



# **Manuale di Misurazione COSMIC per ISO 19761**

## **Parte 2 Linee Guida**

**Versione 5.0**  
**Maggio 2020**  
(revisione: Febbraio 2021)

## **Premessa**

Il Manuale di Misurazione COSMIC per ISO/IEC 19761:2011 consiste di 3 parti:

Parte 1: Principi, Definizioni & Regole (17 pagine) (\*)

Parte 2: Linee Guida (17 pagine) (\*)

Parte 3: Esempi dei concetti e di misurazioni COSMIC (28 pagine)

(\*) Le parti 1 e 2 costituiscono tutto e il solo materiale necessario per la certificazione.

Questa Parte 2 del Manuale di Misurazione COSMIC presenta l'insieme di Linee Guida sviluppate dal COSMIC Group per favorire l'accuratezza, la ripetibilità e la riproducibilità dei risultati delle misurazioni COSMIC.

Febbraio 2021 – revisione minore:

- aggiunta la locuzione 'non c'è nulla di assoluto riguardo...' nelle linee guida per l'identificazione degli utenti funzionali e dei gruppi di dati,
- spostata la definizione di 'tipo' dalla sezione 3.4 (linea guida per l'identificazione dei movimenti di dati) in una nuova sezione 1.3,
- sostituite le linee guida per l'identificazione dei Read e dei Write con una descrizione specifica.

Il Manuale di Misurazione COSMIC si basa sullo standard ISO/IEC 19761:2011, riconfermato nel 2019. Questa versione 5.0 rimpiazza la precedente versione 4.0.2, senza modificarne la sostanza in termini di definizioni, regole e linee guida per la misurazione.

Una versione di pubblico dominio del Manuale di Misurazione COSMIC e altre pubblicazioni, inclusa la versione originale in inglese, si possono trovare nella Knowledge Base del sito web [www.cosmic-sizing.org](http://www.cosmic-sizing.org).

## **Editori:**

Alain Abran, Ecole de technologie supérieure – University of Quebec (Canada)

Peter Fagg, Pentad (Regno Unito)

Arlan Lestherhuis (Paesi Bassi)

## **Altri membri del COSMIC Measurement Practices Committee:**

Diana Baklizky (Brasile),

Jean-Marc Desharnais, Ecole de technologie supérieure – University of Quebec (Canada)

Dylan Ren, Measures Technology LLC (Cina)

Bruce Reynolds, Telecote Research (USA)

Hassan Soubra, German University in Cairo (Egitto)

Sylvie Trudel, Université du Québec à Montréal - UQAM (Canada)

Frank Vogelesang, Metri (Paesi Bassi)

**Traduzione in italiano a cura di:** Luca Santillo, Agile Metrics (Italia)

Copyright 2021. Tutti i diritti riservati. The Common Software Measurement International Consortium (COSMIC). Si concede il permesso di copiare tutto o parte di questo materiale purché le copie non siano fatte o distribuite per scopi commerciali, siano citati il titolo della pubblicazione, il numero di versione e la data e sia data informativa che la copia avviene per concessione del Common Software Measurement International Consortium (COSMIC). Copie di altro genere richiedono un permesso specifico.

# Sommario

<b>1. INTRODUZIONE</b> .....	<b>4</b>
1.1 Scopo del documento.....	4
1.2. Processo di misurazione COSMIC .....	4
1.3. Tipi ed occorrenze.....	4
<b>2. FASE DI STRATEGIA DELLA MISURAZIONE</b> .....	<b>4</b>
2.1 Panoramica della Fase di Strategia della Misurazione .....	4
2.2 Determinare scopo e ambito della misurazione .....	5
2.3 Identificare i FUR dai deliverable del software.....	5
2.4 Requisiti Non Funzionali.....	6
2.5 Identificare gli strati .....	7
2.6 Identificare gli utenti funzionali .....	8
2.7 Livelli di scomposizione.....	8
2.8 Diagrammi di contesto.....	9
2.9 Identificare il livello di granularità.....	9
<b>3. FASE DI MAPPATURA</b> .....	<b>10</b>
3.1 Mappare i FUR sul Modello Generale del Software .....	10
3.2 Identificare i processi funzionali.....	10
3.3 Identificare i gruppi di dati.....	11
3.3.1 Identificare i gruppi di dati.....	11
3.3.2 Sull'identificazione degli oggetti d'interesse e dei gruppi di dati.....	12
3.3.3 Dati o gruppi che non sono candidati validi di gruppi di dati .....	13
3.3.4 Identificare gli attributi dei dati (opzionale).....	13
3.4 Identificare i movimenti di dati .....	13
3.5 Misurare le componenti di un sistema software distribuito .....	15
3.6 Riuso del software.....	16
3.7 Misurare la dimensione delle modifiche del software.....	16
<b>4. ALTRI ARGOMENTI</b> .....	<b>17</b>
4.1 Estendere il Metodo di Misurazione COSMIC – Estensioni locali .....	17
4.2 Il Metodo di Misurazione COSMIC nel contesto Agile.....	17
<b>5. FASE DI MISURAZIONE</b> .....	<b>17</b>
<b>6. PRESENTARE LA MISURAZIONE</b> .....	<b>18</b>

## 1. INTRODUZIONE

### 1.1 Scopo del documento

Scopo di questo documento è fornire linee guida agli utilizzatori per l'applicazione dello standard COSMIC ISO 19761 Function Point, presentato nella Parte 1.

Questa Parte 2 è complementare alla Parte 1 ed è estesa dalla Parte 3, che presenta un gran numero di esempi.

### 1.2. Processo di misurazione COSMIC

Il processo di misurazione COSMIC presentato nella Parte 1 è riprodotto in Figura 1.1.

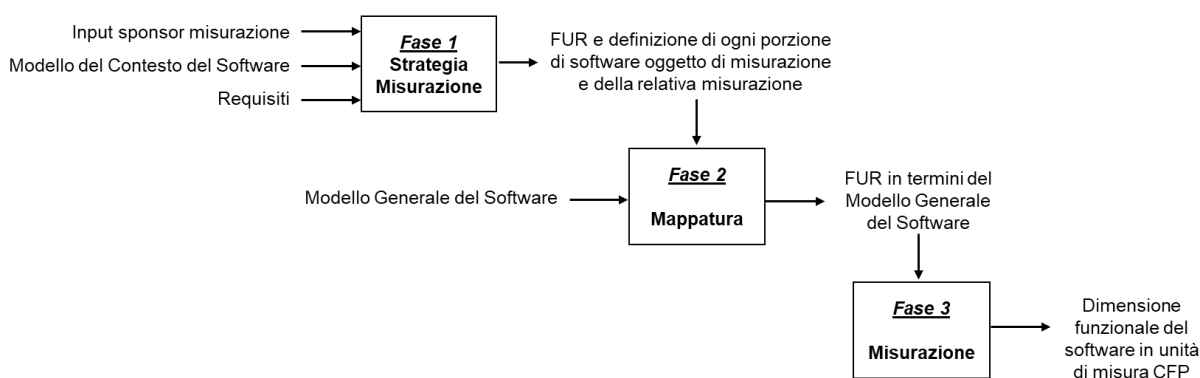


Figura 1.1 – Il processo del metodo di misurazione COSMIC.

