

**Produttività Sviluppo Soluzioni Software
*Consultazione di mercato***

Allegato 2
Tecniche di riferimento per la misurazione del software

Marzo 2021

1 Tecniche di riferimento per la misurazione

Premessa fondamentale è che la misurazione in Function Point è da intendersi riferita alla metodologia IFPUG alla quale si è aggiunta, a metà degli anni '90, la metodologia di misura funzionale cosiddetta COSMIC, che meglio si adatta a contesti quali quelli dei processi batch e/o di componenti a servizi.

Entrambe le metodologie sono metriche di prodotto, cioè misurano la dimensione funzionale di un prodotto software.

Nel dettaglio sono qui presentati:

- il perimetro di attività che possono considerarsi statisticamente proporzionali alla misura funzionale di un prodotto software (misurato in Function Point) e che, pertanto, influiscono sulla produttività dello sviluppo del prodotto medesimo;
- il perimetro di attività che non possono considerarsi statisticamente proporzionali alla misura funzionale di un prodotto software (misurato in Function Point) e che non influiscono sulla produttività del prodotto, ma sulla produttività del progetto;
- le linee guida c
- on le quali il CSI Piemonte misura la dimensione funzionale delle applicazioni nei vari contesti, applicando le metodologie di riferimento del mercato

Quanto di seguito esposto si applica alle seguenti tipologie di soluzione definite al cap.4.3 Tipologie di Soluzioni del documento “Produttività Sviluppo Soluzioni Software Consultazione di mercato”:

- Soluzioni gestionali (MIS);
- Soluzioni Data Integration;
- Soluzioni Front-end Servizi Digitali;
- Soluzioni Back-end API/Microservizi.

1.1 Riferimenti bibliografici

- [BIB001] GUFPI-ISMA: Best Practices Contrattuali – Vol 1: Principi ed assunzioni
[BIB002] IFPUG: Counting Practice Manual (Version 4.3.1)
[BIB003] The COSMIC Functional Size Measurement Method (Version 3.0.1)
[BIB004] Early & Quick Function Points per il metodo IFPUG (Versione 3.1)

1.2 Attività inerenti i requisiti funzionali

Le attività sulle quali viene richiesto di rilevare la produttività e che possono essere considerate proporzionali alla dimensione funzionale sono le seguenti:

- raccolta dei requisiti funzionali;
- analisi di dettaglio dei requisiti funzionali del software (espressa comunemente attraverso casi d’uso di dettaglio);
- definizione modello logico dei dati;
- progettazione del database e dell’architettura applicativa di dettaglio;

- implementazione del database e delle componenti sw (compreso unit test e test di integrazione delle componenti sw);
- test funzionale (con test funzionale si intendono i test che verificano il rispetto dei requisiti funzionali del software);
- bug fixing.

Queste attività sono quelle previste dal così detto “ciclo completo”.

Può essere richiesta ad un fornitore solo parte della realizzazione delle attività di cui sopra, quelle del così detto “ciclo realizzativo”.

Le attività **comprese** nel “ciclo realizzativo” e comprese nella misura in Function Point sono:

- progettazione del Database e dell’architettura applicativa di dettaglio;
- implementazione del Database e delle componenti software (compresi unit test e test di integrazione);
- bug fixing.

Dato il ciclo completo come il 100% del peso delle attività comprese nel Function Point, nel contesto CSI Piemonte le attività del ciclo realizzativo costituiscono circa il 50% di tale peso.

1.3 Attività inerenti i requisiti non funzionali

Le attività da non considerare nella produttività richiesta sono:

- attività di coordinamento e governo del progetto (svolte prevalentemente dal capo progetto);
- attività di predisposizione dell’ambiente (preparazione delle macchine per il test, installazione dei pacchetti software);
- attività di collaudo del prodotto;
- attività di un’eventuale progettazione del servizio, qualora si sviluppi in un paradigma DevOps;
- attività inerente test di carico e di performance;
- attività di migrazione dei dati da una base dati precedente a quella attuale;
- attività non misurabili dal punto di vista funzionale (ad esempio gestione delle tabelle di decodifica).

L’effort per queste attività viene valutato in figure professionali valorizzate a giorni uomo.

