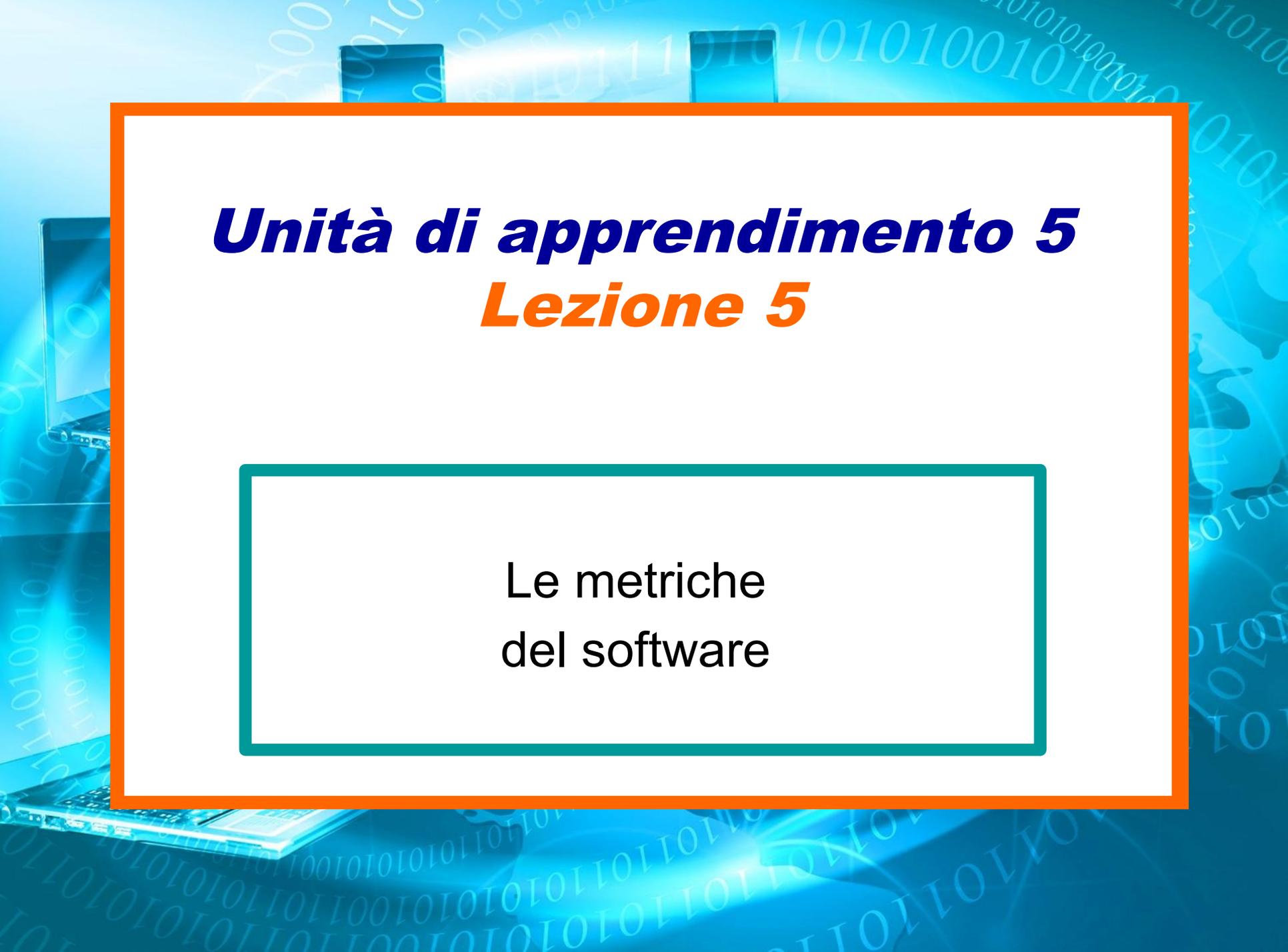


Unità di apprendimento 5

Gestione di progetti
informatici

The background is a vibrant blue gradient with a pattern of binary code (0s and 1s) scattered across it. On the left side, there is a partial view of a laptop, showing its screen and keyboard. The overall aesthetic is high-tech and digital.

Unità di apprendimento 5

Lezione 5

Le metriche
del software

In questa lezione impareremo...

