

Function Point Analysis

Introduzione

- Misurare la dimensione di un software è fondamentale
 - ♦ Per stimare tempi e costi
 - ♦ Per capire lo stato di avanzamento di un progetto
 - ♦ Per caratterizzare il portafoglio applicativo di una organizzazione

Introduzione

- Le Linee di codice (LOC) e varianti (KSLOC, NCLOC, ..) sono state la prima misura di dimensione, ma con vari difetti
 - ♦ Dipendenza da linguaggio di programmazione e stile programmazione
 - ♦ Misurabile tardi nel processo di sviluppo
- Negli anni 70 iniziati tentativi di definire misure funzionali del software

Storia

- 1979 Albrecht
 - ♦ Function point analysis come misura funzionale, indipendente dal linguaggio di programmazione e dalla tecnologia
- Anni 80
 - ♦ Nascita di IFPUG (Int function point user group) per standardizzare il metodo
- Anni 90
 - ♦ Standardizzazione ulteriore, nascita di altri metodi conteggio
 - ♦ Early function points

Standards

- 4 standard ISO (dell'International Organization for Standardization) di Functional Size Measurement Method (FSMM)
 - ◆ 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).
- Metodo IFPUG (evoluzione di Albrecht) rimane ad oggi il piu usato

Uso

- International Software Benchmarking Standards Group (ISBSG).
- L'ISBSG Data Disk R9 contiene alcune centinaia di progetti
 - ◆ 90% misurati in IFPUG Function Point,
 - ◆ 5% in NESMA
 - ◆ 5% COSMIC, Mark II, ...

IFPUG FPA...

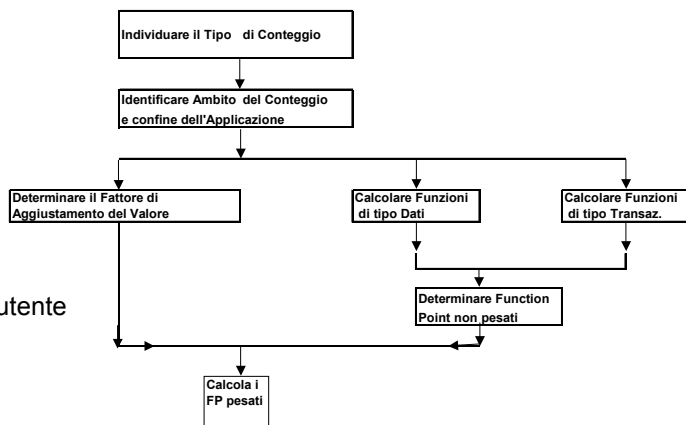
- misura le funzionalità presenti in un'applicazione/soluzione software
 - dal punto di vista dell'utente
 - Funzioni che utente chiede e riceve
 - indipendentemente dalla tecnologia utilizzata e dall'effort necessario
 - A partire dai requisiti
- Definito in CPM v4.2 (Counting Practices Manual), edito da IFPUG

FPA – passi

- Scegliere il tipo di conteggio
- Stabilire il boundary
- Identificare le funzioni
- Contare le funzioni (UFP)
- Aggiustare il conteggio in base a caratteristiche non funzionali



unto di vista dell'utente



- **Conteggio per un Progetto di Sviluppo**
stima le funzioni fornite agli utenti con la prima installazione del software, rilasciato al termine del progetto
- **Conteggio per un Progetto di Manutenzione Evolutiva**
stima le modifiche, apportate o da apportare a un'applicazione esistente, che comportano aggiunte, modifiche o cancellazioni di funzioni utente, rilasciate al termine del progetto.