2 Aspetti generali di misurazione

I metodi di misurazione funzionale del software utilizzati in CSI Piemonte sono:

- IFPUG Function Point [BIB002] per le funzioni online;
- Cosmic Function Point [BIB003] per i batch e i servizi esposti in cooperazione applicativa.

I manuali ufficiali dei metodi IFPUG ([BIB002]) e COSMIC ([BIB003]) costituiscono la base di riferimento per le regole da applicare in un qualsiasi conteggio di Function Point.

Si ritiene opportuno integrare questi riferimenti con l'illustrazione di alcuni aspetti di misurazione adottati nel contesto CSI, per una migliore valutazione ai fini dell'indagine.

2.1 Livello di precisione della misurazione

Vi possono essere due livelli di misurazione: il conteggio e la stima.

Nel caso del conteggio lo scopo è quello di determinare la dimensione funzionale per un progetto di sviluppo software e/o di manutenzione evolutiva. L'ambito del conteggio è l'insieme delle funzionalità da sviluppare per consultare, aggiungere, modificare e cancellare, secondo il confine applicativo definito in base al punto di vista dell'utente.

Il presupposto per il conteggio è che esistano tutti gli elementi necessari, a fronte di requisiti funzionali completi e sufficientemente dettagliati.

Nel caso della stima invece, tipicamente ad inizio progetto, i requisiti funzionali non sono sufficientemente dettagliati per effettuare un conteggio in Function Point: viene fatta una stima (per quanto riguarda il metodo IFPUG la stima viene fatta applicando il metodo Early & Quick [BIB004]).

Sarà necessario fare un conteggio in un momento successivo, quando i requisiti diventeranno sufficientemente dettagliati.

2.2 Utenti, confini e strati architetturali

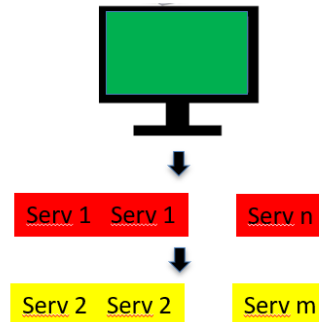
Per utente si intende "ogni persona o oggetto che comunica o interagisce con il software in ogni istante" [BIB002].

Nel caso di uno sviluppo di un servizio applicativo per un Ente, un utente è sicuramente il referente del Sistema Informativo dell'Ente, ovvero colui che definisce i requisiti funzionali. Altri utenti sono le applicazioni esterne con cui il sistema in esame deve interagire (inviando e/o ricevendo dati).

Per le applicazioni che non presentano particolari requisiti rispetto alla architettura interna, il confine si colloca coerentemente con il punto di vista di questi utenti e non si considerano, nella misurazione, aspetti legati all'architettura interna dell'applicazione.

Quando invece è richiesto, come requisito esplicito, lo sviluppo di un sistema può essere articolato in vari "strati" architetturali, come, ad esempio, gli strati VERDE, ROSSO e GIALLO della architettura prevista da diversi applicativi del CSI Piemonte:

- strato VERDE = applicazioni di front-end con funzionalità di interfaccia utente,
- strato ROSSO = applicazioni che offrono servizi di orchestrazione
- strato GIALLO = applicazioni che implementano logica di business/persistenza.



I servizi degli strati interni costituiscono una sorta di “libreria di funzioni” potenzialmente fruibili anche da sistemi esterni, rispondendo a precisi requisiti di riuso e manutenibilità, ciascuno di questi servizi, detti anche “microservizi”, è riconosciuto come un processo elementare indipendente e quindi misurato:

- le funzionalità che costituiscono le GUI verso utenti persone sono misurate in IFPUG (es. strato VERDE),
- i servizi fruiti, o potenzialmente fruibili, da sistemi esterni, compresi quelli degli strati architetture interni, sono misurati in COSMIC (es. strati ROSSO e GIALLO).

3 Specificità di misurazione

3.1 Processi BATCH

I processi Batch, indipendentemente dal linguaggio di programmazione utilizzato e dal fatto che possano o meno prevedere una interfaccia di attivazione, sono misurati con la metodologia COSMIC [BIB003].

3.2 Servizi esposti a sistemi esterni

I servizi esposti, o potenzialmente esponibili, a sistemi esterni, sono misurati con la metodologia COSMIC [BIB003] (indipendentemente dal linguaggio di programmazione utilizzato).