- le metriche per la stima delle quantità
- il modello LOC e FSA
- la tecnica Backfiring

Tipologie di metriche

Nell'ingegneria del software le metriche vengono usate per quantificare e qualificare:

- **i prodotti (metriche di prodotto):** i progetti, il codice sorgente e i casi di test;
- **i processi (metriche di processo):** le attività di analisi, progettazione e codifica;
- **gli elementi di qualità (metriche di qualità):** l'affidabilità del software, l'usabilità del prodotto, ecc;
- **gli "attori" del software:** l'efficienza di un collaudatore, la produttività di un progettista ecc.

Tipologie di metriche

Le **metriche di prodotto** sono strumenti che forniscono misure quantitative del prodotto e delle sue componenti.

Permettono di valutare la quantità (**mole**) di lavoro da svolgere in base alla “dimensione” del software da realizzare.

mole software → **effort misurato in tempo/persona** → **costo**

tempo/persona: **giorni/uomo** (o nei suoi multipli **settimane/uomo** e **mesi/uomo**)

Per misurare la “dimensione” del software sono stati definiti diversi tipi di misurazioni:

- **metriche strutturali**: basate sulla complessità del codice definito in funzione dei possibili cammini percorribili dallo stesso (**numero ciclomatico**);
- **metriche dimensionali**: valutano la lunghezza del codice (**LOC = Line Of Code**);
- **metriche funzionali**: misurano le funzioni offerte dal software (**FPA = Function Point Analysis**).

Metrica strutturale

I metodi strutturali si basano sulla complessità di un programma, identificata dal **numero cicломatico**.

Un programma può sempre essere rappresentato con un grafo i cui nodi sono rappresentati da if, while, switch e da tutte le altre istruzioni condizionali che lo compongono.

Il **numero cicломatico** fornisce la dimensione della complessità del software in funzione dei possibili cammini percorribili nel codice, in relazione ai nodi del grafo rappresentativo del codice.

Questa metrica considera il **grado di complessità del codice inversamente proporzionale alla sua qualità**, cioè, più il codice è complesso, maggiore è la probabilità di errore e la difficoltà nel mantenerlo.

Metrica dimensionale (LOC)

Con il termine **LOC (Lines Of Code)** oppure **SLOC (Source Lines Of Code)** e il suo multiplo **KLOC (1000 LOC)** si intende l'unità di misura della metrica dimensionale che consiste nel misurare la lunghezza del codice (**numero di linee di codice (sorgente)**).

E' la **più semplice** per effettuare la valutazione del software **a consuntivo**.

La misura non è obiettiva per molte ragioni:

- la diversa “espressività” dei linguaggi di programmazione (un programma assembly è molto più lungo di un programma scritto con un linguaggio ad alto livello);
- diversi stili di programmazione;
- la presenza di linee bianche e di commento.

Metrica dimensionale (LOC)

Le differenze derivanti dall'uso di diversi tipi di linguaggi di programmazione viene regolato utilizzando appositi coefficienti moltiplicativi e, oggi, la stima della mole di lavoro viene stimata utilizzando FP (Function Point), da cui è possibile ricavare anche una stima preventiva di LOC.

LOC viene oggi usato per derivare altri attributi, ad esempio:

- la probabilità di presenza di errori nel programma che risulta proporzionale al LOC;
- il grado di documentazione del codice (CLOC/LOC, dove CLOC è il numero di righe di commento).

Metrica dimensionale (LOC)

SLOC (qualche dato)

■ NASA space shuttle flight control	430 K (shuttle)	1,4 M (ground)
■ Microsoft Windows 3.1 (1992)		3 M
■ Microsoft NT (1992)		4 M
■ Sun Solaris (1998)		7 M
■ Microsoft Windows 95		14 M
■ Microsoft Windows 98		18 M
■ Microsoft NT5.0 (1998)		20 M
■ RedHatLinux 6.2 (2000)		17 M
■ RedHatLinux 7.1 (2002)		32 M
■ Microsoft Windows XP		40 M
■ Microsoft Vista		50 M

Metriche funzionali (FPA)

Lo **FPA** (**Function Point of Analysis**) è un processo per misurare la **mole**, tendo conto delle differenze dei linguaggi di programmazione, secondo le seguenti idee di base:

- ricava un indice delle dimensioni del programma in base a ciò che deve fare (funzionalità che deve fornire), rispondendo alla domanda: “**quante funzionalità si dovranno fornire?**”;
- la **quantificazione** dell'applicazione software dovrebbe avvenire dal **punto di vista dell'utente**, e non del programmatore o sviluppatore (utopistico);
- **non considera gli aspetti tecnici** e implementativi dell'applicazione (linguaggi di programmazione, tool di sviluppo, ecc.);
- **non considera la difficoltà** reale dell'**algoritmo**;
- è possibile **passare dai punti funzione al LOC**.

