

FUNCTION POINT

È fondamentale avere un'altra "unità di misura" al posto dei LOC durante le fasi iniziali del processo software per poter stimare la dimensione del software anche senza la presenza effettiva di righe di codice. Bisogna anche trovare una misura che possa essere comprensibile all'utente e che serva a mostrare una stima sui costi del progetto durante l'accettazione di un contratto da entrambi le parti coinvolte. I function point furono introdotti per poter quantificare le funzionalità che un software può offrire. Esse sono delle misure soggettive perché si basano su un documento semi-formale (linguaggio naturale per esprimere le specifiche) a differenza di altre misure come il numero ciclomatico o Halstead che operano su semilavorati formali (codice); che sono misure oggettive. Dopo varie proposte International Function Point Users Group standardizzò il loro uso attraverso linee guida che ridussero la loro soggettività sulla misura. I FP, basandosi sulle specifiche, possono essere discussi con l'utente già all'inizio dello sviluppo del software; ovviamente i FP sono più comprensibili all'utente a differenza dei SLOC. Gli FP possono essere usati anche per misurare la produttività in termini di quantità di funzioni (dell'applicazione) che possono essere consegnate all'utente (più interessato alle funzionalità del programma rispetto alla sua implementazione effettiva). I function point si basano sulle funzionalità richieste dell'utente e vengono calcolati attraverso 7 passi:

- Che tipo di FP devo contare
- Stabilire i confini del mio sistema (counting scope)
- I Data Function insieme alla loro complessità:
 1. Internal Logical Files
 2. External Interface Files
- Transactional Functions insieme alla loro complessità:
 1. External Input
 2. External Output
 3. External Inquiry
- Contare gli Unadjusted function point
- Definire i Value adjustment factor
- Calcolare gli Adjusted function point tramite i VAF

Ci sono tre tipi di function Point:

- Development project FP
- Enhancement project FP

- Application FP

I Development project FP sono usati alla prima installazione del software includendo anche i casi di aggiunta di funzionalità per rendere compatibili il software in altri sistemi o cambio delle specifiche durante lo sviluppo. L'Enhancement project FP viene adoperato in software già esistenti e tengono conto di tutti i cambiamenti ai function point (aggiunta, modifica o eliminazione). Infine Application FP viene usato per testare se realmente tutti i function point previsti (baseline count) sono stati realmente applicati (installed count).

Il counting scope è il confine dell'applicazione che viene definito tramite: la visione dell'utente che commissiona il progetto, per considerazioni di business (il confine sarà il singolo prodotto in una suite di prodotti) e un confine

iniziale che poi potrà modificarsi con l'aggiunta o la eliminazione di funzioni. Il counting scope non deve essere né troppo piccolo (non guardare i singoli moduli ma l'intero sistema per capirlo) e né troppo grande (non tutta la suite di programmi ma il guardare il singolo prodotto separatamente). Dopo avere contato tutti i data function e i transactional function si potrà calcolare gli unadjusted function point:

$$UFP = \sum_i \sum_j \#FunctionTypes_{ij} \cdot ComplexityWeight_{ij} \cdot Value\ Adjustment\ Factor$$

$$VAF = (0.65 + 0.01 \sum_k DI_k)$$

È una somma pesata dove per ognuno dei 5 tipi di funzioni:

- Calcolerò il prodotto di ogni Record/ Referenced Element Types appartenenti al gruppo di funzioni dello stesso tipo che è stato preso in esame partendo da quello con complessità minore (internal file, external input...)
- Poi sommerò tutte le complessità calcolate di ogni record
- Infine sommerò le varie complessità calcolate per ogni tipo di funzione

Il VAF sarà calcolato tramite i 14 General System Characteristics che permetteranno di regolare la misura degli unadjusted function point. Essi vanno da una scala da 1-5 in base al grado di influenza o presenza nel sistema di quella specifica caratteristica considerata, ed ogni caratteristica può influenzare il risultato del 5%. L'adjusted FP sarà calcolato come prodotto tra UFP e VAF.