3.3 Operazione di Login e sistemi di autenticazione

Il Login è misurato come un processo elementare di External Input (EI), secondo la metodologia IFPUG [BIB002]. Come intento primario si intende quello di modificare il comportamento del sistema. Come file referenziati non sono considerati eventuali file di sistema (esempio: Clock).

Per ogni sistema di autenticazione differente (ad es. Shibboleth/Iride, SPID, CIE) si considera l'accesso a File Logici differenti. Ad ognuno di essi sarà associato un differente

Login (EI). Eventuali File Logici mantenuti dall'applicazione saranno misurati come ILF (Internal Logical File).

3.4 Operazione di Logout

Laddove vi sia un'operazione di logout con azioni particolari da compiere (ad esempio la registrazione della data di uscita dell'utente in un file di log) e queste facciano esplicitamente parte dei requisiti funzionali richiesti dall'utente, l'operazione è misurata come un processo elementare di External Input secondo la metodologia IFPUG [BIB002].

3.5 Gestione di una cartella di contenuti

Nelle applicazioni gestionali è assai comune avere la gestione di una serie di informazioni, trattate come un fascicolo con i relativi faldoni. Il fascicolo è il contenitore, con una testata e una serie di faldoni che contengono il dettaglio di informazioni differenti. Un esempio tipico è la gestione di una pratica, con una testata di informazioni generali e una serie di informazioni a corollario.

Dal punto di vista implementativo tale configurazione è realizzata di solito con una videata con una serie di "tab", ciascuno dei quali contiene un preciso insieme di informazioni (potrebbe costituire un "faldone").

Nel caso in cui ogni faldone abbia una gestione indipendente, con esplicite operazioni di CRUD – Create, Read, Update e Delete, si misurano in IFPUG [BIB002] un EI di inserimento, un EI di modifica, un EI di elimina, un External Query (EQ)/External Output (EO) di visualizzazione per ciascun "faldone" con tali caratteristiche.

Se non si hanno esplicite operazioni di inserimento e di eliminazione, ma vi è una indipendenza funzionale che ne permette la modifica (con il salvataggio al passaggio al "tab" successivo) e la relativa visualizzazione, si misurano in IFPUG [BIB002] un EI di modifica, un EQ/EO di visualizzazione per ciascun "faldone" con tali caratteristiche.

Se non esiste alcuna indipendenza funzionale nella gestione dei faldoni, ma il fascicolo è visto come un "tabellone" di informazioni e la modalità di layout è determinata solo dalla numerosità delle informazioni presenti, si misura in IFPUG [BIB002] un EI di inserimento, un EI di modifica, un EI di eliminazione (qualora presente), un EQ/EO di visualizzazione per ogni "tab" visualizzato.

3.6 Output su device multipli

Si considera un unico processo elementare nel caso in cui il processo di reperimento dei dati, con relativo trattamento logico, sia lo stesso per tutti i Media (es. CSV, XML, PDF, ...) utilizzati e sia lo stesso anche il numero di attributi che attraversano il confine, secondo la metodologia IFPUG [BIB002]. Per gli n-1 media non contati in FP, si valuterà un effort in gg/u.

Se invece il trattamento logico è differente per i diversi media e/o sono differenti i dati interessati, la misura prevede più processi elementari distinti, uno per media.

3.7 Storicizzazione dei dati

In alcuni casi un requisito richiede la storicizzazione di dati su alcune entità a fronte di una modifica degli stessi.

Distinguiamo due casi:

1. la storicizzazione crea una copia identica dei dati attuali, solo non modificabile. I dati storici non costituiscono un File Logico differente rispetto a quello dei dati attuali bensì un RET (Record Element Type) in più in questo File Logico. Se i dati storici presentati sono gli stessi di quelli attuali, non si ha alcuna transazione di presentazione dati aggiuntiva; viceversa, se i dati storici presentati, sulla base di un esplicito requisito utente, sono un sottoinsieme dei dati correnti, si hanno delle transazioni di presentazione dati distinte;
2. la storicizzazione crea dati diversi da quelli attuali (es. dati di sintesi). In questo caso i dati storici hanno una indipendenza e sono calcolati come File Logici indipendenti, con relative funzioni di gestione.

3.8 File di LOG/Tracing

Nel caso in cui vi sia un requisito utente che richieda la presenza di un file di log e/o di tracciatura, con la precisa definizione delle informazioni che in esso devono essere contenute, queste costituiscono un insieme logico di dati, coerente e riconosciuto dall'utente.