Metriche funzionali (FPA)

Il **Function Point** nasce nel **1979**, tecnica introdotta per **IBM** da **Allan Albrecht**.

Nel **1986** nasce l'organismo no profit **IFPUG** (International Function Points User Group) per studiare e mettere a punto una metodologia per calcolare la mole, producendo un documento di riferimento.

ISO/IEC recepisce il lavoro di IFPUG ed emana, nel corso degli anni, una **standardizzazione ISO**.

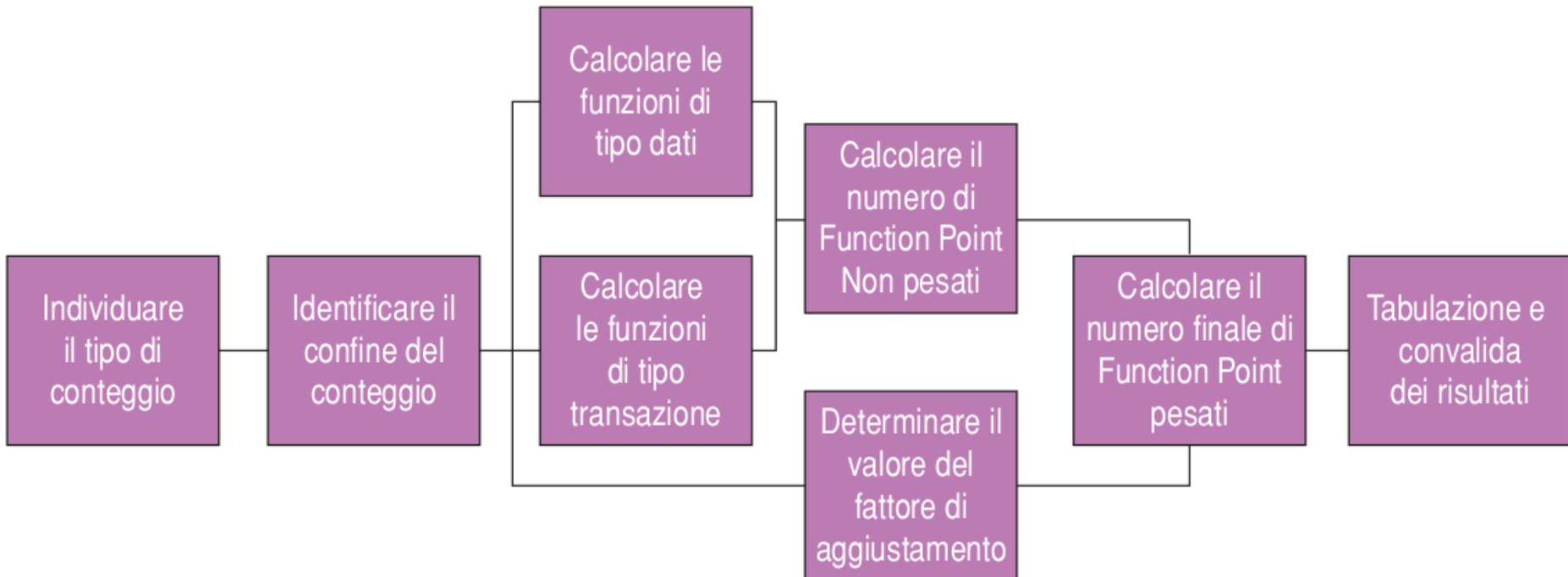
Metodo Standard IFPUG

Approfondimento

Il metodo **IFPUG** è il metodo standard per la misurazione dei **FP** ed è composto da 7 fasi:

- 1) Pianificazione del conteggio dei Punti Funzione e identificazione del tipo di conteggio
- 2) Raccolta della documentazione e identificazione del confine (tra sistema da misurare e applicazioni esterne/utente)
- 3) Inventario delle Funzioni e calcolo **FP non pesati**
- 4) Classificazione dei componenti e calcolo complessità
- 5) Esamina delle 14 Caratteristiche Generali del Sistema (**GSC** - General System Characteristics) e determinazione del valore del Fattore di Aggiustamento (**VAF** - Value Adjustment Factor)
- 6) Tabulazione dei risultati
- 7) Convalida dei risultati

Metodo Standard IFPUG



Metodo Standard IFPUG

1) Pianificazione del conteggio dei Punti Funzione e identificazione del tipo di conteggio

- Il calcolo dei FP deve essere incluso nella pianificazione del progetto.
- Può essere fatta prima dell'analisi dei requisiti e, una modifica dei requisiti implica la modifica dei FP.
- È necessario verificare i FP al termine della raccolta dei requisiti.
- I nuovi FP vanno etichettati in modo diverso per distinguerli dagli altri.