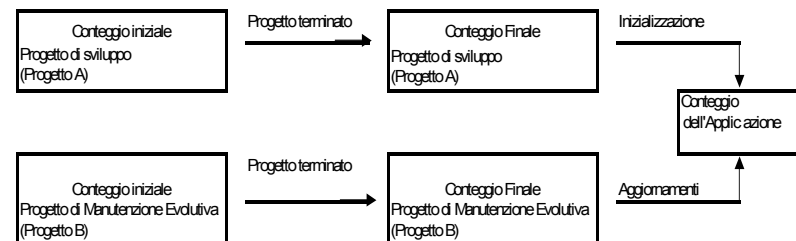
Tipi di Conteggio

Conteggio per un' Applicazione

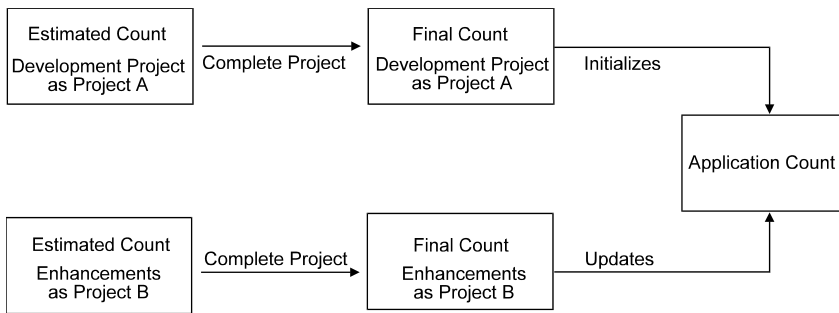
Misura le funzionalità di una applicazione terminata e installata. Chiamato 'numero di function point della *baseline*' o 'installati'.

E' calcolato alla fine del primo sviluppo, e alla fine di ogni manutenzione.

Tipi di Conteggio



- I conteggi iniziali di FP sono delle stime delle funzionalità che verranno rilasciate.
- Un conteggio iniziale può essere seguito da uno o più conteggi intermedi ed un solo conteggio finale. Fra un conteggio e l'altro l'utente può indicare nuove funzionalità o modificare quelle iniziali (scope creep -> slittamento dell'ambito).



Utente

qualsiasi persona che specifichi i requisiti funzionali e/o qualsiasi persona o cosa che comunichi o interagisca con il software in qualsiasi momento

- Il punto di vista dell'utente:
 - ◆ E' una descrizione delle funzionalità applicative
 - ◆ È approvato dall'utente
 - ◆ Può essere usato per il conteggio dei FP
 - ◆ Può variare nella sua rappresentazione fisica

Identificare scope e boundary

- definiscono le funzionalità che dovranno essere incluse in un particolare conteggio di FP:
 - ◆ Definisce un (sotto)insieme del sw oggetto di misura
 - ◆ Identifica quali funzioni dovranno essere incluse nel conteggio così da fornire risposte rilevanti all'obiettivo del conteggio
 - ◆ Potrebbe includere più di una applicazione

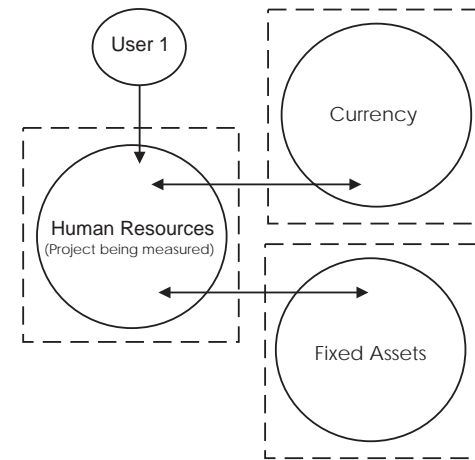
Boundary

- Il confine dell'applicazione indica la linea di separazione fra il sw oggetto di misura e l'utente.
- Il confine dell'applicazione:
 - ◆ Definisce che cosa è esterno all'applicazione
 - ◆ E' l'interfaccia concettuale tra applicazione, che costituisce l'interno e mondo utente che costituisce l'esterno

Il boundary

- ◆ Agisce come una “membrana” attraverso la quale i dati elaborati dalle transazioni(EI,EO,EQ) entrano ed escono dall'applicazione.
- ◆ Comprende i dati logici mantenuti dall'applicazione (ILF).
- ◆ Aiuta ad identificare i dati logici referenziati ma non mantenuti dall'applicazione(EIF).
- ◆ Dipende dal punto di vista applicativo esterno che ha l'utente sull'applicazione. E' indipendente da considerazioni tecniche e/o di natura implementativa.

Scope + boundary - esempio



Identificare le funzioni

- ◆ Di tipo dati
 - File logici interni (ILF)
 - File esterni logici (EIF)
- ◆ Di tipo transazionale
 - Input (EI)
 - Interrogazioni (EQ)
 - Output (EO)

Funzioni di Tipo Dati

- Gruppi logici di informazione, e vanno conteggiati secondo le loro caratteristiche logiche e non l'implementazione fisica (forma cartacea, requisiti, specifiche interne, manuali).