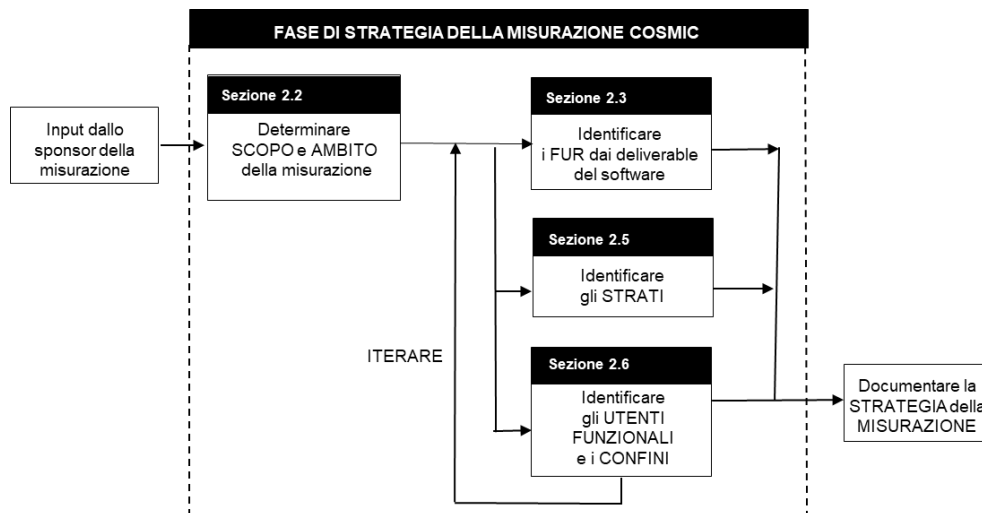
### 1.3. Tipi ed occorrenze

In generale, il 'tipo' di una 'cosa' è una classe astratta di tutte le 'cose' che condividono alcune caratteristiche in comune, cosicché tali 'cose' sono soggette ai medesimi FUR. (Sinonimi di 'tipo': 'categoria', 'genere'). Una 'occorrenza' di una 'cosa' si ha quando essa diviene reale, tramite l'assegnazione di valori specifici ai suoi attributi - nota nel mondo Object-Oriented come 'istanziamento', ossia la creazione di un'istanza.

## 2. FASE DI STRATEGIA DELLA MISURAZIONE

### 2.1 Panoramica della Fase di Strategia della Misurazione

La Figura 2.1 illustra graficamente la Fase di Strategia della Misurazione.



**Figura 2.1 – Il processo per determinare la Strategia della Misurazione.**

## 2.2 Determinare scopo e ambito della misurazione

Il termine 'scopo' è usato nel suo senso ordinario in linguaggio naturale. Lo scopo aiuta il misuratore a determinare:

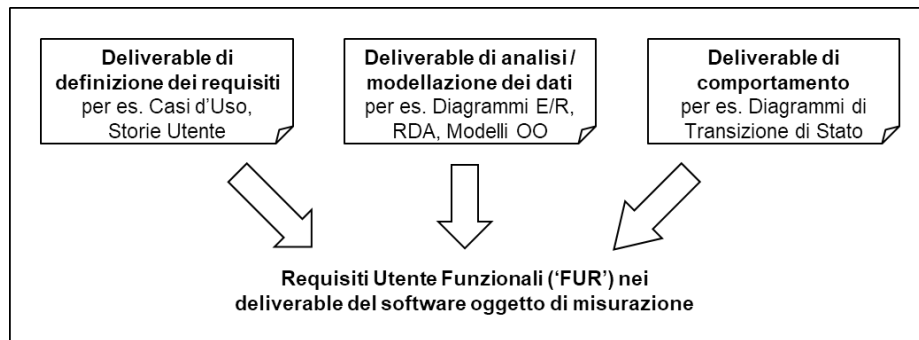
- l'ambito della misurazione e quindi i deliverable che saranno necessari per la misurazione,
- gli utenti funzionali,
- le modifiche funzionali,
- il momento temporale nel ciclo di vita del progetto in cui avrà luogo la misurazione,
- l'accuratezza richiesta della misurazione e quindi se si debba utilizzare il metodo COSMIC standard o una tecnica di approssimazione (per es. in una fase iniziale del ciclo di vita del progetto, prima che siano stati elaborati in maniera completa i FUR).

Come ausilio per determinare una strategia di misurazione, la Linea Guida 'Measurement Strategy Patterns' descrive, per ciascuno di vari diversi tipi di software, un insieme standard di parametri per misurare le dimensioni dei software, chiamato 'pattern di strategia della misurazione' (abbreviato in 'pattern di misurazione'). L'uso costante degli stessi pattern di misurazione dovrebbe aiutare i misuratori ad assicurarsi che misure con lo stesso scopo siano svolte in modo coerente, possano confrontarsi in modo affidabile con altre misure col medesimo pattern e che saranno interpretate correttamente in qualsiasi utilizzo futuro. Un effetto collaterale dell'uso di pattern standard pattern è che si reduce fortemente lo sforzo per determinare i parametri della Strategia di Misurazione.

Il COSMIC raccomanda che i misuratori studino e padroneggino il metodo COSMIC, in particolar modo i parametri della Strategia della Misurazione, prima di adottare i pattern standard.

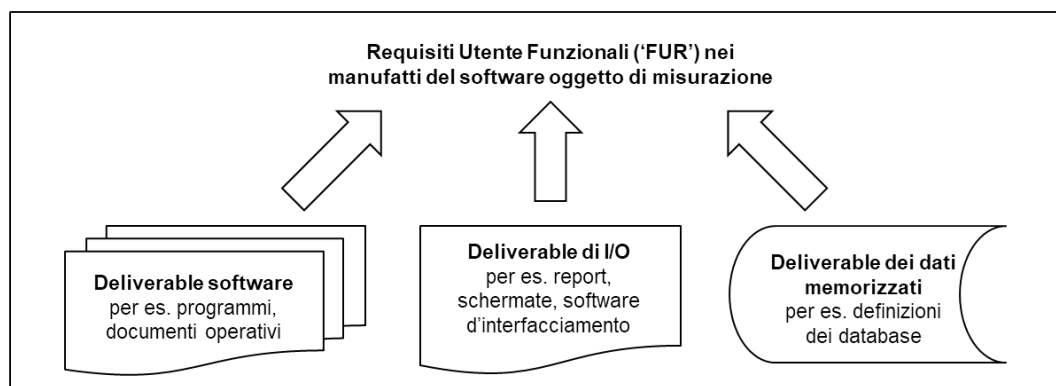
## 2.3 Identificare i FUR dai deliverable del software

Come mostrato in Figura 2.2, i FUR si possono derivare dai deliverable di ingegneria del software prodotti prima che il software sia realizzato. In tal modo, si può misurare la dimensione funzionale del software prima della sua implementazione in un sistema informatico.



**Figura 2.2 – Fonti dei Requisiti Utente Funzionali pre-implementazione.**

NOTA. Alcuni software esistenti potrebbero dover essere misurati in assenza di, o con solo pochi, deliverable architeturali o di progettazione disponibili e i requisiti funzionali potrebbero non essere documentati (per es. nel caso di software “legacy”). In tal caso, è ancora possibile ricavare i FUR dai deliverable del sistema informatico perfino dopo che sia stato implementato, come illustrato in Fig. 2.3.