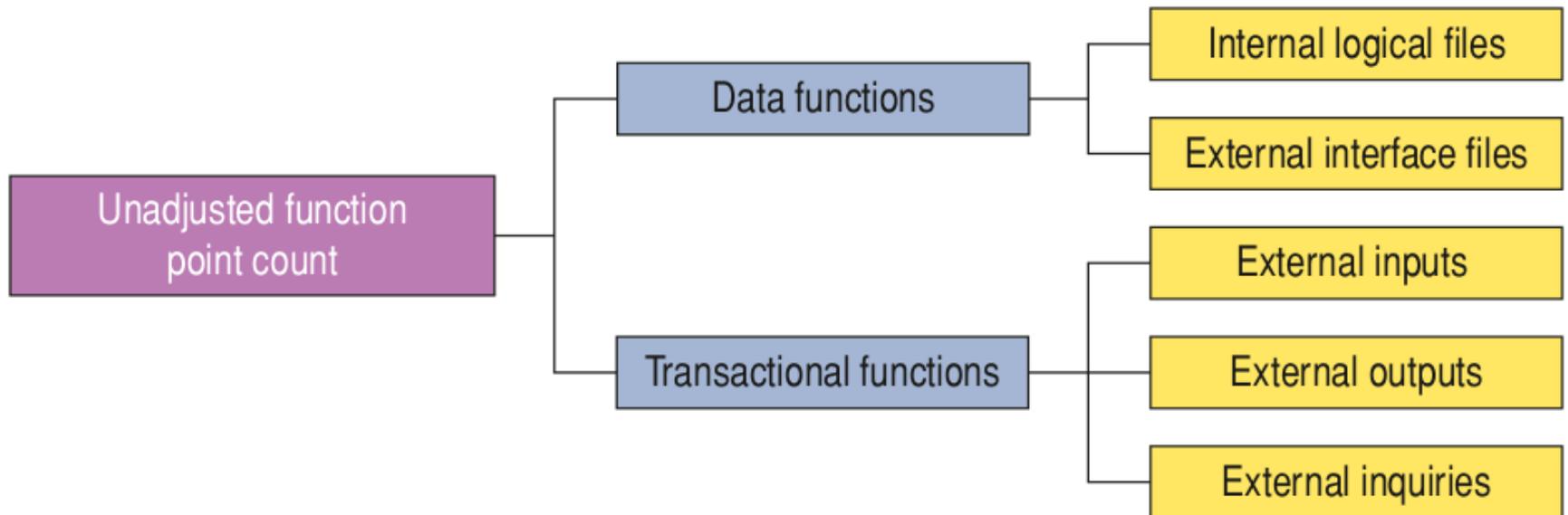
2) Raccolta della documentazione e identificazione del confine (tra sistema da misurare e applicazioni esterne/utente)

- Prima del termine dell'analisi dei requisiti analizzare la documentazione relativa a esigenze degli utenti, dell'eventuale sistema corrente e il nuovo che lo sostituirà, della struttura del sistema da realizzare, di ogni obiettivo primario del progetto e altra documentazione utili, per conteggiare correttamente i FP.
- Dopo la conclusione della fase di analisi e progetto analizzare la documentazione relativa a formato interfacce, schemi di report e stampa, maschere input, interfacce con altri sistemi, modelli logici dei dati, dimensione e formato dei file, opzioni menù, per verificare i FP.

Metodo Standard IFPUG

3) Inventario delle Funzioni e calcolo FP non pesati

È il momento in cui vengono individuate le funzionalità che il software deve fornire.