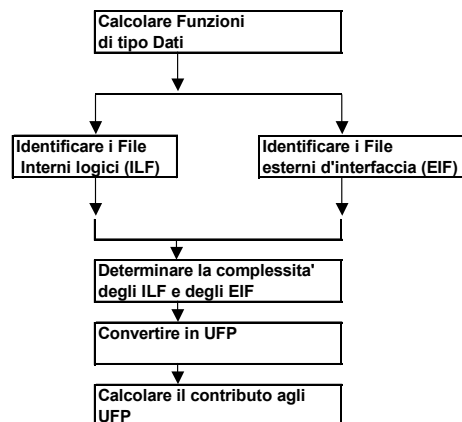
- ◆ ILF – File Interno Logico

gruppo di dati logicamente collegati o di informazioni di controllo, riconoscibili dall'utente, mantenuti all'interno del confine dell'applicazione. L'intento primario di un ILF è contenere dati mantenuti da uno o più processi elementari dell'applicazione sottoposta al conteggio.

- ◆ EIF – File esterno d'interfaccia

gruppo di dati logicamente collegati o di informazioni di controllo, riconoscibili dall'utente, referenziati dall'applicazione ma mantenuti all'interno del confine di un'altra applicazione. L'intento primario di un EIF è contenere dati referenziati da uno o più processi elementari situati nei confini dell'applicazione sottoposta al conteggio.

Procedura Funzioni di tipo dati



Elementi di Complessità

- La complessità funzionale di ILF ed EIF è basata sul numero di DET e RET associati ad ogni ILF/EIF

DET RET

- ♦ DET -> Data - Element Type = Un tipo di elemento dati è un campo unico riconoscibile dall'utente, non ripetuto
- ♦ RET -> Record Element Type = Un tipo di elemento record è un sottogruppo, riconoscibile dall'utente, di elementi dati in un ILF o EIF. Ci sono due tipi di sottogruppi:
 - Opzionali: l'utente ha la possibilità di usarne uno o nessuno durante un processo elementare che crea o aggiunge un'occorrenza di dati
 - Obbligatori: l'utente deve utilizzarne almeno uno

ILF / EIF - Tabelle di conversione

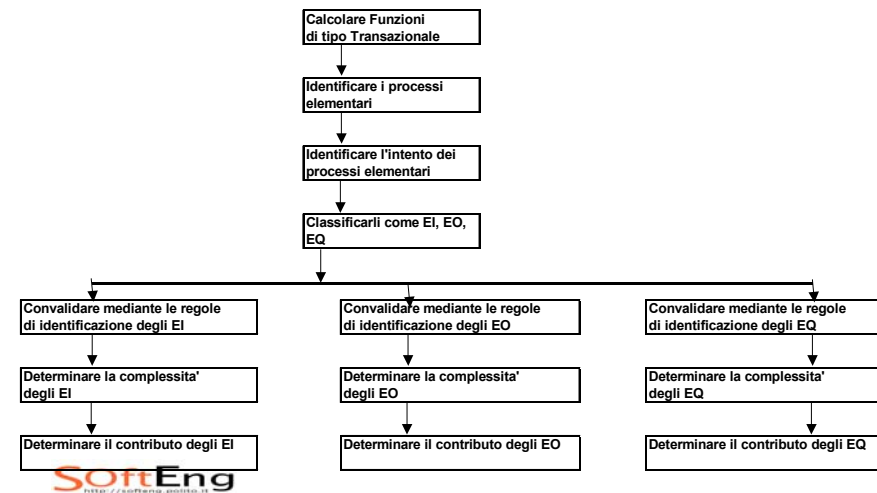
RET\DET	1-19	20 - 50	> 50
1	Bassa	Bassa	Media
2 - 5	Bassa	Media	Alta
> 5	Media	Alta	Alta

ILF		EIF	
Complessità	UFP	Complessità	UFP
Bassa	7	Bassa	5
Media	10	Media	7
Alta	15	Alta	10

Funzioni di Tipo Transazionale

- ♦ EI External Input – Input Esterno
- ♦ EO External Output – Output Esterno
- ♦ EQ External Inquiry – Interrogazione Esterna

Procedura



External Input

- EI – External Input: è un processo elementare che elabora dati o informazioni di controllo che provengono dall'esterno del confine dell'applicazione.
- **Intento primario:** mantenere uno o più ILF e/o alterare il comportamento del sistema.