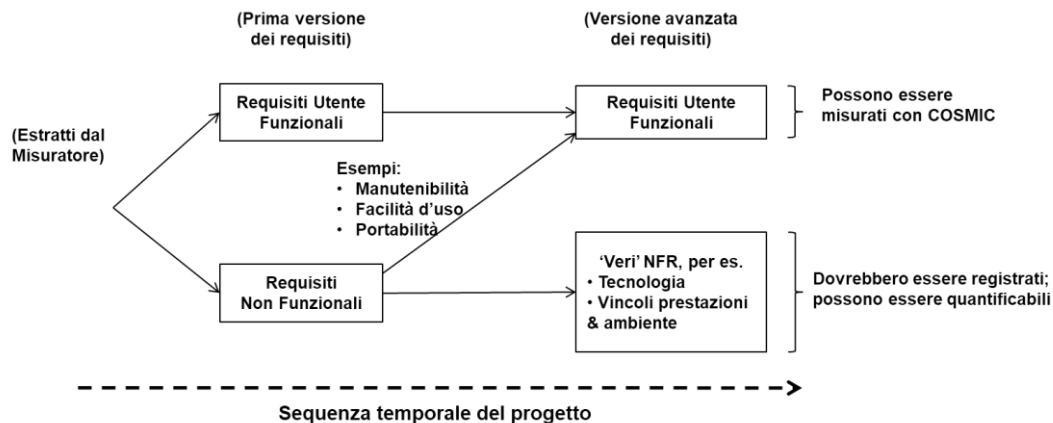
**Figura 2.3 – Fonti dei Requisiti Utente Funzionali post-implementazione.**

Ovviamente, il processo da usare e quindi l’impegno necessario per estrarre i FUR da diversi tipi di manufatti di ingegneria del software, o per derivarli dal software installato, variano da caso a caso e non possono essere affrontati nel Manuale di Misurazione. Il metodo COSMIC assume che i FUR del software oggetto di misurazione o esistono o possono essere estratti, o derivati, dai suoi manufatti, alla luce dello scopo della misurazione.

Se il misuratore comprende questi due modelli, è sempre possibile derivare i FUR di una porzione di software oggetto di misurazione dai suoi manufatti disponibili, sebbene potrebbero essere necessarie delle assunzioni per via di informazioni assenti o poco chiare.

## 2.4 Requisiti Non Funzionali

I Requisiti Non Funzionali (Non-Functional Requirements, NFR) del sistema possono variare significativamente in un progetto software. In casi estremi, un’affermazione nei requisiti di un sistema ‘software-intensive’ può richiedere tanta documentazione per la parte non funzionale quanta per quella funzionale. Il metodo COSMIC può essere utilizzato per misurare alcuni requisiti inizialmente espressi come non funzionali. Vari studi hanno mostrato come alcuni requisiti che appaiono inizialmente come NFR di sistema, al progredire del progetto evolvono in un mix di requisiti che possono essere implementati in funzionalità software e di altri requisiti o vincoli che sono realmente ‘non funzionali’, v. Fig. 2.4.



**Figura 2.4 – Molti requisiti apparsi inizialmente come NFR evolvono in FUR al progredire del progetto.**

Ciò è vero per molti vincoli di qualità del sistema, come il tempo di risposta, la facilità d'uso, la manutenibilità, etc. Una volta identificate, queste funzionalità del software, che erano state 'nascoste' nei NFR all'inizio del progetto, possono essere dimensionate con il metodo COSMIC, come qualsiasi altra funzionalità del software. Il non riconoscere questa dimensione funzionale 'nascosta' è una delle ragioni per cui la dimensione del software può sembrare crescere al progredire del progetto.

Una discussione più approfondita dei requisiti non funzionali del sistema e del software si trova in: ['Requisiti Non Funzionali e Dimensionamento COSMIC – Guida per Praticanti'](#).

## 2.5 Identificare gli strati

Le architetture software possono essere molto complesse, tuttavia il COSMIC adotta una vista molto semplificata dell'architettura del software, adeguata agli scopi delle misurazioni. È compito del misuratore attuare questa semplificazione. Questa sezione fornisce indicazioni in merito al misuratore.

Poiché l'ambito di una porzione di software oggetto di misurazione deve essere confinato in un solo strato software, il processo di definizione dell'ambito/i della FSM può richiedere al misuratore di stabilire in primo luogo quali siano gli strati dell'architettura del software. Questa sezione discute gli 'strati' del software nel senso del termine adottato nel metodo COSMIC, perché:

- il misuratore può trovarsi a dover misurare del software in un ambiente 'legacy' di software evolutosi nel corso di molti anni senza esser mai stato progettato seguendo una specifica architettura sottostante – il misuratore può quindi aver bisogno di supporto su come distinguere gli strati secondo la terminologia COSMIC;
- le espressioni 'strato' e 'architettura a strati' non sono usate in modo coerente nell'industria software. Quando il misuratore deve misurare un software descritto come collocato in una 'architettura a strati', è consigliabile verificare che gli 'strati' in questa architettura siano definiti in un modo che sia compatibile con il metodo COSMIC – per far ciò, il misuratore dovrebbe stabilire un'equivalenza tra specifici oggetti architetturali nel paradigma 'architettura a strati' e il concetto di strato come definito in questo manuale.

In un'architettura software definita, ogni strato deve avere le seguenti caratteristiche:

- a) Il software in un singolo strato fornisce un insieme di servizi che sono omogeni rispetto a un qualche criterio definito e che i software in altri strati possono utilizzare senza sapere come sono implementati tali servizi.
- b) La relazione tra i software in due strati qualsiasi è definita da una 'regola di corrispondenza' che può essere:

- o 'gerarchica', cioè al software nello strato A è permesso usare i servizi forniti dal software nello strato B, ma non viceversa (comunemente denotata come Client-Server);
  - o 'bidirezionale', cioè al software nello strato A è permesso usare il software nello strato B, e viceversa (comunemente denotata come peer-to-peer (P2P)).
- c) Il software in uno strato e il software in un altro strato si scambiano gruppi di dati tramite i loro rispettivi processi funzionali.
- d) Il software in uno strato non usa necessariamente tutti i servizi funzionali messi a disposizione dal software in un altro strato.
- e) Il software in uno strato di un'architettura software definita può essere partizionato in altri strati secondo una diversa architettura software definita.

#### **Guida alla Regola 4: Identificare gli strati**

Se l'ambito complessivo della FSM si estende su più strati, il misuratore dovrebbe procedere come segue.

- Se il software oggetto di misurazione è realizzato in un'architettura stabilita di strati che possono essere mappati sulle caratteristiche degli strati COSMIC sopra definite, allora si dovrebbe adottare tale architettura per identificare gli strati per scopi di misurazione.
- Se tuttavia lo scopo della misurazione richiede di misurare un software che non è strutturato secondo le caratteristiche degli strati COSMIC, il misuratore dovrebbe cercare di partizionare il software in strati applicando i principi sopra definiti.
- Per convenzione, i pacchetti software infrastrutturali, come i database management system (DBMS), i sistemi operativi (OS) o i driver dei dispositivi, che forniscono servizi che possono essere usati da altri software in altri strati, sono tutti collocati in strati separati.