Metodo Standard IFPUG

3) Inventario delle Funzioni e calcolo FP non pesati

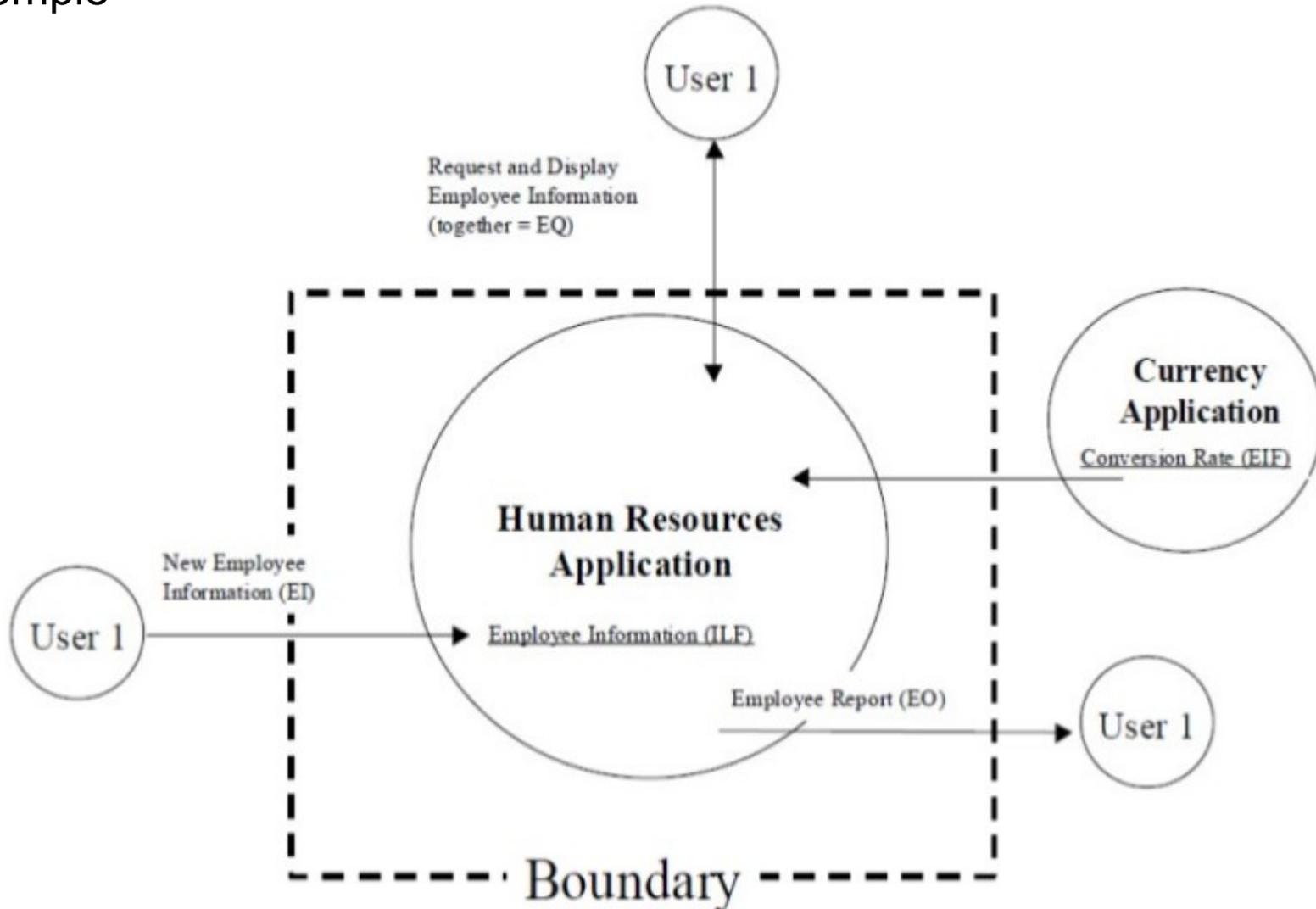
Il calcolo dei FP non pesati avviene come **somma delle cinque componenti o “funzioni logiche” ILF, EIF, EI, EO, EQ**. Ogni parametro ha dei pesi differenti e la somma di tutti i parametri genera appunto i FP non pesati (**UPF - Unadjusted FP**).

- **Internal Logical File (ILF)**: gruppo di dati logicamente collegati gestito all'interno dell'applicazione (file modificabili).
- **External Interface File (EIF)**: gruppo di dati logicamente collegati utilizzati dall'applicazione, ma gestito esternamente da un'altra applicazione (file non modificabili).
- **External Inputs (EI)**: input dei dati effettuato dall'utente o da un'altra applicazione.
- **External Outputs (EO)**: output di dati esterno all'applicazione (report, file, stampe).
- **External Inquiries (EQ)**: interrogazioni per reperire dati da file senza implicare la relativa modifica.