Pertanto si misura un ILF di LOG [BIB002] nel caso di logging di funzioni online, e in tutti i processi elementari deputati a loggare e/o tracciare determinate informazioni, il relativo ILF di LOG costituisce un FTR (File Type Referenced) aggiuntivo.

Se tali informazioni sono consultabili tramite l'applicazione queste visualizzazioni saranno misurate con un EQ.

In funzioni misurate con il metodo COSMIC [BIB003] si misurano movimenti di Write per il logging.

Qualora invece i file di LOG e/o di tracciatura non siano esplicitamente richiesti dall'utente, ma siano semplicemente file di log di sistema e non vi siano funzioni esplicite nell'applicazione per visualizzarli ma siano consultabili solo da sistemisti attraverso strumenti ad hoc, essi non sono misurati.

3.9 Regole di business

Quando sono presenti delle regole di business definite dall'utente (ad esempio calcoli, validazioni), non strettamente legate ad uno specifico processo elementare ed invece da applicarsi in più punti dell'applicazione a fronte di ben definite azioni, e che pertanto hanno una loro indipendenza e coerenza funzionale, esse sono viste come processi elementari indipendenti.

Se sono misurate in IFPUG [BIB002], sono degli EO, in quanto il loro intento primario è di "presentare un risultato e/o di alterare il comportamento del sistema".

Sono misurate in COSMIC [BIB003] solo quando sono utilizzate all'interno di processi Batch e/o servizi web services.

3.10 Regole di Permission

Nel caso in cui ogni funzionalità, prima di essere attivata, esegua il controllo se sia o meno attivabile per l'utente, il processo elementare relativo alla funzionalità ha come file referenziato aggiuntivo quello dei Profili e, nel caso sia presente, anche quello relativo alle Funzionalità, secondo le regole IFPUG [BIB002].

Nel caso in cui i controlli siano fatti all'atto del Logon/Login al fine di nascondere le funzionalità non attivabili dal profilo dell'utente (non facendole comparire nel menu), i file referenziati Profili ed eventualmente Funzionalità sono associati alla funzione di Logon/Login.

3.11 Memorizzazione temporanea (nella sessione di ricerca) dei filtri di ricerca

Un requisito utente può richiedere che dopo che si sia fatta una operazione di ricerca o di inserimento, per una ulteriore operazione si possano ritrovare i dati inseriti precedentemente.

Viene misurato un ILF per la memorizzazione dei dati: questi sono l'insieme dei dati digitati, anche attraverso più videate; i RET sono le eventuali informazioni multiple digitate (ad esempio più persone associate ad un inserimento). Si misurano inoltre un EI per la loro memorizzazione e un EQ per la loro visualizzazione (ripristino).

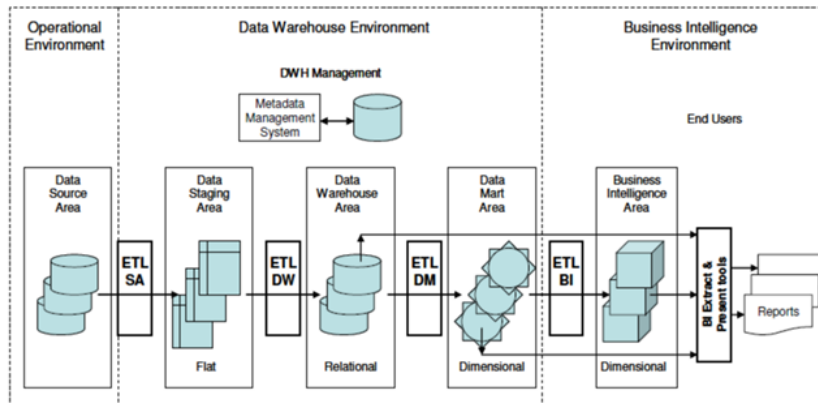
3.12 Help

Nel caso in cui l'help di un'applicazione non sia modificabile attraverso delle funzioni applicative, ma sia statico, esso non costituisce un file logico e dunque non è misurabile in Function Point.