Normalmente nelle architetture software lo strato 'più alto', cioè lo strato che non è subordinato a qualsiasi altro strato in una gerarchia di strati, è denotato come lo strato 'applicativo'. Il software in questo strato applicativo si appoggia sui servizi dei software in tutti gli altri strati per operare correttamente. Il software in questo strato 'più alto' può a sua volta essere stratificato, per es. come nell'architettura 'three tier' ('a tre strati') delle componenti di Interfaccia Utente, Regole di Business e Servizi sui Dati.

Una volta identificato, si può registrare ogni strato nel report della misurazione con la relativa etichetta.

## **2.6 Identificare gli utenti funzionali**

#### **Guida alla Regola 7: Identificare gli utenti funzionali**

L'identificazione del/degli utenti funzionali è determinata dai Requisiti 'Utente' Funzionali che devono essere misurati e dallo scopo della misurazione.

NOTA. Non c'è nulla di assoluto riguardo ad un utente funzionale, cioè si identifichino gli utenti funzionali per ciascun processo funzionale. Un mittente/destinatario di dati può essere l'utente funzionale di uno o più processi funzionali, ma non l'utente funzionale di altri processi funzionali, anche per il medesimo software oggetto di misurazione.

## **2.7 Livelli di scomposizione**

Le misurazioni della dimensione delle componenti di una porzione di software sono direttamente confrontabili tra loro solo per le componenti individuate al medesimo livello di scomposizione. Ciò è importante perché le dimensioni di porzioni di software a livelli di scomposizione diversi non si possono sommare tra loro senza tenere in considerazione le regole di aggregazione della sezione 5. Inoltre, come conseguenza, le performance di



progetto (per es. la produttività = dimensione / impegno) per sviluppare differenti porzioni di software possono essere confrontate tra loro se tutte le porzioni di software si trovano allo stesso livello di scomposizione.

Differenti livelli di scomposizione di una porzione di software possono corrispondere a ‘viste’ differenti degli strati del software, come per es. in Fig. 2.4 della Parte 3. Comunque, si può scomporre il software in ‘livelli’ a prescindere dal fatto che esso sia progettato utilizzando o meno un modello architetturale a strati.

## 2.8 Diagrammi di contesto

Nel definire un ambito della FSM e gli utenti funzionali, può essere di aiuto tracciare un ‘diagramma di contesto’ del software oggetto di misurazione. I diagrammi di contesto rappresentano i flussi di dati tra la porzione di software e i suoi utenti funzionali (esseri umani, dispositivi hardware o altri software) e rappresentano anche i flussi di data tra la porzione di software e la memoria persistente.

Un diagramma di contesto è un’istanziamento di un pattern di misurazione applicato al software oggetto di misurazione. Si riportano i simboli usati nei diagrammi di contesto in Fig. 2.5.

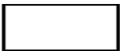
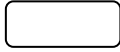

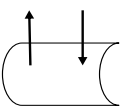
Simbolo	Interpretazione
	La porzione di software da misurare (riquadro con bordo spesso), cioè la definizione di un ambito di misurazione.
	Qualsiasi utente funzionale del software oggetto di misurazione.
	Le frecce rappresentano tutti i movimenti di dati che attraversano un confine (linea tratteggiata) tra un utente funzionale ed il software oggetto di misurazione.
	Le frecce rappresentano tutti i movimenti di dati tra il software oggetto di misurazione e la ‘memoria persistente’. (Il simbolo tipico dei diagrammi di flusso per i ‘depositi di dati’ sottolinea che la memoria persistente è un concetto astratto. Questo simbolo indica che il software non interagisce direttamente con la memoria hardware fisica.)

Figura 2.5 – Simboli dei diagrammi di contesto.

## 2.9 Identificare il livello di granularità

Il metodo COSMIC richiede che i FUR siano espressi ad un livello di dettaglio sufficiente a creare i modelli di misurazione COSMIC: ciò è chiamato livello di granularità.

Per ricavare una dimensione funzionale secondo COSMIC usando le regole dello standard ISO 19761, il livello di granularità necessario è quello al quale si possono identificare e definire i singoli processi funzionali e i loro movimenti di dati. In mancanza di dettagli funzionali ad altri livelli di granularità dei requisiti, le misurazioni dovrebbero svolgersi utilizzando una delle tecniche di approssimazione della dimensione descritte nella relativa linea guida: [‘Dimensionamento Anticipato del Software con il Metodo COSMIC Guida per Praticanti’](#).

NOTA 1. Nelle fasi iniziali di un progetto di sviluppo software, i requisiti effettivi sono specificati ‘ad alto livello’, ossia come sommario o con scarso dettaglio. Al progredire del progetto, i requisiti effettivi vengono raffinati (per es. in versioni successive 1, 2, 3 etc.), rivelando più e più dettagli ‘a livello più basso’. Questi diversi gradi di dettaglio dei requisiti effettivi sono noti come differenti ‘livelli di granularità’.

NOTA 2. I misuratori dovrebbero tener conto del fatto che quando i requisiti evolvono all’inizio del ciclo di vita di un progetto software, in ogni momento parti differenti delle funzionalità software richieste saranno state tipicamente documentate a differenti livelli di granularità.

Per un esempio di misurazione a differenti livelli di granularità e di scomposizione, si veda l’esempio del sistema di telecomunicazioni in: ‘Guideline for early or rapid COSMIC functional size misurazione using approximation approaches’.

### 3. FASE DI MAPPATURA

I processi funzionali sono costituiti da sotto-processi che muovono dati (*'movimenti di dati'*) e che eventualmente possono manipolare i dati (*'manipolazione di dati'*).

#### 3.1 Mappare i FUR sul Modello Generale del Software

La Figura 3.1 mostra i passi per mappare i FUR nei deliverable disponibili del software nella forma richiesta dal Modello Generale del Software COSMIC.

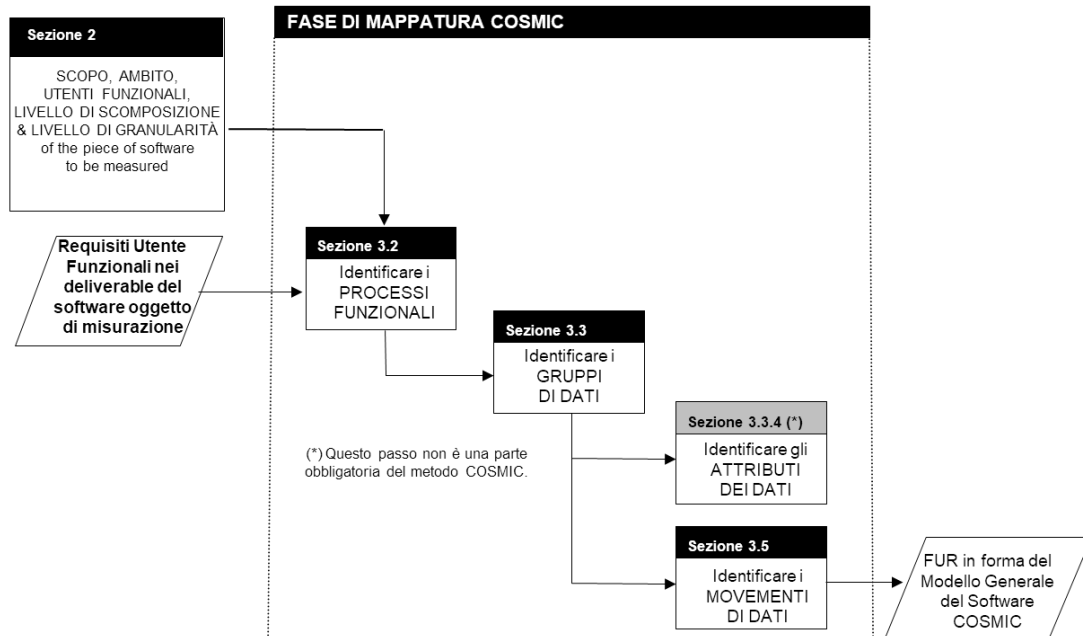


Figura 3.1 – Il processo della Fase di Mappatura COSMIC.

