Data communications | \_\_\_ |
| Heavily use configuration | \_\_\_ |
| Transaction rate | \_\_\_ |
| End-user efficiency | \_\_\_ |
| Complex processing | \_\_\_ |
| Installation ease | \_\_\_ |
| Multiple sites | \_\_\_ |
| Performance | \_\_\_ |
| Distributed functions | \_\_\_ |
| On-line data entry | \_\_\_ |
| On-line update | \_\_\_ |
| Reusability | \_\_\_ |
| Operational ease | \_\_\_ |
| Extensibility | \_\_\_ |

Processing Complexity (PC): \_\_\_\_\_\_

Adjusted Processing Complexity (PCA) = 0.65 + (0.01 \* \_\_\_\_\_\_\_\_\_\_\_\_)

Total Adjusted Function Points (TAFP): \_\_\_\_\_\_\_\_\_\_\_ \* \_\_\_\_\_\_\_\_\_\_\_\_ =

# Function points: the approach

A **function point** is a "*unit of measurement*" to **estimate the cost** of a software product with respect to some criteria. We can say that function points measure **software size** and **costs** with a certain approximation.

Historically, **Allan Albrecht** (from **IBM**) introduced function points in **1979** in his work "*Measuring Application Development Productivity*". The functional user requirements of the software are grouped in **five macro-categories:**

* **Internal Logical File (ILF):** homogeneous set of data used and managed by the application.
* **External Interface File (EIF)**: homogeneous set of data used by the application but generated and maintained by other external applications.
* **External Input**: elementary operation to elaborate data coming from the external environment (from users).
* **External Output**: elementary operation that generates data for the external environment. It usually includes the elaboration and a proper representation of data from logic files.
* **External Inquiry**: elementary operation that involves input and output (e.g. specific requests of the user in order to visualize his/her information). This category does not implies significant elaboration of data from logic files.

The general idea is to give a **weight** for each category, with respect to a "*difficulty range*” (Simple, Medium and Complex).

/\* other maybe \*/