Metodo Standard IFPUG

3) Inventario delle Funzioni e calcolo FP non pesati

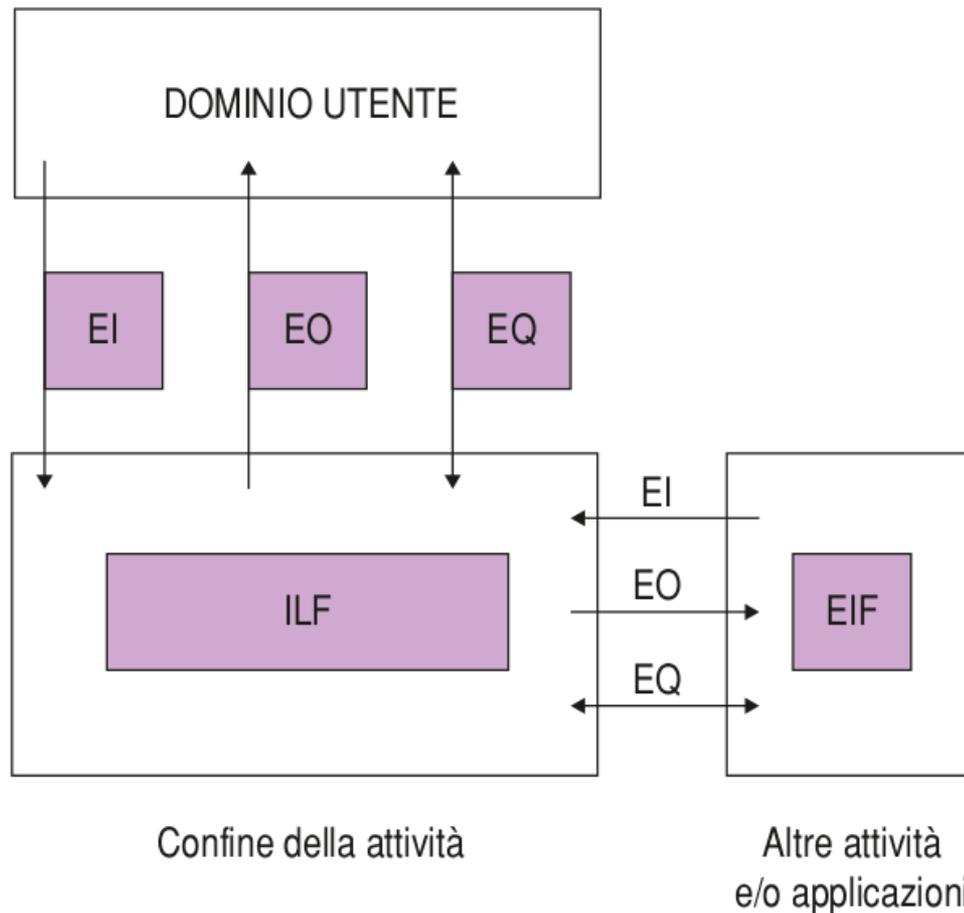
Esempio



Metodo Standard IFPUG

4) Classificazione dei componenti e calcolo complessità

Relazione fra i diversi tipi di FP, mettendo in evidenza il legame tra il dominio dell'utente, il confine delle attività e le altre applicazioni.



Metodo Standard IFPUG

4) Classificazione dei componenti e calcolo complessità

Non è sempre immediato classificare i processi nelle tre funzioni transazionali precedentemente indicate e, spesso, è difficile individuare anche il **processo elementare**.

- Il **processo elementare** è la più piccola unità di attività che è significativa per l'utente. Deve essere autonomo e lasciare l'applicazione in uno stato di coerenza funzionale.

I processi elementari possono essere identificati tramite i seguenti verbi che li identificano:

- **EI**: acquisire, inserire, aggiungere, inviare da parte dell'utente, impostare, importare, popolare, assegnare, pianificare, schedare, cancellare, eliminare, modificare, aggiornare, alterare, accettare, variare, rivedere, controllare, validare, memorizzare, mantenere;
- **EO**: calcolare e contabilizzare, fornire, stampare, visualizzare, produrre, trasmettere, esportare, inviare da parte del computer, presentare, pubblicare;
- **EQ**: fornire, stampare, visualizzare, produrre, trasmettere, esportare, inviare da parte del computer, presentare, pubblicare, interrogare, ricercare, decodificare, leggere, accedere, listare.

Metodo Standard IFPUG

4) Classificazione dei componenti e calcolo complessità

Le funzionalità possono avere complessità diverse, quindi IFPUG prevede che, dopo averle classificate, siano “pesate” utilizzando delle tabelle di riferimento dalle quali si deduce inizialmente un giudizio di complessità (alto, medio, basso), successivamente tradotto in un punteggio da assegnare alla funzione.