Sono disponibili varie linee guida che descrivono come mappare vari metodi di analisi dati e definizione dei requisiti, usati in differenti domini, sui concetti del metodo COSMIC:

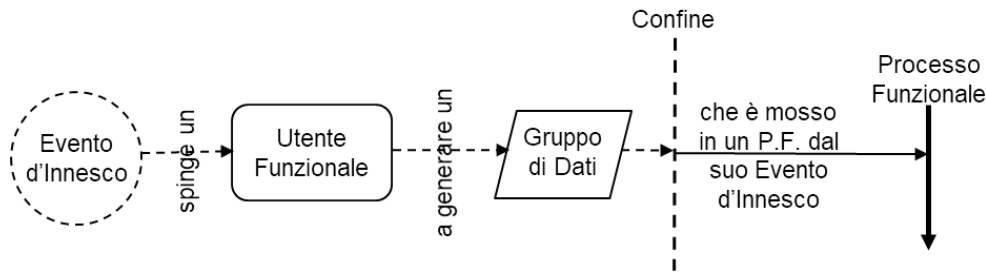
- [Guideline for Sizing Business Application Software](#),
- [Guideline for Sizing Data Warehouse Application Software](#),
- [Guideline for Sizing Service-Oriented Architecture Software](#), and
- [Guideline for Sizing Real-time Software](#).

Per i software appartenenti ai domini [business](#) e [real-time](#) sono anche disponibili delle Quick Reference Guide, che forniscono in poche pagine una sintesi del processo da seguire.

#### 3.2 Identificare i processi funzionali

Il primo passo della Fase di Misurazione è identificare l'insieme dei processi funzionali della porzione di software da misurare, dai suoi FUR.

Le relazioni tra evento d'innescio, utente funzionale e movimento di dati di tipo Entry, che innesca il processo funzionale oggetto di misurazione, sono illustrate in Fig. 3.2, dove: *un evento d'innescio spinge un utente funzionale a generare un gruppo di dati che è mosso dall'Entry d'innescio di un processo funzionale ad avviare il processo funzionale*.



**Figura 3.2 – Relazioni tra evento d’inesco, utente funzionale e processo funzionale.**

NOTA. Per motivi di leggibilità, si omette il riferimento al gruppo di dati quando si afferma che un utente funzionale inizializza un Entry d’inesco che avvia un processo funzionale, o ancor più semplicemente che un utente funzionale inizializza un processo funzionale.

### Guida alla Regola 10: Identificare i processi funzionali

Il processo di identificazione dei processi funzionali, dopo che siano stati identificati gli utenti funzionali, dati i FUR del software oggetto di misurazione, segue la catena logica di Fig. 3.2:

1. Identifica i distinti eventi nel mondo degli utenti funzionali a cui il software oggetto di misurazione deve rispondere – gli ‘eventi d’inesco.

NOTA. Gli eventi d’inesco possono essere identificati nei diagrammi di stato e nei diagrammi di ciclo di vita delle entità, poiché alcune transizioni di stato e di ciclo di vita delle entità corrispondono agli eventi d’inesco a cui deve reagire il software.

2. Identifica quali utenti funzionali del software devono rispondere a ciascun evento d’inesco.
3. Identifica i gruppi di dati (cioè il/gli Entry d’inesco) che ogni utente funzionale può inizializzare in risposta all’evento.
4. Identifica il processo funzionale avviato da ciascun Entry d’inesco.

Usa i seguenti quesiti per verificare che i candidati processi funzionali siano identificati correttamente:

1. Tutti i P.F. della porzione di software misurata risiedono nel medesimo strato?
2. Tutti i P.F. includono un Entry ed almeno 1 Write o Exit?

## 3.3 Identificare i gruppi di dati

### 3.3.1 Identificare i gruppi di dati

Avendo identificato i processi funzionali, il passo successivo è identificare i loro movimenti di dati. Le linee guida seguenti sono di ausilio nell’identificazione dei gruppi di dati e quindi degli oggetti d’interesse, in particolare negli output dei processi funzionali.

### Guida alla Regola 11: Identificare gruppi di dati differenti mossi nel medesimo processo funzionale

Per tutti gli attributi dei dati presenti in un Entry/Exit/Read/Write del processo funzionale:

- a) insiemi di attributi dei dati occorrenti con frequenze differenti descrivono oggetti d’interesse differenti;
- b) insiemi di attributi dei dati occorrenti con la stessa frequenza, ma attributo/i chiave identificativi differente/i descrivono oggetti d’interesse differenti
- c) tutti gli attributi dei dati in un insieme risultante dall’applicazione dei punti a) e b) di questa linea guida appartengono al medesimo gruppo di dati, a meno che i FUR non specifichino che possa esserci più di un gruppo di dati descrivente il medesimo oggetto d’interesse (v. **Guida alle Regole 13 e 14 – Unicità dei movimenti di dati**, casi b) e c)).

NOTA 1. Un utente funzionale del software oggetto di misurazione può essere l'oggetto d'interesse di un gruppo di dati inviato o ricevuto dall'utente funzionale.

NOTA 2. In teoria, un gruppo di dati potrebbe contenere un unico attributo, se questo è tutto ciò che è richiesto dal punto di vista dei FUR per descrivere il suo oggetto d'interesse. In pratica, questo caso si presenta frequentemente nel software delle applicazioni realtime (per es. il gruppo di dati inserito per trasportare il segnale di un clock in tempo reale o l'input dello stato di un sensore); è meno frequente nel software applicativo gestionale.

NOTA 3. Non c'è nulla di assoluto riguardo ad un oggetto d'interesse, cioè si identifichino gli oggetti d'interesse per ciascun processo funzionale. Una 'cosa' può essere un oggetto 'd'interesse' per un utente funzionale attraverso uno o più processi funzionali, ma non un oggetto 'd'interesse' per un altro utente funzionale attraverso altri processi funzionali, anche per il medesimo software oggetto di misurazione

L'origine di un gruppo di dati può assumere varie, per esempio:

- a) una struttura fisica di record in un dispositivo di memoria hardware (file, tabella di database, memoria ROM, etc.),
- b) una struttura fisica nella memoria volatile di un computer (struttura dati allocata dinamicamente o tramite un blocco pre-allocato nello spazio di memoria).
- c) una presentazione aggregata di attributi dei dati funzionalmente correlati in un dispositivo di input/output (schermo, stampa, display di pannello di controllo, etc.),
- d) un messaggio nella comunicazione tra un dispositivo e il computer, in una rete, etc.