I data function si occupano di dati logici che non descrivono come essi sono impelati fisicamente e infatti essi sono identificati da strutture logiche indipendenti dall'hardware. **Quindi gli internal**

logical file sono un gruppo identificabile dall'utente di dati logicamente correlati o informazioni di controllo, che sono "mantenuti" (modificati) all'interno dei confini dell'applicazione (counting scope).L'intento primario di un ILF è di trattenere i dati mantenuti attraverso uno o più processi elementari dell'applicazione. Un gruppo è identificabile dall'utente se è compreso e accettato sia dagli utenti che dagli sviluppatori. Un gruppo è formato da dati logicamente correlati che sono fortemente coesi quindi indipendenti da altri ILF (basso accoppiamento).I dati considerati sono un insieme di "fatti o cifre" mantenute all'interno della applicazione (data,id della trasazione,quantità della transazione...).Un gruppo può anche contenere informazioni di controllo che influenzano quando,come e che tipo di dati devono essere eseguiti da un processo elementare ed essi sono memorizzati come dati,regole o parametri. Un ILF deve essere "mantenuto" ossia modificato dalla applicazione e solo quelli modificati devono essere presi in considerazione e contati una volta per applicazione, anche se presentano copie di essi. Un ILF è mantenuto da un processo elementare, che è la più piccola unità di attività che è considerata sensata da parte di un utente. Gli ILF da non contare sono: i file temporanei e file introdotti per ragioni implementative e non riconoscibili dall'utente.

Gli external Interface file sono un gruppo identificabile dall'utente di dati logicamente correlato o informazioni di controllo, a cui l'applicazione fa riferimento e che sono "mantenuti" (modificati) all'interno dei confini di un'altra applicazione (counting scope).L'intento primario di un EIF è di trattenere i dati a cui fa riferimento attraverso uno o più processi elementari entro i confini dell'applicazione. Questo significa che una EIF di una applicazione deve essere una ILF di un'altra applicazione.

Sia gli ILF e gli EIF grazie alla loro complessità funzionale (valore tra low,medium and high che indica il grado di complessità) contribuiscono al calcolo dei function point, basandosi sul numero dei loro:

- Data Element Types
- Record Element Type

I DET sono degli univoci (non ripetuti) campi contenenti informazioni riconosciute dall'utente (user-recognizable) all'interno di sottogruppi di data element (RET) che sono a loro volta contenuti in un ILF o EIF. Il numero di istanze non ha influenza sul conteggio di un data element ma si conta il singolo DET mantenuto (ILF) o usato (EIF) dalla applicazione; si contano anche i campi richiesti dall'utente che sono usati per creare relazioni con altri ILF o EIF (foreign key).Un processo elementare potrebbe non necessariamente mantenere o usare tutti i RET ma deve

obbligatoriamente materne o usarne almeno uno.

Un external Input è un processo elementare che elabora un dato o un informazione di controllo che proviene dall'esterno del confine dell'applicazione. Un dato computato mantiene uno o più ILF; informazioni di controllo computate potrebbero o non potrebbero mantenere un ILF.

L'intento primario di un EI è di mantenere uno o più ILF e/o alterare il comportamento della applicazione attraverso la sua logica di elaborazione. La logica di elaborazione di un processo elementare è un insieme di operazioni che permettono di eseguire un formula matematica, un calcolo, creazione di dati derivati, aggiornamento di una ILF...un external input deve essere auto contenuto e deve lasciare il sistema della applicazione in uno stato consistente; esso deve ricevere dati o informazioni di controllo dall'esterno dell'applicazione. I file ILF che vengono mantenuti o letti o i EIF che vengono solo letti sono chiamati File Type Referenced, perché un processo elementare fa riferimento a quel file ma in particolare a specifici campi (data element type) di uno o più record in esso contenuti.

Un external Output è un processo elementare che invia dati o informazioni di controllo all'esterno dei confini della applicazione. L'intento primario è di rappresentare l'informazione all'utente attraverso una logica di elaborazione invece di, o in aggiunta di, recuperare i dati o informazioni di controllo. La logica di elaborazione deve contenere almeno una formula matematica o di calcolo, creare dati derivati, mantenere una o più ILF o alterare il comportamento di un sistema.

Un external inquiry è un processo elementare che invia dati o informazioni di controllo all'esterno dei confini della applicazione. L'intento primario è di rappresentare l'informazione all'utente attraverso il recupero di dati o informazioni di controllo da un ILF o EIF. La logica di elaborazione non contiene una formula matematica o di calcolo e non crea dati derivati; nessun ILF è mantenuto e non altera il comportamento di un sistema.