Gli indicatori elementari utilizzabili per la valutazione della complessità sono:

- **RET (Record Element Type)**: un sottogruppo di dati all'interno di un ILF/EIF (ad es. la singola tabella fisica di un DB relazionale);
- **DET (Data Element Type)**: singoli campi del RET, formati da tipi di dati semplici (numerico, alfabetico, data, ecc.);
- **FTR (File Type Referenced)**: file logico interno letto o mantenuto (ILF), o file esterno di interfaccia letto (EIF).

Metodo Standard IFPUG

4) Classificazione dei componenti e calcolo complessità

ILF-EIF: la loro complessità dipende dal numero di **RET** che definiscono e dal numero di **DET** che questi contengono al loro interno. Un esempio di tabella utilizzabile per determinare il **giudizio di complessità per ILF/EIF** è la seguente:

N[^] RET \ N[^] DET	1-19	20-50	>50
1	bassa	bassa	media
2-5	bassa	media	media
>5	media	alta	alta

EI-EO-EQ: la loro complessità dipende dal numero di **FTR** che utilizzano (ILF/EIF) e dal numero di **DET** che questi file contengono al loro interno. Esempi di tabelle utilizzabili per determinare il **giudizio di complessità per EI-EO-EQ** sono le seguenti (EQ utilizza il maggiore fra EI e EO):

FTR/DET	1-4	5-15	16-
0-1	Bassa	Bassa	Media
2	Bassa	Media	Alta
3-	Media	Alta	Alta

EI

FTR/DET	1-5	6-19	20-
0-1	Bassa	Bassa	Media
2-3	Bassa	Media	Alta
4-	Media	Alta	Alta

EO

Metodo Standard IFPUG

4) Classificazione dei componenti e calcolo complessità

Ad ogni giudizio di complessità si associa un peso in base, ad esempio, alle seguenti tabelle:

■ ILF	Bassa	7	■ EIF	Bassa	5			
■	Media	10	■	Media	7			
■	Alta	15	■	Alta	10			
■ EI	Bassa	3	■ EO	Bassa	4	■ EQ	Bassa	3
■	Media	4	■	Media	5	■	Media	4
■	Alta	6	■	Alta	7	■	Alta	6

Infine si calcola il numero di **UFP** (Unadjusted Function Points) con la seguente sommatoria:

$$UFP = \sum_{i=1}^{15} ((\# \text{ elementi varietà } i) * \text{peso}_i)$$

Metodo Standard IFPUG

4) Classificazione dei componenti e calcolo complessità

Esempio

■	ILF	Bassa	7
■		Media	10
■		Alta	15
■	EIF	Bassa	5
■		Media	7
■		Alta	10
■	EI	Bassa	3
■		Media	4
■		Alta	6
■	EO	Bassa	4
■		Media	5
■		Alta	7
■	EQ	Bassa	3
■		Media	4
■		Alta	6

Esempio: supponiamo che siano stati individuati:

1 ILF, di complessità B

2 EIF, uno di complessità B e uno di complessità M

3 EI, di complessità B

3 EO, due di complessità B e uno di complessità A

0 EQ

si ha:

$$\begin{aligned} \text{UFP} &= 1 \cdot 7 + (1 \cdot 5 + 1 \cdot 7) + 3 \cdot 3 + (2 \cdot 4 + 1 \cdot 7) = \\ &= 43 \end{aligned}$$

Metodo Standard IFPUG

5) Esamina delle 14 Caratteristiche Generali del Sistema (GSC - General System Characteristics) e determinazione del valore del Fattore di Aggiustamento (VAF - Value Adjustment Factor)

- Il VAF (Value Adjustment Factor) permette di determinare in modo più adeguato la metrica dell'applicazione. Viene calcolato valutando 14 caratteristiche generali del sistema (GSC – General System Characteristics), determinando qual è il grado di influenza di ciascuna secondo una scala numerica da 0 (*ininfluente*) a 5 (*incidenza essenziale*).
- Il VAF è determinato tramite la seguente formula, in cui i punteggi assegnati alle singole caratteristiche sono identificati dal termine F_i :

$$VAF = 0,65 + 0,01 \cdot \sum_{i=1}^{14} F_i$$

Metodo Standard IFPUG

5) Caratteristiche Generali del Sistema (GSC - General System Characteristics)