### **3.3.2 Sull'identificazione degli oggetti d'interesse e dei gruppi di dati**

Le definizioni ed i principi relativi a oggetto d'interesse e gruppo di dati sono volutamente generali per essere applicabili alla gamma più ampia possibile di software: ciò a volte rende difficoltoso applicare le definizioni ed i principi nel misurare una specifica porzione di software. Vedi la Parte 3 per esempi che aiutino nell'applicazione dei principi a casi specifici.

Quando ci si trova ad analizzare un group of attributi dei dati mosso dentro o fuori di un processo funzionale o da un processo funzionale verso o dalla memoria persistente, è *di cruciale importanza* stabilire se tutti gli attributi trasportino dati riguardanti un solo 'oggetto d'interesse', poiché è quest'ultimo che determina il numero di distinti 'gruppi di dati', come definito nel metodo COSMIC come mosso da un movimento di dati.

Per esempio, se gli attributi dei dati oggetto di inserimento in un processo funzionale sono attributi di tre distinti oggetti d'interesse, si devono identificare tre distinti movimenti di dati di tipo 'Entry'. Decidere del numero di gruppi di dati può essere difficoltoso quando si analizza l'output di un processo funzionale di un'applicazione gestionale, che può includere:

- gruppi di dati multipli, ognuno descrivente un diverso oggetto d'interesse, per es. un report che riporta totali a vari livelli di aggregazione;
- i risultati delle interrogazioni il cui output varierà in funzione dell'input;
- gruppi di dati che possono anche essere scorrelati tra di loro, per es. una fattura che include la promozione di un servizio non correlato al suo oggetto.

Nell'analizzare degli output complessi, per es. report con dati descrittivi vari oggetti d'interesse, considera ogni distinto candidato gruppo di dati come se fosse l'output di un processo funzionale separato. Ciascuno (tipo di) gruppo di dati così identificato deve essere distinto e contato separatamente nel misurare il report complesso.

### **3.3.3 Dati o gruppi che non sono candidati validi di gruppi di dati**

Qualsiasi dato che compare nelle schermate di input o di output o nei report non correlato ad un oggetto d'interesse per un utente funzionale non dovrebbe essere identificato come denotante un gruppo di dati.

Il Modello Generale del Software COSMIC assume che tutta la manipolazione dei dati in un processo funzionale è associata con i quattro tipi di movimento di dati. Quindi, non si possono identificare dei gruppi di dati originati dalla manipolazione dei dati in un processo funzionale in aggiunta ai gruppi di dati mossi dagli Entry, Exit, Read e Write del processo funzionale.

### **3.3.4 Identificare gli attributi dei dati (opzionale)**

Nel metodo COSMIC, non è obbligatorio identificare gli attributi dei dati. Tuttavia, è necessario comprendere il concetto di 'attributo dei dati' per capire come misurare le 'modifiche'. Il requisito di modifica di un attributo dei dati può portare a denotare come 'modificato' il movimento di dati a cui appartiene l'attributo in questione.

Inoltre, può essere utile analizzare e identificare gli attributi dei dati nel differenziare tra loro i gruppi di dati e gli oggetti d'interesse.

## **3.4 Identificare i movimenti di dati**

Questo passo consiste nell'identificare i movimenti di dati (Entry, Exit, Read e Write) di ogni processo funzionale.

La seguente linea guida aiuta a confermare lo stato di un candidato movimento di dati Entry.

### **Guida alla Regola 16: Entry (E)**

- a) Il gruppo di dati di un Entry d'innescò può consistere di un unico attributo dei dati che informa semplicemente il software che 'è avvenuto l'evento Y'.
- b) Per segnali di clock che sono eventi d'innescò identifica un Entry da parte di un utente funzionale, in questo caso il clock.
- c) A meno che non sia necessario uno specifico processo funzionale, ottenere la data e/o l'ora dal clock di sistema non è considerato un Entry o qualsiasi altro movimento di dati.

NOTA. Molto spesso, nel software gestionale, il gruppo di dati di un Entry d'innescò possiede vari attributi di dati, che informano il software che 'è avvenuto l'evento Y e questi sono i dati di quel particolare evento'.

### **Guida alla Regola 17: Exit (X)**

- a) Per un'interrogazione che espone in output del testo fisso (dove 'fisso' significa che il messaggio non contiene valori di dati variabili), identifica un Exit per l'output di testo fisso.
- b) Nell'identificare gli Exit, ignora tutti i campi e altre intestazioni che consentono agli utenti umani di comprendere i dati in output.

### **Guida alla Regola 18: Read (R)**

Non identificare un Read quando i FUR del software oggetto di misurazione specificano un qualsiasi utente funzionale software o hardware come la fonte di un gruppo di dati o come il mezzo per muovere il gruppo di dati nella memoria persistente. L'interazione con altri utenti funzionali avviene per definizione attraverso un confine (Modello Generale del Software, 1° principio) ed è gestita da un movimento di dati di Entry. Il Read effettivo ha luogo all'interno del confine del software oggetto di misurazione.

### Guida alla Regola 19: Write (W)

Non identificare un Write quando i FUR del software oggetto di misurazione specificano un qualsiasi utente funzionale software o hardware come la destinazione del gruppo di dati o come il mezzo per reperire un gruppo di dati memorizzato in modo persistente. L'interazione con altri utenti funzionali avviene per definizione attraverso un confine (Modello Generale del Software, 1° principio) ed è gestita da un movimento di dati di eXit. Il Write effettivo ha luogo all'interno del confine del software oggetto di misurazione.

NOTA. Quando i FUR richiedono di memorizzare dati nella o reperire dati dalla memoria, il misuratore deve verificare se i dati possono essere memorizzati o reperiti all'interno del confine, cioè nella/dalla 'memoria persistente', o se è necessario che i dati siano memorizzati/reperiti tramite un utente funzionale del software oggetto di misurazione (cioè tramite un'altra porzione di software, direttamente o da/verso un dispositivo hardware).

### Guida alle Regole da 16 a 19 – Manipolazione dei dati associata ai movimenti di dati

La manipolazione di dati associata con uno qualsiasi dei movimenti di dati non include qualsiasi ulteriore manipolazione di dati che sia richiesta dopo che il movimento di dati si sia completato con successo, né include qualsiasi manipolazione di dati associata con un altro movimento di dati.

La seguente linea guida copre la situazione più comune (punto a)) e altri possibili casi validi (punti b) e c)):

- In a), le occorrenze del gruppo di dati sono soggette agli stessi FUR, quindi si identifica un solo gruppo di dati ed un solo movimento di dati.
- In b) e c), lo stesso si applica *a ogni distinto gruppo di dati separatamente*, quindi si identificano un solo gruppo di dati ed un solo movimento di dati per ciascun distinto gruppo di dati.

### Guida alle Regole 13 e 14 – Unicità dei movimenti di dati

a) A meno che i FUR non siano come ai punti b) o c), tutti i dati descrittivi uno stesso oggetto d'interesse per il quale sia richiesto l'inserimento in ciascun processo funzionale sono identificati come un solo gruppo di dati mosso da un solo Entry.

NOTA 1. Un processo funzionale può, ovviamente, avere molteplici Entry, dove ciascun Entry muove dati descrittivi un diverso oggetto d'interesse.