Nel caso in cui l'help sia gestito da una applicazione (sia cioè dinamico), esso costituisce un file logico (ILF, se gestito dall'applicazione stessa o EIF, se gestito da una applicazione esterna) e sono misurate le relative funzioni transazionali per mantenerlo (se interno) o solo per visualizzarlo, se esterno. Questa regola si applica a ciascuna tipologia di help (generale, contestuale, di campo), secondo le regole IFPUG [BIB002].

3.13 Misurazione di un sistema di dataintegration/data warehouse

Nella figura sottostante è rappresentato il flusso di trasformazioni (non necessariamente tutte presenti) in un Data Warehouse.



In tale schema si possono distinguere due fasi distinte: la prima di gestione e trasformazione del dato (detta anche fase di Preparation o Ingestion), la seconda tipicamente di presentazione dei dati (detta fase di Presentation).

Le funzioni di ETL previste dalla fase di Preparation sono misurate applicando il metodo COSMIC ([BIB003]).

La fase di Presentation consiste in reportistica spesso realizzata con l’ausilio di tools. Nel contesto CSI Piemonte si è rilevato che la misurazione funzionale di questi artefatti non porta a risultati utili per stimare in modo attendibile i relativi effort/costi di realizzazione. Pertanto questa parte non sarà quotata in Function Point bensì in figure professionali valorizzate in giorni uomo.

3.14 Cambiamenti in corso d’opera

Nel corso della realizzazione di un progetto software è normale che vi siano dei cambiamenti nei requisiti, o per approfondimento di requisiti che all’inizio non erano sufficientemente chiari, o per reali cambiamenti (ad esempio cambiamento di un impianto legislativo).

In termini di misurazione, nel primo caso essi fanno parte dell’indeterminatezza della stima iniziale, comportano una revisione del perimetro funzionale del progetto che viene intercettata, se non prima, al momento del conteggio. Questo fenomeno è chiamato “Scope Creep” e, di solito, determina, alla fine della realizzazione, un aumento delle funzionalità stimate inizialmente. Obiettivo di una buona stima iniziale è contenere l’ammontare di tale fenomeno, auspicabilmente non oltre il sesto quinto (20%) del totale.

Nel secondo caso invece, se i cambiamenti comportano un reworking su funzionalità già in fase di realizzazione (ad esempio già in codifica) o addirittura realizzate, le modifiche di tipo funzionale possono essere misurate in Function Point come funzionalità modificate (CHG) e/o cancellate (DEL). Le variazioni verranno misurate in IFPUG o COSMIC a seconda dell’elemento interessato.

Qualora invece venissero sviluppate funzionalità non richieste dai requisiti iniziali e non previste, queste non saranno riconosciute al fornitore.

3.15 Riuso

Un caso comune è quello in cui in un sistema viene integrato un software open source che eroga funzionalità agli utenti. In questo caso l'effort non è legato alla realizzazione delle funzionalità (che sono già presenti nel sw open source), ma piuttosto ad attività di integrazione e parametrizzazione del software open source. Questo effort tendenzialmente non dipende dalla dimensione funzionale delle funzioni erogate dal software open source e quindi i Function Point non sono un parametro utile per stimarlo; occorre invece una valutazione in giorni uomo.

Restano invece valutabili in FP le altre funzionalità da realizzare, non coperte dall'open source.

Non ricadono in questo perimetro i progetti software che utilizzano framework open source ad un livello architetturale più basso (ad es. framework quale Angular o di gestione della persistenza quale Hibernate).

3.16 Sviluppo di Stub per Test di integrazione

Nel caso in cui, al fine di effettuare test di integrazione con altri prodotti, sia necessario implementare degli "Stub" o "pezzi di codice" per simulare il prodotto da integrare, tale codice può essere misurato in COSMIC come funzioni esclusivamente ad uso del progetto, che non rientrano nella baseline di prodotto.

3.17 Realizzazione di siti web mediante CMS

Si ritiene che, nel contesto CSI Piemonte, il dimensionamento funzionale di siti web realizzati con strumenti CMS – Content Management System, non sia utile a determinare l'effort di realizzazione di tali siti. Potrebbe invece essere misurato in Function Point un progetto per la realizzazione di un CMS.

3.18 Soluzioni ECM/BPM/Case Management

Si ritiene che, nel contesto CSI Piemonte, i Function Point non siano lo strumento adatto per stimare l'effort di configurazione e personalizzazione di piattaforme commerciali o opensource di ECM/BPM.