GSC	Caratteristiche generali del Sistema	Descrizione
1	Comunicazione dati	Quante semplificazioni per le comunicazioni ci sono al fine di aiutare il trasferimento o lo scambio di informazioni con l'applicazione o il sistema? 0: Semplice elaborazione batch 5: Transazionale con più di un front end e più protocolli di comunicazione
2	Elaborazioni dati distribuita	Come sono gestiti i dati e le funzioni di elaborazione distribuite? 0: Nessuna funzione presente 5: Elaborazione distribuita dinamicamente
3	Prestazioni	Qual è il tempo di risposta o la capacità dati richiesta dall'utente? 0: Nessun requisito 5: Requisiti molto stringenti (applicazioni time critical)
4	Configurazione utilizzata intensamente	Quanto intensamente è utilizzata la piattaforma hardware dove l'applicazione sarà eseguita? 0: Nessun vincolo 5: Utilizzo di molte configurazioni con pesanti vincoli sulle risorse
5	Tasso di transazioni	Quanto frequentemente vengono eseguite le transazioni? (giornalmente, settimanalmente, mensilmente ...) 0: Nessun picco di utilizzo 5: Vari picchi e stringenti livelli di servizio

Metodo Standard IFPUG

5) Caratteristiche Generali del Sistema (GSC - General System Characteristics)

GSC	Caratteristiche generali del Sistema	Descrizione
6	Immissione dati online	Quale percentuale di informazioni saranno inserite online? 0: Tutto batch 5: Le transazioni per inserimento dati sono > 30% di quelle presenti nella applicazione
7	Efficienza orientata all'utente finale	L'applicazione è stata progettata per un'efficienza orientata all'utente finale? 0: Nessun supporto per l'utente 5: Altissima usabilità
8	Aggiornamento online	Quanti File Logici Interni (ILF) saranno aggiornati da transazioni online? 0: Non fornisce nessun aggiornamento interattivo 5: Altissima protezione contro la perdita dei dati, con procedure automatiche
9	Complessità dell'elaborazione	L'applicazione ha una logica estesa o elaborazioni matematiche? 0: Nulla 5: Elaborazioni matematiche o logiche complesse
10	Riusabilità	L'applicazione è stata sviluppata per soddisfare le necessità di uno o più utenti? 0: non esiste codice riusabile 5: alta necessità di riusabilità e parametrizzazione del codice

Metodo Standard IFPUG

5) Caratteristiche Generali del Sistema (GSC - General System Characteristics)

GSC	Caratteristiche generali del Sistema	Descrizione
11	Facilità di installazione	Quanto è complessa la conversione e l'installazione? 0: Non sono richieste particolari inizializzazioni 5: Sono necessarie guide e strumenti di installazione e conversione
12	Facilità operativa	Quanto efficaci e/o automatici sono le procedure di awio, backup e re-recovery? 0: Nessuna esigenza 5: Applicazione per operazioni non presidiate
13	Installazioni multiple	L'applicazione è stata specificamente progettata, sviluppata e supportata per essere installata su più postazioni e per più organizzazioni? 0: Un solo sito (applicazioni stand alone) 5: Molti siti caratterizzati da ambienti hw/sw eterogenei
14	Modificabilità	L'applicazione è stata specificamente progettata, sviluppata e supportata per essere facilmente modificata? 0: Applicazione rigida 5: Esistono molti strumenti interni alla applicazione per permetterne una facile modifica (macro, tabelle ecc.)

Metodo Standard IFPUG

6) Tabulazione dei risultati

Il calcolo degli FP “adjusted” si effettua moltiplicando il valore FP “unadjusted” per il valore di aggiustamento (VAF):

$$FP = UFP * VAF$$

I diversi valori calcolati vengono normalmente tabulati in un foglio di calcolo.

7) Convalida dei risultati

L'ultima fase del processo di determinazione dei FP è quella di validazione, mediante il controllo da parte di tutti i membri del gruppo di sviluppo che verificano la completezza di quanto analizzato e valutato, per assicurarsi che tutte le componenti (EI, EO, EQ, ILF, EIF) siano state incluse.

Passaggio da FP a SLOC

Esistono delle **tabelle di corrispondenza** che associano ad ogni **FP** un certo numero di **SLOC**, in base al tipo di linguaggio di programmazione, ad esempio:

Linguaggio	SLOC/FP
Assembler	320
C	150
FORTRAN	106
Cobol	105
Pascal	91
Basic	64
Query languages	16
Speadsheet lang.	6