NOTA 2. La medesima linea guida equivalente vale per un movimento di dati di tipo Read, Write o Exit in ciascun processo funzionale.

b) Se i FUR specificano che si devono inserire differenti gruppi di dati in ciascun processo funzionale, ognuno da parte di un diverso utente funzionale, dove ogni gruppo di dati descrive il medesimo oggetto d'interesse, si identifica un Entry per ciascuno di questi differenti gruppi di dati.

NOTA 1. La medesima linea guida equivalente vale per gli Exit di dati verso differenti utenti funzionali da parte di ciascun processo funzionale.

NOTA 2. Ciascun processo funzionale ha un solo Entry d'innescamento.

c) Se i FUR specificano che si devono muovere differenti gruppi di dati dalla memoria persistente in ciascun processo funzionale, ognuno dei quali descrive il medesimo oggetto d'interesse, si identifica un Read per ciascuno di questi differenti gruppi di dati.

NOTA 1. La medesima linea guida equivalente vale per i Write in ciascun processo funzionale.

NOTA 2. Questo punto è analogo al punto b). Nel caso dei FUR di lettura di differenti gruppi di dati descrittivi il medesimo oggetto d'interesse, essi saranno stati probabilmente originati da utenti funzionali differenti. Nel caso dei FUR di scrittura di differenti gruppi di dati, essi saranno probabilmente resi disponibili per la lettura da parte di utenti funzionali differenti.

### **Guida alle Regole da 16 e 17 – Processo funzionale che richiede dei dati da parte di un utente funzionale**

- a) Quando il processo funzionale non necessita di specificare all'utente funzionale quali dati inviare, è sufficiente un solo Entry (per oggetto d'interesse).
- b) Quando il processo funzionale necessita di specificare all'utente funzionale quali dati inviare, è necessario un Exit seguito da un Entry.

### **Guida alle Regole da 16 a 19 – Comandi di controllo in applicazioni con interfaccia umana**

In un'applicazione con interfaccia umana, i 'comandi di controllo' si ignorano, poiché non implicano alcun movimento di dati riguardante un oggetto d'interesse.

Messaggi di errore e di conferma sono forme specifiche di un Exit e si applicano le relative regole di identificazione.

### **Guida alle Regole da 16 a 19 – Messaggi di errore/conferma & altri indicatori di condizioni di errore**

- a) Si identifica un solo Exit per tenere in conto tutti i tipi di messaggi di errore/conferma forniti da ciascun processo funzionale del software oggetto di misurazione, per tutte le possibili cause in base ai suoi FUR.
- b) Se un messaggio ad un utente funzionale umano in aggiunta alla conferma che i dati inseriti sono stati accettati o che presentato degli errori, questi dati aggiuntivi si identificano come un distinto gruppo di dati mosso da un Exit nel modo usuale.
- c) Tutti gli altri dati, emessi o ricevuti dal software oggetto di misurazione, da/verso i suoi utenti funzionali hardware o software, dovrebbero essere analizzati in base ai FUR come Exit o Entry, rispettivamente, secondo le normali regole COSMIC, a prescindere dal fatto che i valori dei dati indichino una condizione di errore o meno.
- d) I Read ed i Write si considera che tengano in conto qualsiasi riferimento associato di condizioni d'errore. Perciò, non si identifica alcun Entry verso il processo funzionale oggetto di misurazione per qualsiasi indicazione di errore ricevuta come risultato di un Read o Write dei dati persistenti.
- e) Non si identifica alcun Entry o Exit per qualsiasi messaggio indicante una condizione di errore, che potrebbe essere emesso nell'utilizzo del software oggetto di misurazione, ma che non si richiede di elaborare in alcun modo nei FUR di quel software, per es. un messaggio di errore emesso dal sistema operativo.

## **3.5 Misurare le componenti di un sistema software distribuito**

Quando lo scopo di una misurazione è di misurare separatamente la dimensione di ogni componente di un sistema software distribuito, si deve definire un distinto ambito di FSM per ciascuna componente. In tal caso, il dimensionamento dei processi funzionali di ciascuna componente segue tutte le regole già descritte.

Dal processo di ogni misurazione (... definire l'ambito, poi gli utenti funzionali e il confine, etc. ...), segue che se una porzione di software consiste di due o più componenti, non può esserci alcuna sovrapposizione tra gli ambiti di FSM delle componenti. L'ambito di FSM di ogni componente deve definire un insieme di processi funzionali completi. Per esempio, non può esserci un processo funzionale in parte in un ambito e in parte in un altro. Allo stesso modo, i processi funzionali nell'ambito di misurazione di una componente non hanno alcuna nozione dei processi funzionali nell'ambito di un'altra componente, anche sebbene le due componenti scambino messaggi tra di loro.



Gli utenti funzionali di ogni componente vengono determinati dall'esame di dove avvengano gli eventi che innescano i processi funzionali della componente esaminata. (Gli eventi d'innescio possono avvenire solo nel mondo di un utente funzionale.)

### 3.6 Riuso del software

Due o più processi funzionali nel medesimo software oggetto di misurazione possono avere una parte di funzionalità identica o molto simile tra di loro, descritta altrimenti separatamente nei requisiti. Questo fenomeno è denotato come 'commonality', o 'similarità', funzionale.

Tuttavia, ogni processo funzionale è definito, modellato e misurato indipendentemente da, cioè senza riferimento a, 'altri' FUR nel medesimo software oggetto di misurazione.

Quindi, se i FUR di un dato processo funzionale fanno riferimento ai FUR in altri punti dei requisiti, la dimensione di quella parte di funzionalità cui si fa riferimento va inclusa nella dimensione del processo funzionale misurato.

### 3.7 Misurare la dimensione delle modifiche del software

Una 'modifica funzionale' ad un software esistente è interpretata nel metodo COSMIC come 'qualsiasi combinazione di aggiunta di nuovi movimenti di dati o di modifica o cancellazione di movimenti di dati esistenti, includendo la manipolazione di dati associata'. I termini 'manutenzione evolutiva' ('enhancement') e 'manutenzione ordinaria' ('maintenance') sono spesso utilizzati per quello che qui chiamiamo 'modifica funzionale'.

La necessità di una modifica al software può sorgere da:

- un nuovo FUR (cioè, sole aggiunte alle funzionalità esistenti), o
- una modifica dei FUR (che eventualmente implica aggiunte, modifiche e cancellazioni), o
- un'esigenza di 'manutenzione ordinaria' per correggere un difetto.

Le regole per dimensionare una qualsiasi di queste modifiche sono sempre le stesse, ma si avvisa il misuratore di distinguere le varie circostanze, nello svolgere misurazioni e stime di performance.

Un movimento di dati è considerato funzionalmente modificato come segue.

#### Guida alla Regola 24 – Modificare un movimento di dati

a) Un movimento di dati è considerato funzionalmente modificato se vale almeno una delle seguenti:

- si modifica il gruppo di dati mosso,
- si modifica la manipolazione di dati associata.

b) Un gruppo di dati è modificato se vale almeno una delle seguenti:

- si aggiungono uno o più attributi nuovi al gruppo di dati,
- si rimuovono uno o più attributi esistenti dal gruppo di dati,
- si modificano uno o più attributi esistenti, per es. nel significato o nel formato (ma non nei valori).

c) Una manipolazione di dati è modificata se è modificata funzionalmente in qualsiasi modo.

d) Se si deve modificare un movimento di dati per una modifica della manipolazione di dati associata ad esso e/o per una modifica nel numero o nel tipo degli attributi del gruppo di dati mosso, si misura un solo CFP modificato, a prescindere dal numero effettivo di modifiche in quel movimento di dati.