External Output

- EO – External Output è un processo elementare che invia dati o informazioni di controllo all'esterno del confine della applicazione.
- **Intento primario:** presentare informazioni ad un utente, utilizzando un trattamento logico diverso da, o in aggiunta a, il semplice reperimento di dati o informazioni di controllo. Il trattamento logico deve contenere almeno una formula matematica o un calcolo, creare dati derivati, mantenere uno o più ILF o alterare il comportamento del sistema.

External Inquiry

- EQ – External Inquiry è un processo elementare che invia dati o informazioni di controllo all'esterno del confine della applicazione.
- **Intento primario:** presentare informazioni ad un utente attraverso il reperimento di dati o informazioni di controllo da un ILF o EIF. Il trattamento logico non deve contenere formule matematiche o calcoli e non crea dati derivati. Nessun ILF è mantenuto durante l'elaborazione e il comportamento del sistema non risulta modificato.

EI, EO, EQ

La differenza principale fra i tipi di funzione transazionali e' il loro intento primario.

FUNZIONALITA'	TIPO DI FUNZIONE TRANSAZIONALE		
	EI	EO	EQ
ALTERARE IL COMPORTAMENTO DEL SISTEMA	IP	P	V
MANTENERE UNO O PIU' ILF	IP	P	V
PRESENTARE INFORMAZIONI A UN UTENTE	P	IP	IP

Legenda:

IP = Intento Primario del tipo di funzione

P = Possibile

V = Vietato

Trattamento logico

- Insieme dei requisiti specificatamente richiesti dall'utente per completare un processo elementare. Questi requisiti possono includere le seguenti azioni (CPM 7-8):
 - ♦ Eseguire validazioni
 - ♦ Eseguire formule matematiche e calcoli
 - ♦ Convertire valori equivalenti
 - ♦ Filtrare e selezionare dati utilizzando criteri specifici per confrontare insiemi multipli di dati
 - ♦ Analizzare condizioni per determinare quali siano applicabili

Trattamento logico

- ♦ Aggiornare almeno un ILF
- ♦ Referenziare almeno un ILF o EIF
- ♦ Reperire dati o informazioni di controllo
- ♦ Creare dati derivati
- ♦ Alterare il comportamento del sistema
- ♦ Preparare e presentare informazioni all'esterno del confine
- ♦ Capacità di accettare dati o informazioni di controllo che entrano nel confine dell'applicazione
- ♦ Ordinare o riorganizzare un insieme di dati (da solo non è sufficiente per l'identificazione del tipo o l'unicità di una funzione di tipo transazionale)

Definizioni di FTR e DET

▪ FTR:

Un *tipo di file referenziato* (FTR) è

- ♦ Un file logico interno letto o mantenuto da una funzione di tipo transazionale, ovvero
- ♦ Un file d'interfaccia esterno letto da una funzione di tipo transazionale

▪ DET:

Un *tipo di elemento dati* (DET) è un campo unico, riconoscibile dall'utente, non ripetuto.

Tablelle conversione

EI - MATRICE DI COMPLESSITA'

FTR\DET	1-4	5-15	> 15
0-1	Bassa	Bassa	Media
2	Bassa	Media	Alta
> 2	Media	Alta	Alta

EO/EQ - MATRICE DI COMPLESSITA'

FTR\DET	1-5	6-19	> 19
0-1	Bassa	Bassa	Media
2-3	Bassa	Media	Alta
> 3	Media	Alta	Alta

EI / EQ	
Complessità	UFP
Bassa	3
Media	4
Alta	6

EO	
Complessità	UFP
Bassa	4
Media	5
Alta	7

Calcolo UFP

- Pari alla somma degli UFP per ogni funzione di tipo dati e di tipo transazionale identificato

Calcolo VAF (Value Adjustment Factor)

- Corregge $\pm 35\%$ il valore di UFP in base a 14 caratteristiche generali del sistema (GSC, General System Characteristics)
- I gradi di influenza per ogni caratteristica variano in una scala da zero a cinque, da nessuna influenza a forte influenza.