- e) Se si deve modificare un gruppo di dati, i movimenti di dati che muovono quel gruppo di dati che non sono impattati funzionalmente dalla modifica al gruppo di dati stesso non si identificano come modificati.

NOTA. La modifica di un qualsiasi dato che compare nelle schermate di input o di output, non correlato ad un oggetto d'interesse per un utente funzionale, non si identifica come un CFP modificato.

- f) Una normale convenzione di misurazione è che la dimensione funzionale di una porzione di software non cambia se si deve modificare il software per correggere un difetto in modo da riallineare il software con i suoi FUR. La dimensione funzionale del software non cambia se la modifica consiste nel correggere un difetto nei FUR.

Movimenti di dati modificati non hanno impatto sulla dimensione della porzione di software, poiché essi esistono sia prima che dopo delle modifiche.

Quando si rimpiazza completamente una porzione di software, per esempio riscrivendola da zero, con o senza estensioni od omissioni di parti delle funzionalità, la dimensione funzionale di questa 'modifica' è la dimensione per intero del software di rimpiazzo, misurata secondo le normali regole di dimensionamento di un nuovo software.

NOTA. Solitamente, la dimensione di una modifica funzionale (qui discussa) e la modifica della dimensione funzionale del software differiscono tra loro.

## 4. ALTRI ARGOMENTI

### 4.1 Estendere il Metodo di Misurazione COSMIC – Estensioni locali

Il metodo COSMIC per misurare la dimensione funzionale non presume di misurare tutti i possibili aspetti della 'dimensione' del software. Quindi, il metodo non è attualmente progettato per misurare distintamente ed esplicitamente la dimensione dei FUR relativi ai sotto-processi di manipolazione dei dati. L'impatto sulla dimensione dei sotto-processi di manipolazione dei dati è tenuto in conto per mezzo di un'assunzione semplificatrice, che è valida per una vasta gamma di domini software.

Nonostante ciò, la misura della dimensione COSMIC è considerata una buona approssimazione per gli scopi dichiarati e i domini di applicazione del metodo. Nondimeno, è possibile che in un contesto locale di un'organizzazione che usi il metodo di misurazione COSMIC, sia desiderabile prendere esplicitamente in considerazione tali funzionalità in modo significativo come uno standard locale. Quando si usano tali estensioni locali, si devono riportare i risultati delle misurazioni secondo la convenzione speciale presentata nella Sezione 6.

### 4.2 Il Metodo di Misurazione COSMIC nel contesto Agile

Il metodo COSMIC è usato con successo per misurare le dimensioni delle User Story nello sviluppo agile, che può avere numeri molto bassi di movimenti di dati. Questa prassi è ben fondata, con molti report delle dimensioni in CFP delle iterazioni (o 'sprint') agile, ottenute aggregando le dimensioni delle singole User Story, le quali sono molto ben correlate con l'impegno di sviluppo dell'iterazione (e molto meglio rispetto alle dimensioni misurate con gli Story Point). V. <https://cosmic-sizing.org/publications/guideline-for-sizing-agile-projects-with-cosmic/>.

## 5. FASE DI MISURAZIONE

Il processo generale per misurare una porzione di software, una volta che i suoi Requisiti Utente Funzionali siano stati espressi in termini del Modello Generale del Software COSMIC, è riassunto in Fig. 5.1 di seguito.

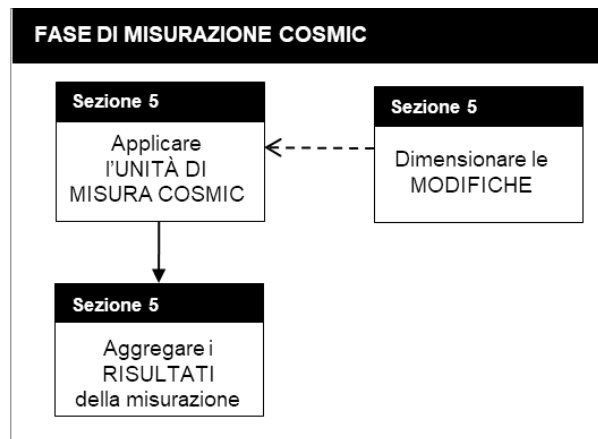


Figura 5.1 – Il processo della Fase di Misurazione COSMIC.

### Guida alla Regola 23 – Aggregare le dimensioni funzionali

- Le dimensioni di porzioni di software o di modifiche di porzioni di software possono sommarsi tra loro solo se misurate al medesimo livello di granularità dei processi funzionali dei loro FUR.
- Le dimensioni di porzioni di software e/o modifiche delle dimensioni di porzioni di software di un unico strato o di differenti strati si sommano tra loro solo se ha senso farlo, alla luce dello scopo della misurazione.
- La dimensione di una porzione di software si ottiene sommando le dimensioni delle sue componenti (a prescindere da come sia scomposta tale porzione) ed eliminando i contributi alle dimensioni dei movimenti di dati tra una componente e l'altra.

In ogni strato identificato, la funzione di aggregazione è completamente scalabile. Si possono calcolare subtotali per specifici processi funzionali, o un totale per tutti i processi funzionali del software in uno strato, a seconda dello scopo e dell'ambito della misurazione.

## 6. PRESENTARE LA MISURAZIONE

Si devono riportare e registrare i risultati ed i dettagli della misurazione per assicurarne un'interpretazione sempre priva di ambiguità. Si devono riportare e archiviare i risultati di ogni misurazione COSMIC secondo le seguenti convenzioni.

### Guida alla Regola 25 – Catalogare la misurazione COSMIC

Un risultato di misurazione COSMIC è riportato come 'x CFP (v)', dove:

- 'x' rappresenta il valore numerico della dimensione funzionale,
- 'v' rappresenta l'identificativo della versione del metodo standard COSMIC usata per ottenere il valore numerico di dimensione funzionale 'x'.

NOTA. Se si è fatto uso di un metodo di approssimazione locale per ottenere la misurazione, ma, a parte ciò, la misurazione è stata svolta applicando le convenzioni di una versione standard del metodo COSMIC, si applica la convenzione di catalogazione sopra descritta, ma si dovrebbe annotare altrimenti l'utilizzo del metodo di approssimazione.

### Guida alla Regola 25 – Catalogare le estensioni locali COSMIC

Un risultato di misurazione COSMIC facente uso di estensioni locali è riportato come:

'x CFP (v.) + z FP locali', dove:

- 'x' rappresenta il valore numerico ottenuto aggregando tutti i singoli risultati di misurazione conformi al metodo standard COSMIC, versione 'v',
- 'v' rappresenta l'identificativo della versione del metodo standard COSMIC usata per ottenere il valore numerico di dimensione funzionale 'x'.
- 'z' rappresenta il valore numerico ottenuto aggregando tutti i singoli risultati di misurazione ottenuti da estensioni locali al metodo COSMIC.