nb: Il VAF seppure ancora incluso negli argomenti di esame spesso non è usato in quanto non è conferme all'ISO (International Organization For Standardization ISO/IEC 14143)

GSC

General System Characteristics (GSCs)	Degree of Influence (DI) 0 - 5
1. Data Communications	—
2. Distributed Processing	—
3. Performance	—
4. Heavily Used Configuration	—
5. Transaction Rates	—
6. Online Data Entry	—
7. Design for End User Efficiency	—
8. Online Update	—
9. Complex Processing	—
10. Usable in Other Applications	—
11. Installation Ease	—
12. Operational Ease	—
13. Multiple Sites	—
14. Facilitate Change	—
Total Degree of Influence (TDI)	—
Value Adjustment Factor (VAF)	—
$VAF = (TDI * 0.01) + 0.65$	

Calcolo del VAF

1. Sommare i gradi di influenza di tutte le 14 caratteristiche generali del sistema per ottenere il grado di influenza totale (TDI, Total Degree of Influence).
2. Sostituire il TDI nella seguente formula per ottenere il fattore di aggiustamento del valore.

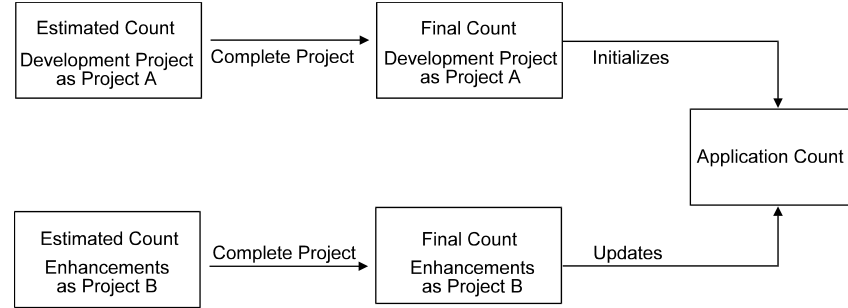
$$VAF = (TDI \times 0,01) + 0,65$$

Per esempio, il seguente fattore di aggiustamento del valore è calcolato quando per ciascuna delle descrizioni relative alle 14 GSC il grado di influenza vale tre (3×14).

$$VAF = (42 \times 0,01) + 0,65$$

$$VAF = 1,07$$

Tipi di count



Estimated count – development

$$DFP = (UFP + CFP) \times VAF$$

DFP = numero dei FP per un progetto di sviluppo

UFP = numero dei FP non pesati delle funzioni che saranno disponibili dopo l'installazione

CFP = FP non pesati aggiunti dal conteggio dei FP non pesati provenienti dalle funzionalità di conversione

VAF = fattore di aggiustamento del valore

Estimated count – enhancement

$$EFP = ((ADD + CHGA + CFP) \times VAFA) + (DEL \times VAFB)$$

EFP = risultato conteggio evolutivo

ADD = UFP delle funzioni aggiunte nella manutenzione

CHGA = UFP delle funzioni modificate

CFP = FP per le funzioni di conversione dati

VAFA = il VAF del sistema considerato dopo la manutenzione

DEL = gli UFP delle funzionalità cancellate

VAFB = il VAF considerato prima della manutenzione

Final count

- due varianti di calcolo.
 - ♦ Iniziale (dopo il primo sviluppo)
 - ♦ dopo successive manutenzioni

Final count – initial

- Iniziale

$$AFP = ADD \times VAF$$

AFP = conteggio applicazione iniziale

ADD = UFP funzionalità installata con il progetto di sviluppo

Final count – enhancement

- *dopo la Manutenzione* quando tale attività ha aggiunto, cambiato, modificato funzioni dati ed il VAF:

$$AFP = (UFPB + ADD + CHGA) - (CHGB + DEL) \\ \times VAFA$$

AFP = numero dei FP pesati dell'applicazione

UFPB = UFP prima manutenzione

ADD = UFP delle funzioni aggiunte nella manutenzione

CHGA = UFP delle funzioni modificate

CHGB = UFP delle funzioni cambiate prima manutenzione

DEL = gli UFP delle funzionalità cancellate

VAFA = il VAF considerato dopo la manutenzione